

## Simulasi Sistem Pengendalian Lampu Penerangan Berbasis Program Komputer (PC)

Libianko Sianturi, S.T., M.T.<sup>1)</sup>, Febrianre Hutaaruk<sup>2)</sup>  
Program Studi Teknik Elektro Fakultas Teknik Universitas HKBP Nommensen  
email: [libiankosianturi@uhn.ac.id](mailto:libiankosianturi@uhn.ac.id)

### Abstrak

*Di setiap gedung-gedung bertingkat pasti ada sebuah sakelar ON atau OFF untuk menghidupkan dan mematikan lampu di gedung tersebut, sehingga untuk mematikan atau menghidupkan lampu akan memakan waktu dan tenaga. Sebuah simulator ini dibuat untuk memberikan solusi untuk menghidupkan dan mematikan lampu dimaksud. Simulator ini akan bekerja dengan menggunakan Personal Computer sebagai alat pengontrol semua lampu yang terpasang ke relay yang terdapat pada kaki-kaki mikrokontroler. Simulator akan berfungsi untuk menghidupkan atau mematikan lampu dengan menggunakan tampilan program Visual Basic yang dihubungkan dengan mikrokontroler jenis ATMEL AT89C2051 lewat komunikasi serial menggunakan kabel DB9.*

**Kata kunci :** Kendali lampu, program komputer, visual basic, mikrokontroler

### Abstract

*In every high-rise building there must be an ON or OFF switch to turn the lights on and off in the building, so turning off or turning on the lights will take time and effort. A simulator was created to provide a solution to turn on and turn off the light in question. This simulator will work by using a Personal Computer as a controller for all lights attached to the relays located on the legs of the microcontroller. The simulator will function to turn on or turn off the lights using a Visual Basic program display that is connected to an ATMEL AT89C2051 type microcontroller via serial communication using a DB9 cable.*

**Key Words :** Lamp controler, computer programm, visual basic, microcontroller

### PENDAHULUAN

Sebuah gedung perkantoran yang biasanya terdiri dari banyak ruangan, ada yang beroperasi hampir 24 jam. Tenaga listrik memiliki peranan penting, mulai dari menghidupkan komputer-komputer, printer, dan yang tidak kalah pentingnya sebagai tenaga yang menghidupkan lampu-lampu untuk penerangan.

Untuk itu, memikirkan suatu cara dalam menghemat penggunaan energi listrik, lampu dapat digunakan lebih efisien dan jauh lebih hemat dimana lampu dapat di nyalakan dan dapat dimatikan dari sebuah komponen pengendali. Dengan memanfaatkan pemrograman Visual Basic yang terinstal di sebuah Personal Computer, program akan berinteraksi terhadap pengguna, dengan memanfaatkan komunikasi data serial pada komputer yang terhubung dengan mikrokontroler jenis ATMEL AT89C2051. Mikrokontroler dihubungkan dengan relay sebagai saklar yang berfungsi memutus dan

menyambung sumber tegangan PLN dan lampu. Adapun tujuan penulisan ini adalah untuk merancang, sehingga akan diperlukan suatu sistem sebagaimana yang di inginkan yaitu pertama, merealisasikan simulasi pengendalian lampu agar bekerja dengan baik, sehingga PC dapat berinteraksi dengan pengguna, untuk mengendalikan lampu ON dan OFF. Kedua, membentuk prototype pengendalian lampu untuk dapat diimplementasikan kesistem fisik yang sebenarnya, dengan melihat dari segi kemudahan dan keamanan dalam mengendalikan satu titik lampu.

### DASAR TEORI

#### Mikrokontroler

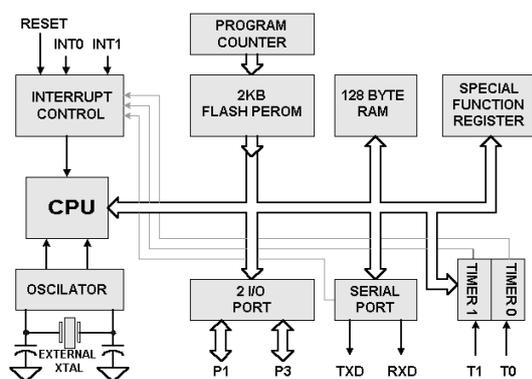
Mikroprosesor merupakan bagian dari sebuah CPU (Central Processing Unit) dari sebuah komputer, memiliki I/O (Input / Output) sebagai peripheral pemrosesan data yang membentuk suatu sistem lengkap, dan mikroprosesor akan berfungsi bila ada

perangkat pendukung berupa RAM (Random Acces Memory), ROM (Read Only Memory).

Mikrokontroler AT89C2051 adalah kelompok mikrokontroler dengan informasi ROM atau EPROM, yang dikeluarkan oleh ATMEL, dengan keistimewaan sebagai berikut:

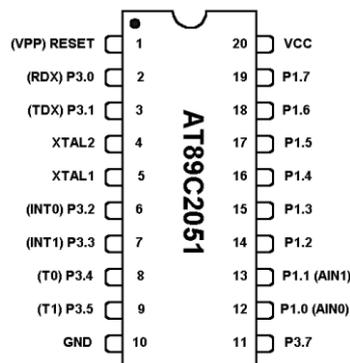
1. Memiliki satu buah unit pemrosesan pusat 8 bit,
2. Memiliki *Internal Oscillator* dan *Timing Circuit*,
3. Memiliki Internal RAM sebesar 128 *byte*,
4. Memiliki Programmable I/O port yang terdiri atas 15 buah jalur I/O,
5. Memiliki dua buah Timer/Counter 16 bit,
6. Memiliki lima buah jalur interupsi,
7. Memiliki satu buah serial I/O port dengan Serial Control,
8. Kemampuan untuk melakukan operasi *Boolean*,
9. Memiliki 2 Kbyte sistem *Flash Memory*,
10. Frekuensi kerja antara 0 Hz sampai dengan 24 Hz.

Berikut ini pada Gambar 1 adalah blok diagram fungsional dari mikrokontroler AT89C2051.



Gambar 1. Blok Diagram AT89C2051

Susunan pin Mikrokontroler AT89C2051 diperlihatkan pada gambar 2 berikut ini:



Gambar 2. Susunan Pin AT89C2051

Penjelasan dari masing-masing pin adalah sebagai berikut:

1. *Port 1* (pin 12 sampai dengan 19) adalah *port* paralel 8-bit dua arah (*bidirectional*) yang dapat digunakan untuk berbagai keperluan.
2. *Port 1.2* sampai dengan 1.7 dilengkapi dengan *internal Pull-ups*, sedangkan *port 1.0* dan 1.1 memerlukan *eksternal pull-ups*, *port* ini juga melayani inputan positif (AIN0) dan inputan negatif (AIN1).
3. Pin 1 adalah pin masukan untuk reset. Perubahan pulsa dari rendah ke tinggi pada pin ini membuat CPU menghentikan semua kegiatan dan kembali ke keadaan awal untuk menjalankan program dari awal.
4. Pin 2 sampai dengan 17 (*port 3*) adalah *port* paralel 8-bit dua arah yang memiliki fungsi pengganti, yaitu:
  - a. P3.0 sebagai pin RxD, yaitu pin input data serial (*Serial Data Receiver*) untuk komunikasi serial,
  - b. P3.1 sebagai pin TxD, yaitu pin input data serial (*Serial Data Transmitter*) untuk komunikasi serial,
  - c. P3.2 (INT0), input interupsi eksternal 0,
  - d. P3.3 (INT1), input interupsi eksternal 1,
  - e. P3.4 (T0), input eksternal untuk *timer 0* atau *counter 0*,
  - f. P3.5 (T1), input eksternal untuk *timer 1* atau *counter 1*,
5. Pin 4 (XTAL1) adalah pin masukan ke rangkaian Osilator internal. Sebuah osilator kristal atau sumber osilator luar dapat digunakan,
6. Pin 5 (XTAL2) adalah pin keluaran ke rangkaian osilator internal. Pin ini dapat dipakai bila menggunakan osilator kristal,
7. Pin 10 (*Ground*) untuk dihubungkan ke *ground*.

8. Pin 20 (Vcc) dihubungkan ke Vcc (+5 Volt).

Mikrokontroler AT89C2051 mempunyai 20 kaki, 16 kaki diantaranya digunakan sebagai port paralel. Satu port paralel terdiri dari 8 kaki, dengan demikian 16 kaki tersebut membentuk 2 buah port paralel, yaitu port 1 dan port 3.

**a) Port 1**

Port 1 merupakan port I/O dwi-arah yang dilengkapi dengan *pull-up* internal. Penyangga keluaran Port 1 mampu memberikan atau menyerap arus empat masukan TTL (sekitar 1,6 mA).

Jika '1' dituliskan pada kaki-kaki port1, maka masing-masing kaki akan di-*pull-high* dengan pull-up internal sehingga dapat digunakan sebagai masukan.

**b) Port 3**

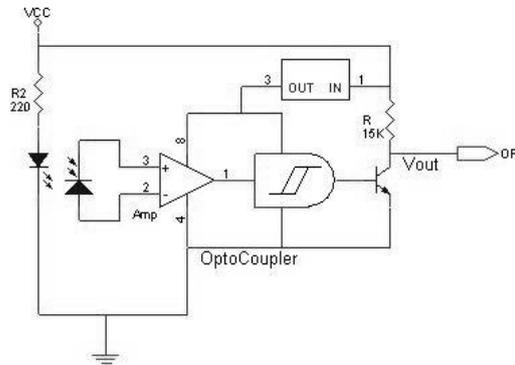
Port 3 merupakan port I/O dwi-arah dengan dilengkapi pullup internal. Penyangga keluaran port 3 mampu memberikan atau menyerap arus empat masukan TTL (sekitar 1,6 mA).

Perangkat lainnya yaitu *Optocoupler* adalah suatu piranti yang terdiri dari 2 bagian yaitu *transmitter* dan *receiver*, yaitu antara bagian cahaya dengan bagian deteksi sumber cahaya terpisah seperti pada Gambar 3. *Optocoupler* adalah suatu komponen penghubung (*coupling*) yang bekerja berdasarkan picu cahaya *optic*. *Optocoupler* terdiri dari dua bagian yaitu:

- a) Pada *transmitter* dibangun dari sebuah LED infra merah. Jika dibandingkan dengan menggunakan LED biasa, LED infra merah memiliki ketahanan yang lebih baik terhadap sinyal tampak. Cahaya yang dipancarkan oleh LED infra merah tidak terlihat oleh mata telanjang.
- b) Pada bagian *receiver* dibangun dengan dasar komponen *Photodiode*. *Photodiode* merupakan suatu transistor yang peka terhadap tenaga cahaya. Suatu sumber cahaya menghasilkan energi panas, begitu pula dengan spektrum infra merah. Karena spektrum infra mempunyai efek panas yang lebih besar dari cahaya tampak, maka *Photodiode* lebih peka untuk menangkap radiasi dari sinar infra merah.

Prinsip kerja dari *optocoupler* adalah :

- Jika antara Photodiode dan LED terhalang maka Photodiode tersebut akan *off* sehingga *output* dari kolektor akan berlogika *high*.
- Sebaliknya jika antara Photodiode dan LED tidak terhalang maka *Photodiode* tersebut akan *on* sehingga *output*-nya akan berlogika *low*.



Gambar 3. Gambar Rangkaian Optocoupler  
Ditinjau dari penggunaannya, fisik *optocoupler* dapat berbentuk seperti Gambar 4. Bila hanya digunakan untuk mengisolasi level tegangan atau data pada sisi transmitter dan sisi receiver, maka *optocoupler* ini biasanya dibuat dalam bentuk solid (tidak ada ruang antara LED dan Photodiode).



Gambar 4. Rangkaian IC Optocoupler

Dari segi komunikasi dikenal ada dua cara komunikasi secara serial, yaitu komunikasi data serial sinkron dan komunikasi data serial secara asinkron. Pada komunikasi data serial sinkron, *clock* dikirimkan bersama-sama dengan data serial, sedangkan komunikasi serial asinkron, *clock* tidak dikirimkan bersama data serial, tetapi dibangkitkan secara sendiri-sendiri baik pada sisi pengirim (*transmitter*) maupun pada sisi penerima (*receiver*). Pada IBM PC kompatibel port serialnya termasuk jenis asinkron. Komunikasi

serial ini dikerjakan oleh UART (*Universal Asynchronous Receiver/Transmitter*). IC UART dibuat khusus untuk mengubah data paralel menjadi data serial yang kemudian diubah kembali menjadi data paralel.

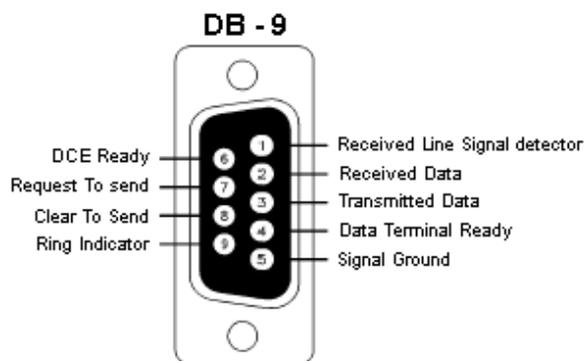
Standar Standar sinyal serial RS232 memiliki ketentuan level tegangan sebagai

Nomor Pin	Nama Sinyal	Direc-tion	Keterangan
1	DCD	In	<i>Data Carrier Detect/Received Line Signal Detect</i>
2	RxD	In	<i>Received Data</i>
3	TxD	Out	<i>Transmit Data</i>
4	DTR	Out	<i>Data Terminal Ready</i>
5	GND	-	<i>Ground</i>
6	DSR	In	<i>Data Set Ready</i>
7	RST	Out	<i>Request to Send</i>
8	CTS	In	<i>Clear to Send</i>
9	RI	In	<i>Ring Indicator</i>

berikut:

- Logika '1' disebut 'Mark' terletak antara -3 Volt hingga -25 Volt.
- Logika '0' disebut 'Space' terletak antara +3 Volt hingga +25 Volt.

Selanjutnya pada gambar 5 diberikan struktur konektor port serial DB-9 pada bagian belakang CPU. Pada komputer PC kompatibel biasanya kita dapat menemukan dua konektor port serial DB-9 yang biasa dinamai COM1 dan COM2.



Gambar 5. Struktur Konektor Serial DB-9 (COM1)  
Keterangan mengenai fungsi saluran RS232 pada konektor DB-9 adalah sebagai berikut:

- Received Line Signal Detect*, dengan saluran ini DCE memberitahukan ke DTE bahwa pada terminal masukan ada data masuk.

- Received Data*, digunakan DTE untuk menerima data dari DCE.
- Transmit Data*, digunakan DTE untuk mengirimkan data ke DTE.
- Data Terminal Ready*, pada saluran ini DTE memberitahukan kesiapan terminalnya.
- Signal Ground*, saluran ground.
- Ring Indicator*, pada saluran ini DCE memberitahukan bahwa DTE sebuah stasiun menghendaki hubungan dengannya.
- Clear to Send*, dengan saluran ini DCE memberitahukan bahwa DTE boleh mulai mengirim data.
- Request To Send*, dengan saluran ini DCE diminta mengirim data oleh DTE.
- DCE Ready*, sinyal aktif pada saluran ini menunjukkan bahwa DCE sudah siap.

Tabel 1. Konfigurasi Pin dan Nama Sinyal Konektor Serial

### Program MCS51

Secara umum kelompok instruksi yang dipakai untuk mengatur program terdiri atas instruksi-instruksi JUMP (setara dengan GOTO dalam Pascal), instruksi-instruksi untuk membuat dan memakai sub-rutin/modul (setara dengan PROCEDURE dalam Pascal) dan instruksi-instruksi JUMP bersyarat (Conditional Jump, setara dengan pernyataan IF ... THEN dalam Pascal).

*Program Counter* adalah satu-satunya register dalam mikrokontroler yang mengatur aliran program, maka kelompok instruksi pengatur aliran program tersebut, semuanya ini dijalankan, isi *Program Counter* bisa pula berubah karena pengaruh perangkat keras, yaitu saat mikrokontroler di-*reset* atau menerima sinyal interupsi dari perangkat *input/output*. Dengan demikian, maka aliran program MCS51 memiliki beberapa kelompok instruksi, yaitu:

#### 1. Kelompok Instruksi JUMP

Mikrokontroler menjalankan instruksi demi instruksi, selesai menjalankan satu instruksi mikrokontroler langsung menjalankan instruksi berikutnya, hal ini dilakukan dengan cara menaikkan isi *Program Counter*, dengan demikian pada saat instruksi bersangkutan dijalankan Program Counter selalu menyimpan lokasi dari

memori program yang menyimpan instruksi berikutnya. Mikrokontroler MCS51 mempunyai 2 macam instruksi JUMP, yaitu:

- a) LJMP (*Long Jump*)
- b) AJMP (*Absolute Jump*)
- c) SJMP (*Short Jump*)

Kerja dari ketiga instruksi ini persis sama, yakni memberi nilai baru pada Program Counter, kecepatan melaksanakan ketiga instruksi ini juga persis sama, yakni 2 periode instruksi, yang berbeda adalah dalam ukuran instruksinya, instruksi LJMP ukurannya 3 *byte*, sedangkan instruksi AJMP dan SJMP hanya 2 *byte*.

## 2. Kelompok Instruksi Sub-Rutin

Instruksi-instruksi yang melibatkan penggunaan sub-rutin program. Selain melibatkan Program Counter, juga melibatkan Stack yang diatur oleh penunjuk Register Stack. Instruksi ACALL dipakai untuk me-‘manggil’ program sub-rutin dalam daerah memori program 2 KiloByte yang sama, setara dengan instruksi AJMP yang sudah dibahas sebelumnya. Sedangkan instruksi LCALL setara dengan instruksi LJMP, yang bisa menjangkau seluruh memori program mikrokontroler MCS51, sebanyak 64 KiloByte (tapi tidak ada instruksi SCALL yang setara dengan LJMP). Program untuk AT89C2051 tidak perlu memakai instruksi LCALL.

## 3. Kelompok Instruksi Lompatan Bersyarat

Instruksi *Jump* bersyarat merupakan instruksi inti bagi mikrokontroler tanpa kelompok instruksi ini program yang ditulis tidak banyak berarti. Instruksi-instruksi ini selain melibatkan *Program Counter*, juga melibatkan kondisi-kondisi tertentu yang biasanya dicatat dalam bit-bit khusus yang terhimpun dalam register tertentu. Instruksi-instruksi yang termasuk dalam kelompok instruksi lompatan bersyarat, yaitu:

- a) Instruksi JZ (*Jump if Zero*) dan JNZ (*Jump if Not Zero*)
- b) Instruksi JC (*Jump on Carry*) dan JNC (*Jump on Not Zero*)

- c) Instruksi JB (*Jump in Bit Set*), JNB (*Jump on Not Bit Set*), dan JBC (*Jump on Bit Set then Clear Bit*)

## 4. Kelompok Instruksi Proses dan Tes

Instruksi-instruksi *Jump* bersyarat: Kedua instruksi yang dimaksud adalah DJNZ dan instruksi CJNE.

### a) Instruksi DJNZ

Instruksi DJNZ (*Decrement and Jump if Not Zero*) merupakan instruksi yang akan mengurangi 1 nilai register serbaguna (R0 ..R7) atau memori-data dan akan lompat ke memori program yang dituju jika ternyata setelah pengurangan 1 tersebut hasilnya nol.

### b) Instruksi CJNE

Instruksi CJNE (*Compare and Jump if Not Equal*) membandingkan dua nilai yang disebut dan MCS akan lompat ke memori program yang dituju kalau kedua nilai tersebut tidak sama.

Bahasa Assembly menggantikan kode-kode biner dari bahasa mesin dengan “Mnemonic” yang mudah diingat, misalnya sebuah instruksi penambahan dalam bahasa mesin disajikan dengan kode “10110011” yang dalam bahasa *Assembly* dapat disajikan dalam bentuk *Mnemonic ADD*, sehingga mudah diingat dibanding sederetan angka nol dan satu.

Format Program Bahasa Assembly dalam Program bahasa *Assembly* berisikan:

- a) Instruksi-instruksi mesin;
- b) Pengarah-pengarah *Assembler*;
- c) Kontrol-kontrol *Assembler*;
- d) Komentar-komentar

Instruksi-instruksi merupakan mnemonic yang menyatakan suatu instruksi yang bisa dijalankan (misalnya MOV). Pengarah *Assembler* (*Assembler Directive*) merupakan instruksi ke program *Assembler* yang didefinisikan struktur program, simbol-simbol, data, konstanta, dan lain-lain (misalnya ORG). Baris-baris program yang mengandung instruksi mesin atau pengarah *Assembler* harus mengikuti aturan program *Assembler* ASM51. Masing-masing baris atas beberapa field yang dipisahkan dengan spasi tabulasi.

[label]: mnemonic [operasi] [, operan] [...] [; Komentar]

## 1. Label

Sebuah label mewakili suatu alamat dari instruksi atau data yang mengikat. Label ini digunakan sebagai operan pada instruksi percabangan (misal: SJMP TERUS). Simbol dan label dua hal yang berbeda. Simbol tidak menggunakan titik dua, sedangkan label menggunakan titik dua.

## 2. Mnemonik

Mnemonik instruksi atau pengarah *Assembler* dimasukkan dalam “*Mnemonic Field*” yang mengikuti “*Label Mnemonic*”. Mnemonik instruksi misalnya: ADD, MOV, INC, dan lain-lain. Sedangkan pengarah *Assembler* misalnya ORG, EQU, DB, dan lain-lain.

## 3. Operan

Operan ditulis setelah mnemonik, bisa berupa alamat atau data yang digunakan instruksi yang bersangkutan. Bisa juga berupa label yang digunakan instruksi yang bersangkutan. Bisa juga berupa label yang mewakili alamat suatu data atau berupa simbol yang mewakili suatu data konstanta.

## 4. Komentar

Komentar harus diawali dengan titik koma (;). Sebuah baris atau bagian dari suatu baris akan dianggap sebagai komentar jika diawali dengan titik koma. Sub-rutin dari bagian-bagian besar program yang mengerjakan suatu operan biasanya diawali dengan blok komentar yang menjelaskan fungsi sub-rutin atau bagian besar program tersebut.

## 5. Simbol Assembler Khusus

Simbol-simbol *Assembler* khusus digunakan untuk mode-mode pengalamatan melalui register. Simbol-simbol ini mencakup A, R0 sampai dengan R7, DPTR, C, PC, dan AB. Juga tanda dolar (\$) yang dapat digunakan untuk menunjuk nilai pencacah program (*program counter*) saat itu.

## 6. Pengalamatan Tidak Langsung

Beberapa instruksi menggunakan operan berupa register yang menyimpan alamat data disimpan. Dalam hal ini digunakan tanda “at” (@) yang dapat digunakan bersama dengan R0, R1, DPTR, atau PC tergantung dari instruksi yang digunakan.

## 7. Data Langsung

Data-data langsung diawali dengan tanda *pound* (#) dan menyatu dengan instruksi yang bersangkutan. Semua operasi yang melibatkan data langsung (kecuali instruksi MOV, DPTR, #Data) hanya membutuhkan data 8-bit (1 *byte*). Data langsung akan dievaluasi sebagai suatu konstanta 16-bit dan *byte*-rendah yang digunakan. Semua bit di-*byte* tinggi harus sama (00h atau FFh) atau nantinya akan mengakibatkan kesalahan “*value will not fit in byte*”.

## 8. Alamat Data

Banyak instruksi yang mengakses lokasi-lokasi memori menggunakan pengalamatan langsung dan membutuhkan alamat memori data internal (00h hingga 7Fh) atau alamat SFR (80h hingga FFh) pada operasi.

## 9. Alamat Bit

Salah satu kelebihan mikrokontroler 51 adalah kemampuannya bisa mengakses alamat-alamat per-bit tanpa menggunakan cara khusus. Instruksi yang melibatkan lokasi-lokasi yang teralamat bit (*bit addressable*) harus menyediakan suatu alamat bit memori data internal (00h dan 7Fh) atau alamat bit di ruang SFR (80h hingga FFh). Cara menuliskannya ada tiga cara, yaitu:

Secara eksplisit menggunakan alamatnya langsung (misal: **SETB 0Eh**)

Menggunakan tanda titik (.) antara alamat *byte* dan posisi bit (misal: **SETB ACC.7**).

Menggunakan simbol yang baku (misal: **JNB TI, \$**).

## 10. Alamat Kode

Suatu alamat kode digunakan dalam operan instruksi lompatan, termasuk lompatan relatif (SJMP dan lompatan bersyarat), lompatan dan CALL absolut (ACALL dan AJMP) dan lompatan *Call Far* (LCALL dan LJMP). Alamat halaman 11 atau alamat panjang 16 bit sesuai dengan kasusnya.

## 11. Jump Call Umum

ASM51 membolehkan kita untuk menggunakan mnemonik JMP atau CALL yang umum; *mnemonic* JMP digunakan sebagai wakil dari SJMP, AJMP atau LJMP, sedangkan mnemonik CALL mewakili ACALL atau LCALL. *Assembler* akan mengkonversi mnemonik umum ini

menjadi instruksi yang sesungguhnya mengikuti beberapa aturan sederhana, yaitu:

Diubah ke SJMP jika ada dalam acuan alamat di depan (tujuan lompatan sebelum instruksi JMP yang bersangkutan) dan jangkauan (lompatan berada dalam 128 *byte* (lokasi)).

Diubah ke bentuk AJMP jika tidak ada acuan lompatan di depan dan tujuan lompatan masih di dalam blok 2K yang sama.

Jika aturan a) dan b) tidak dipenuhi maka akan diubah ke bentuk LJMP.

## 12. Pengarah Assembler

Pengarah *Assembler* merupakan instruksi pada program *Assembler* itu sendiri, bukan sebagai instruksi bahasa *Assembly* yang akan dijalankan mikrokontroler yang bersangkutan. ASM51 menyediakan beberapa pengarah *Assembler* yang dikelompokkan sebagai berikut:

- Kontrol kondisi *Assembler* (ORG, END, dan USING);
- Definisi simbol (SEGMENT, EQU, SET, DATA, IDATA, XDATA, BIT, dan CODE);
- Pemesanan inisialiasi penyimpanan (DS, DBIT, DB, dan DW);
- Rantai (*Linkage*) program (PUBLIC, EXTRN, dan NAME); dan
- Pemilihan segmen (RGES, CSEG, DSEG, ISEG, BSEG, XSEG)

## Operasi Gabungan

Saat mengembangkan suatu program aplikasi yang besar, biasanya program aplikasi yang subprogram atau modul yang berisikan bagian-bagian kode (biasanya berupa sub-rutin). Strategi pemrograman ini dinamakan sebagai "Pemrograman Modular". Umumnya, modul-modul tersebut bersifat dapat direlokasikan (*relocatable*), artinya tidak ditempatkan pada lokasi-lokasi tertentu dalam ruang kode maupun data. Sebuah proses penggabungan harus dilakukan untuk menggabungkan (*link*) modul-modul tersebut menjadi sebuah modul absolut yang bisa dijalankan.

Dalam pembuatan program aplikasi pada VB 6.0, langkah pertama adalah

dengan membuat sebuah project. Adapun caranya dapat dilakukan dengan beberapa cara, diantaranya mengklik **Start | program | Microsoft Visual Basic 6.0 | Microsoft Visual Basic 6.0**. Cara lainnya adalah dengan membuka Windows Explorer dengan mengklik kanan **Start** dan mengklik **Explore**. Setelah explorer muncul klik akses program files\ Microsoft Visual Studio\ VB98 dan klik ganda vb6.exe. Selanjutnya akan terlihat tampilan seperti pada gambar 2.6 (*lihat lampiran*).

Kemudian pilih Standard EXE lalu klik open maka akan terlihat tampilan form project seperti pada Gambar 2.7 (*lihat lampiran*).

### 1) Menu Bar

Menu bar merupakan salah satu fasilitas yang dapat digunakan untuk membantu user dalam membuat program aplikasi pada Visual Basic. Ada tiga belas menu yang dapat digunakan dan masing-masing mempunyai fungsi yang berbeda. Untuk menggunakan fasilitas menu, klik menu yang dipilih dan selanjutnya mengklik submenu yang akan digunakan seperti pada Gambar 2.8 (*lihat lampiran*).

### 2) Toolbar

Toolbar berfungsi sama dengan menu, hanya saja berbeda tampilan. Pada toolbar cukup mengklik icon yang ingin digunakan yang terdapat pada toolbar. Jumlah icon pada toolbar dapat diatur dengan mengklik Menu View | toolbars. Selanjutnya ada pilihan menambah toolbar, diantaranya Debug, Edit, Form editor, Standart, dan Customize. Pada submenu Customize terdapat pilihan untuk mengatur toolbar yang akan digunakan. Tampilan toolbar terlihat seperti Gambar 2.9 (*lihat lampiran*).

### 3) Toolbox

Toolbox merupakan tempat icon – icon untuk objek yang akan dimasukkan dalam form pada pembuatan program aplikasi. Secara default pada toolbox hanya terdapat objek - objek seperti Gambar 2.10 (*lihat lampiran*).

Secara garis besar fungsi dari masing-masing control tersebut adalah sebagai berikut:

**a. Pointer**

*Pointer* bukan merupakan suatu kontrol tapi *icon* ini digunakan ketika memilih kontrol yang sudah berada pada *form*.

**b. Label**

*Label* adalah kontrol yang digunakan untuk menampilkan *text*, yang tidak dapat diperbaharui.

**c. Frame**

*Frame* adalah kontrol yang digunakan untuk mengidentifikasi sebuah *group* pengontrolan.

**d. Check Box**

*Check Box* adalah kontrol yang digunakan untuk memilih satu atau beberapa *check Box* secara bersamaan.

**e. Combo Box**

*Combo Box* adalah kontrol yang digunakan untuk mengetikkan pilihan atau untuk memilih item lewat *Drop-Down List*.

**f. Horizontal Scroll Bar**

*Horizontal Scroll Bar* adalah kontrol yang digunakan untuk memungkinkan pemakai untuk memilih suatu objek selama dalam jangkauan *Horizontal Objek*.

**g. Timer**

*Timer* adalah kontrol yang digunakan untuk mengoperasikan waktu kejadian pada rutin program termasuk internal waktu.

**h. Directory List Box**

*Directory List Box* adalah kontrol yang digunakan untuk menampilkan daftar *directory* pada *drive* terpilih sehingga dapat dipilih sebuah *Directory* dan *path*.

**i. Shape**

*Shape* adalah kontrol yang digunakan untuk membentuk objek dua dimensi, bujur sangkar, lingkaran, empat persegi panjang, ellips.

**j. Image**

*Image* adalah kontrol yang digunakan untuk menampilkan gambar *bitmaps*, *windows*, *metafile*, dan *icon*.

**k. OLE**

*OLE* adalah kontrol yang digunakan untuk memungkinkan pemakai untuk menempelkan suatu objek dari aplikasi

visual basic ke aplikasi yang mendukung *OLE*.

**l. Data List dan Data Combo**

*Data List* dan *Data Combo* adalah kontrol yang digunakan untuk menampilkan data dalam sebuah *Drop-Down List Box*

**m. SSTab**

*SSTab* adalah objek berbentuk lembaran-lembaran, setiap lembaran berisi kelompok informasi.

**n. Picture Box**

*Picture Box* adalah kontrol yang akan menampilkan file gambar, dengan format *Bitmaps*, *icon*, *Gif*, *Jpeg*, dan sebagainya

**o. TextBox**

*TextBox* adalah kontrol yang digunakan untuk menempatkan teks dalam form dan pemakai dapat mengedit teks tersebut.

**p. Command Button**

*Command Button* adalah kontrol yang digunakan untuk memilih satu atau beberapa *check Box* secara bersamaan.

**q. Option Button**

*Option Button* sering digunakan lebih dari satu sebagai pilihan terhadap beberapa *option* yang hanya dapat dipilih satu.

**r. List Box**

*List Box* mengandung sejumlah item dan pemakai dapat memilih lebih dari satu

**s. Vertikal Scroll Bar**

*Vertical Scroll Bar* adalah kontrol yang memungkinkan pemakai untuk memilih suatu objek dan digunakan untuk membentuk *scroll bar* berdiri sendiri.

**t. Drive List Box**

*Drive List Box* digunakan untuk menampilkan daftar *drive* pada komputer pemakai dan memungkinkan untuk memilih sebuah *drive*.

**u. File List Box**

*File List Box* digunakan untuk menampilkan daftar file pada *directory* terpilih dan memungkinkan untuk memilih sebuah *drive*.

**v. line**

*Line* adalah kontrol yang memungkinkan pemakai untuk membuat garis lurus.

**w. Data dan Adodc**

Data dan *Adodc* digunakan untuk

menampilkan database pada suatu *form*.

#### x. Data Grid

*Data Grid* digunakan untuk menampilkan data dalam bentuk *grid* seperti di *worksheet excel*.

#### 4) Project Explorer

Project Explorer merupakan tempat untuk melihat daftar form, modules, dan design dengan mengklik kanan pada bagian project explorer dan pilih add, lalu pilih yang akan ditambah, seperti Gambar 2.12 (*lihat lampiran*).

#### 5) Properties Windows

Properties Windows merupakan tempat yang digunakan untuk mengatur properti dari setiap objek kontrol. Pada properti windows ini semua objek control dapat diatur sesuai dengan program aplikasi yang akan dibuat. Tampilan properties tampak seperti gambar 2.13 (*lihat lampiran*).

#### 6) Form Layout Windows

Form layout windows merupakan tempat untuk melihat posisi tampilan form saat dieksekusi atau program dijalankan. Untuk mengubah posisi tampilan saat dijalankan, klik pada form layout window dan atur sesuai dengan keinginan. Tampilan form layout windows seperti gambar 2.14 (*lihat lampiran*).

#### 7) Form Objek

Kontrol – kontrol pada toolbox akan diletakkan disini sesuai dengan rancangan program aplikasi. Untuk menampilkan form objek ini, klik ganda pada icon / nama form pada jendela project explorer atau dengan klik kanan pada icon / nama pilih **view object**. Contoh tampilan form seperti gambar 2.15 (*lihat lampiran*).

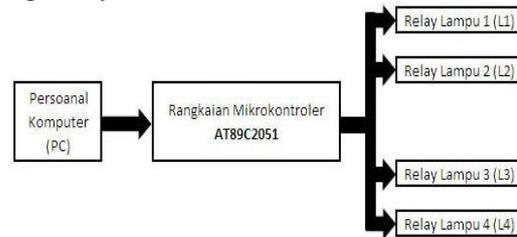
#### 8) Form Kode

Form kode merupakan tempat untuk menulis kode – kode atau syntax program aplikasi Visual Basic yang didalamnya dapat memanipulasi, mengatur dan memberikan perintah-perintah terhadap objek-objek yang kita buat. Untuk menampilkan form kode ini, klik form pada project explorer, klik kanan pilih **View Code**.

#### 1. Metode Penelitian

Sebagai proses awal perancangan blok diagram di bawah ini akan sangat membantu

untuk memberikan rancangan hardware secara keseluruhan. Berikut adalah gambar blok diagramnya:



Gambar 6. Diagram Blok Perancangan Hardware

Berdasarkan gambar di atas, Personal Komputer berfungsi sebagai interface berupa gambar simulator yang dibuat dengan menggunakan pemrograman visual basic, dan juga sebagai pusat pengendali untuk lampu L1.sampai dengan lampu L4 yang di hubungkan dengan relay dan sumber tegangan AC 220 Volt.

Mikrokontroler berfungsi sebagai pemberi tegangan 12 Volt, untuk mengaktifkan relay sebagai pemutus / penyambung antara lampu dan sumber tegangan. Adapun cara kerja mikrokontoler tersebut adalah: saat port yang terhubung ke relay diberikan perintah untuk ON melalui PC maka tegangan port berubah menjadi 5 Volt DC, berlogik 1. Sebaliknya, saat PC memberikan perintah untuk OFF maka tegangan port berubah menjadi 0 V, berlogik 0

Berikut adalah instruksi-instruksi pengendalian:

1. Instalasi Listrik Gedung Bertingkat
2. Sistem Kendali, Diagram Blok Sistem Pengendalian
3. Mikrokontroler Sebagai Pengendali
4. Program Visual Basic

Sebagian besar komponen - komponen yang telah dijelaskan pada bab II akan dirangkai dan diletakkan di atas sebuah papan triplek dengan ukuran 29 x 16 cm.



Gambar 7. Simulator

Spesifikasi Sistem Pengendali:

- ✓ Input Tegangan : 220 Volt AC, 5 Volt DC
- ✓ Dimensi Sistem Pengendali: panjang 29 cm, lebar 16 cm

Simulator ini dapat dibagi atas 4 bagian penting, adapun bagian itu adalah:

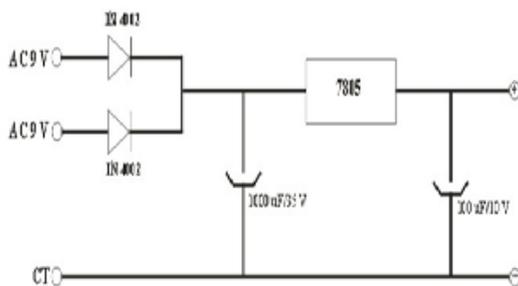
- a. Rangkaian Catu daya
- b. Rangkaian Input
- c. Rangkaian Proses
  - c.1 Rangkaian Komunikasi Serial
  - c.2 Rangkaian Mikrokontroler
- d. Rangkaian Output ( Rangkaian penggerak / driver )

### Rangkaian Catu daya

Rangkaian ini memiliki fungsi sebagai penyediaan tegangan untuk rangkaian mikrokontroler dan rangkaian penggerak (*driver*). Besarnya tegangan yang dibutuhkan rangkaian mikrokontroler adalah +5 Volt DC dan rangkaian penggerak adalah +12 Volt DC, dan semua ini akan dihasilkan oleh rangkaian catu daya.

Rangkaian catu daya ini bekerja dengan tegangan input 220 V AC, yang diturunkan tegangannya menggunakan trafo *step-down*, kemudian disearahkan menggunakan dioda penyearah. Untuk tegangan +5 V DC, rangkaian ini menggunakan regulator LM7805 dan untuk tegangan +12 V DC menggunakan dioda penyearah.

Berikut ini adalah gambar rangkaiannya:

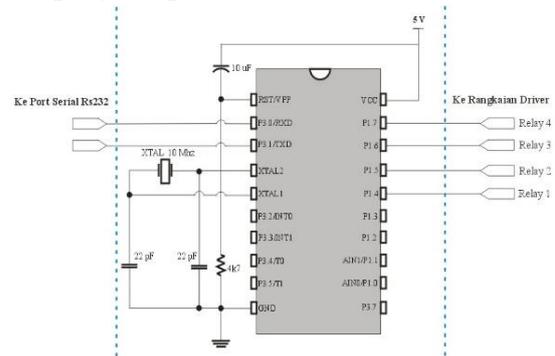


Gambar 8. Rangkaian Catu Daya

### 3.2 Rangkaian Mikrokontroler

Rangkaian mikrokontroler ini berfungsi sebagai pengantara antara PC dan

rangkaian driver. Rangkaian ini akan membaca perintah dari PC melalui rangkaian serial yang terhubung melalui port P3.0 sebagai jalur RX penerima perintah, dan P3.1 sebagai sebagai TX pengirim perintah.



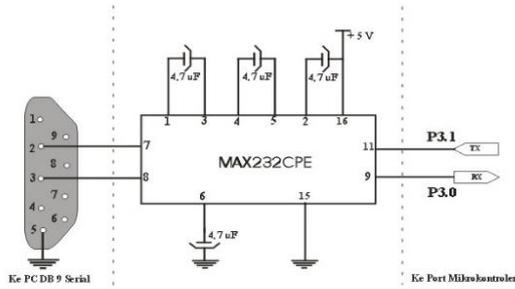
Gambar 9. Rangkaian Mikrokontroler

Kemudian perintah itu akan dipakai menjalankan driver berupa relay yang terhubung melalui port P1.4 – P1.7. Adapun alokasi port-nya adalah sebagai berikut, port P1.4 untuk relay 1, port P1.5 untuk relay 2, port P1.6 untuk relay 3, port P1.7 untuk relay 4.

### Rangkaian Komunikasi Serial

Rangkaian ini berfungsi dengan komponen utamanya berupa IC TTL RS-232 yang terdiri dari 16 port kaki. Port 9 IC TTL terhubung ke mikrokontroler melalui kaki port 2, yaitu P3.0 dan port 10 IC TTL ke port 3 mikrokontroler, yaitu P3.1. Kemudian port 7 terhubung ke pin 2 DB 9 serial, dan port 8 terhubung ke pin 3 DB 9 serial.

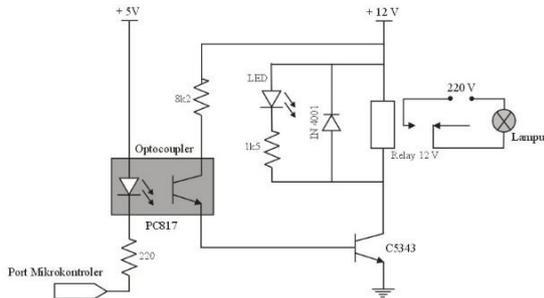
Adapun fungsi utama dari rangkaian ini adalah mengubah tegangan TTL ke serial. Besar tegangan output chip RS-232 adalah +/- 12 Volt, dalam hal ini rangkaian akan membaca tegangan + 12 Volt sebagai sebagai logik 0 dan tegangan - 12 Volt sebagai logik 1. Maka nilai logik tersebutlah yang akan dibaca mikrokontroler, dimana nilai dari logik 0 adalah 0 Volt, sedangkan nilai logik 1 bernilai 5 Volt. Untuk lebih jelas, berikut adalah gambar 3.5.



Gambar 10. Rangkaian Serial ke PC

**Rangkaian Penggerak ( driver )**

Komponen utama dari rangkaian ini berupa opto coupler yang berfungsi untuk memutus arus bila terjadi perbedaan tegangan, dalam hal ini fungsi mirip seperti kerja saklar. Kemudian pada rangkaian ini juga ada komponen relay yang berfungsi sebagai saklar untuk menghubungkan tegangan 220 Volt untuk menyalakan lampu.



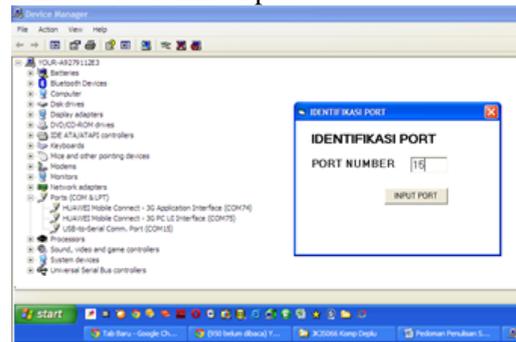
Gambar 11. Rangkaian Driver

Rangkaian ini akan bekerja pada saat mendapat input tegangan 0 Volt dan tegangan 5 Volt dari mikrokontroler, pada saat tegangan 5 Volt (berlogik 1) sakelar akan tertutup dan lampu akan menyala, dan pada saat tegangan 0 Volt (berlogik 0) sakelar terbuka dan lampu akan tertutup.

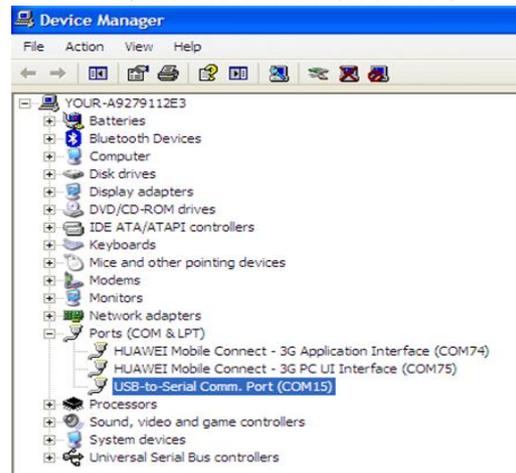
**HASIL DAN ANALISIS**

Program ini menggunakan visual basic versi 6.0, dan di jalankan di PC desktop atau laptop yang memiliki operasi sistem window xp. Sebagai langkah pertama, program yang telah dibuat dalam pemrograman visual basic ini akan selalu memunculkan tampilan window identifikasi port yang bertujuan menginialisasi alokasi port dari PC saat mikrokontroler dihubungkan lewat kabel serial. Setelah data port itu bersesuaian maka aplikasi baru mau dijalankan

Dibawah ini adalah gambar tampilan window identifikasi port:



Gambar 12. Identifikasi Port di PC



Gambar 13. Alokasi port PC

Simulator ini terhubung ke PC menggunakan kabel usb-serial, yaitu kabel yang mengkonversi serial to usb. Sehingga, pada saat simulator ini terhubung ke PC, maka perlu untuk selalu mengecek terlebih dahulu alokasi port yang tersedia. Kita dapat mengeceknya lewat control panel - device manager - port (COM &LPT) .

Setelah data port dimasukkan,maka akan muncul aplikasi sebagai berikut:



Gambar 14. Tampilan Simulator di PC

Keterangan gambar:

- ✓ Tombol On / Off Berfungsi untuk menghidup/memadamkan lampu
- ✓ Tanda kotak diatas gambar fitting/dudukan lampu adalah indikator pada PC untuk status lampu. Merah tandanya mati, dan hijau tandanya hidup

Berikut adalah data yang diambil sebagai pengujian alat telah bekerja sesuai dengan spesifikasi yang seharusnya. Dalam keadaan alat standby dan aktif besar tegangan pin DB 9 Serial adalah untuk penerimaan dan pengiriman data adalah sebagai berikut:

Tabel 2. Besar Tegangan Pin DB 9 Serial

Pin DB 9 Serial	Keterangan
Pin 2	4,9 V
Pin 3	2,6 mV
Pin 5	Ground

Pada rangkaian driver atau penggerak, relay yang berfungsi sebagai sakelar berfungsi sebagai pemutus dan penyambung. Besar tegangan yang diukur adalah semua kaki/port mikro kontroler yang terhubung ke relay.

Tabel 3. Besar tegangan port Mikrokontroler

Port Mikro kontroler	Logik	Relay	Tegangan	Lampu
P1.4	0	On	0 Volt	Nyala
	1	Off	4,98 V	Padam
P1.5	0	On	0	Nyala
	1	Off	4,97 V	Padam
P1.6	0	On	0	Nyala
	1	Off	4,96 V	Padam
P1.7	0	On	0	Nyala
	1	Off	4,98	Padam

Berdasarkan data tabel diatas maka dapat disimpulkan relay bekerja dengan keadaan normal terbuka (normally open).

Personal computer (PC) yang berperang sebagai user interface, dan pemberi input-an untuk mengaktifkan fungsi dari mikrokontroler harus bekerja bersesuaian dengan status dari indikator pada PC.



Gambar 15. Indikator pada PC saat Lampu 1: Nyala  
 ✓ Tampilan indikator aplikasi pada saat Tombol 2: ON ditekan



Gambar 16. Indikator pada PC saat Lampu 2: Nyala

Tampilan indikator aplikasi pada saat Tombol 3: ON ditekan



Gambar 17. Indikator pada PC saat Lampu 3: Nyala

✓ Tampilan indikator aplikasi pada saat Tombol 4: ON ditekan



Gambar 18. Indikator pada PC saat Lampu 4: Nyala

Agar terlihat lebih jelas dan ringkas, maka berikut ini adalah tabel saat tombol – tombol pada PC diaktifkan:

- ✓ Pengujian lampu I: Tombol 1 ditekan

Tabel 4. Pengujian Tombol Lampu 1 Di Tekan

Tombo l PC	Rela y 1	Rela y 2	Rela y 3	Rela y 4	Lampu
Tombo l 1	On				Lampu 1 Nyala
	Off				Lampu 1 Padam
Tombo l 2		-			-
		-			-
Tombo l 3			-		-
			-		-
Tombo l 4				-	-
				-	-

✓ Pengujian lampu II: Tombol 2 ditekan

Tabel 5. Pengujian Tombol Lampu 2 Di Tekan

Tombo l PC	Rela y 1	Rela y 2	Rela y 3	Rela y 4	Lampu
Tombo l 1	-				-
	-				-
Tombo l 2		On			Lampu 1 Nyala
		Off			Lampu 1 Padam
Tombo l 3			-		-
			-		-
Tombo l 4				-	-
				-	-

✓ Pengujian lampu III: Tombol 3 ditekan

Tabel 6. Pengujian Tombol Lampu 3 Di Tekan

Tombo l PC	Rela y 1	Rela y 2	Rela y 3	Rela y 4	Lampu
Tombo l 1	-				-
	-				-
Tombo l 2		-			-
		-			-
Tombo l 3			On		Lampu 3 Nyala
			Off		Lampu 3 Padam
Tombo l 4				-	-
				-	-

✓ Pengujian lampu IV: Tombol 2 ditekan

Tabel 7. Pengujian Tombol Lampu 4 Di Tekan

Tombo l PC	Rela y 1	Rela y 2	Rela y 3	Rela y 4	Lampu
Tombo l 1	On				-
	Off				-
Tombo l 2		On			-
		Off			-
Tombo l 3			On		-
			Off		-
Tombo l 4				On	Lampu 4 Nyala
				Off	Lampu 4 Padam

✓ Pengujian lampu V: Lampu dinyalakan secara bersamaan

Tabel 8. Pengujian Tombol Semua Lampu Di Tekan

Tombo l PC	Rela y 1	Rela y 2	Rela y 3	Rela y 4	Kondisi Lampu
Tombo l 1,2,3, 4	On	On	On	On	Lampu 1,2,3,4 Nyala
	Off	Off	Off	Off	Lampu 1,2,3,4 Padam

Dalam pengujian lampu pertama dan pengujian lampu kedua dapat dinyatakan bahwa alat dapat bekerja sesuai dengan yang diharapkan yaitu menyalakan lampu melalui aplikasi visual basic pada PC.

**Analisa**

Simulator ini bekerja dengan menggunakan program visual basic yang bekerja baik pada windows xp. Simulator ini akan berfungsi bila proses identifikasi port telah di masukkan. Jika sekiranya data port salah, atau tidak dimasukkan sama sekali, maka aplikasi tidak akan terbuka sama sekali.

Relay bekerja secara normal terbuka (normally open), dalam pengertian bahwa pada saat tegangan port kaki mikrokontroler sebesar 5 volt relay tidak

akan bekerja menghubungkan antara tegangan sumber dan beban / lampu. Justru pada saat tegangan port kaki mikrokontroler adalah nol, maka relay bekerja untuk menyalakan lampu.

Pada saat tombol ON dijalkan maka mikrokontroler akan bekerja mengeksekusi perintah tersebut, dan menunggu sampai ada perintah selanjutnya untuk OFF. Bahkan pada saat kabel serialnya pun terputus, perintah tombol ON akan terus mengaktifkan fungsi relay.

### **KESIMPULAN**

Berdasarkan hasil pengujian dan analisa yang dilakukan, dapat disimpulkan bahwa:

- a. Simulator pengendali lampu dengan menggunakan PC ini dapat berjalan, yakni pengguna dapat mengendalikan lampu untuk menyala dan padam melalui tampilan program visual basic pada PC.
- b. Simulator ini sangat mudah dikendalikan, karena semua lampu yang dikendalikan diwakili oleh tampilan tombol – tombol visual pada PC sehingga aman untuk digunakan. Dengan alat ini lampu – lampu yang dikendalikan terpusat pada satu titik sehingga mudah untuk mengetahui bila ada kerusakan.

### **DAFTAR PUSTAKA**

- [1] Tarek I. El-Shennawy, Abdel-Mon'em Moussa, Mahmoud A. El-Gammal and Amr Y. Abou-Ghazala, "A Dynamic Voltage Restorer for Voltage Sag Mitigation in a Refinery with Induction Motors Loads", American J. of Engineering and Applied Sciences 3 (1): 144-151, 2010
- [2] ([www.digitalengineeringlibrary.com](http://www.digitalengineeringlibrary.com))  
Copyright © 2004.
- [3] Colonel William T. McLyman, "Transformer And Inductor Design Handbook", Marcel Dekker, 2004