

Aplikasi Kontrol Switch Menggunakan *Silicon Controlled Rectifier* (SCR) Melalui Parallel Port

Dikdik Krisnandi

Pusat Penelitian Informatika Lembaga Ilmu Pengetahuan Indonesia

Jl. Cicitu (Komplek LIPI) No.21/154 Bandung

Email: dikdik@informatika.lipi.go.id

Abstrak

Telah dilakukan pembuatan suatu aplikasi kontrol switch menggunakan *Silicon Controlled Rectifier* (SCR) melalui parallel port. Menggunakan parallel port sebagai media antarmuka, SCR sebagai switch elektronis, optocoupler sebagai pengisolasi listrik dan bahasa pemrograman yang digunakan Ezy Pascal. Dilakukan pengujian pada bagian output parallel port, bagian tegangan input, penggerak SCR dan bagian beban. Sistem yang dibuat membuktikan bahwa media parallel port dapat berfungsi sebagai media pengontrol SCR untuk mengaktifkan atau menonaktifkan beban.

Kata kunci: kontrol, saklar, SCR, port paralel

1. Pendahuluan

Komputer, lebih tepatnya disebut *Personal Computer* (PC), memiliki banyak fungsi. Salah satu fungsinya adalah dapat digunakan untuk melakukan pengontrolan suatu sistem. Pada umumnya komputer memiliki beberapa port yang dapat digunakan untuk akuisisi. Port tersebut adalah serial port, parallel port dan *Universal Serial Bus* (USB) port. Dari port-port yang ada, pengguna komputer jarang sekali menggunakan parallel port. Sebagian besar pengguna komputer hanya menggunakan parallel port sebagai media untuk koneksi printer. Padahal, parallel port termasuk media yang handal jika digunakan pada suatu sistem kontrol. Keistimewaan parallel port adalah penggunaannya pada sistem pengiriman data digital dimana beberapa bit data dikirim sekaligus pada satu saat dengan menggunakan jalur terpisah ke tempat yang sama. Keunggulan lainnya adalah register yang dibutuhkan dalam berkomunikasi dengan prosesor juga relatif lebih sedikit dibandingkan jika menggunakan serial port.

Walaupun komputer yang ada sekarang ini lebih mengutamakan penggunaan USB

port sebagai media koneksinya, penggunaan parallel port justru dapat menjadi nilai tambah karena di negara kita masih banyak komputer-komputer lama yang memiliki parallel port pada bagian belakang panelnya. Hal ini yang seharusnya dimanfaatkan secara maksimal penggunaannya karena selain menghemat biaya karena tidak perlu membeli parallel port-nya lagi juga. Selain itu, parallel port memiliki kemudahan dari sisi pemrograman dan antarmuka dengan hardware sehingga dapat lebih dieksploitasi penggunaannya.

Bit-bit kontrol yang keluar melalui parallel port akan diteruskan kepada switch untuk dapat mengendalikan beban. Switch yang digunakan pada penelitian ini adalah *Silicon Controlled Rectifier* (SCR). SCR mampu berfungsi dan bertindak sebagai switch untuk arus yang besar sehingga komponen ini lebih banyak digunakan untuk aplikasi tegangan AC (*Alternating Current*). Hal tersebut yang mendasari pemilihan SCR untuk menggantikan transistor dan relay sebagai driver output.

Tujuan penelitian ini adalah membuat suatu aplikasi pengontrol beban dengan menggunakan *Silicon Controlled Rectifier* (SCR) sebagai switch melalui media parallel port yang ada pada komputer. Dalam

penelitian ini indikator beban yang digunakan berupa lampu. Tulisan ini merupakan laporan hasil penelitian yang telah direalisasikan dan dilakukan pengujiannya.

2. Teori Dasar

2.1 Parallel Port

Parallel port adalah salah satu jenis soket pada PC untuk berkomunikasi dengan peralatan luar. *Parallel port* biasanya terletak pada salah satu *port* yang terletak pada panel bagian belakang komputer. Arti istilah *parallel* yakni sistem pengiriman data digital, dimana beberapa bit data dikirim sekaligus pada satu saat dengan menggunakan jalur terpisah. *Parallel port* terdiri atas 4 jalur kontrol, 8 jalur status dan 8 jalur data.

Parallel port mempunyai 3 buah standar yakni SPP (*Standard Parallel Port*), EPP (*Enhanced Parallel Port*) dan ECP (*Extended Capability Port*). Tujuan standarisasi adalah untuk kompatibilitas antara satu *device* dengan lainnya. Pada PC, konektor yang biasa digunakan adalah konektor DB-25 yang mempunyai jumlah pin sebanyak 25 buah.

2.2 Register-Register *Parallel Port*

Semua data, kontrol dan status *parallel port* berhubungan dengan register-register yang ada di dalam komputer. Dengan mengakses langsung register-register tersebut, *input* dan *output parallel port* dapat diatur. Register-register pada *parallel port* adalah:

1. Register data (*base + 0*)
Register data digunakan untuk mengeluarkan data pada jalur data *parallel port* melalui pin 2, 3, 4, 5, 6, 7, 8 dan 9. Register ini normalnya hanya sebagai *read only port*. Jika mencoba membaca dari *port* ini akan didapatkan *byte* terakhir yang terkirim. Jika *port* bersifat dua arah maka dapat juga menerima alamat ini.
2. Register status (*base + 1*)

Register status hanya sebagai *read only port*. Data apa saja yang dituliskan ke *port* ini akan diabaikan. *Port* status berasal dari lima masukan *parallel port* (pin 10, 11, 12, 13 dan 15), sebuah register status IRQ (*Interrupt Request*) dan dua bit sebagai cadangan.

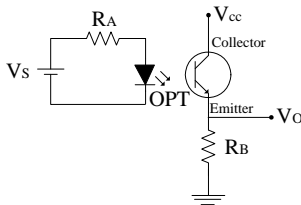
3. Register kontrol (*base + 2*)
Register kontrol hanya sebagai *write only port*. Saat sebuah *device* (dalam hal ini *printer*) disambungkan pada *parallel port* maka dibutuhkan 4 kontrol yakni *Strobe* (pin 1), *Auto Linefeed* (pin 14), *Initialize* (pin 16) dan *Select Printer* (pin 17).

Pada saat pertama kali PC dinyalakan, *Basic Input Output Service* (BIOS) akan menentukan jumlah *port* yang dimiliki. Selanjutnya akan diberi label *Line Printer Terminal* (LPT). Alamat dasar LPT adalah \$278 - \$27F, \$378 - \$37F dan \$3BC - \$3BF. Alamat ketiga register tersebut di atas dapat ditentukan dengan menjumlahkan alamat dasar *parallel port* dengan bilangan desimal tertentu. Misalnya jika ingin mengakses register data suatu *parallel port* LPT1, alamat register datanya sama dengan alamat dasar LPT yaitu \$378. Sedangkan alamat register status sama dengan alamat register dasar + 1 atau \$379 dan alamat register kontrolnya sama dengan alamat register dasar + 2 atau \$37A.

2.3 Optocoupler

Optocoupler merupakan piranti elektronika yang berfungsi sebagai pengisolasi listrik antara rangkaian *power* dengan rangkaian kontrol jika terjadi arus balik dari tegangan jala-jala PLN menuju komputer. *Opto* berarti optik dan *coupler* berarti penggandeng. *Optocoupler* dapat berupa *Light Emitting Diode* (LED) yang digandeng dengan foto transistor. Identy adalah sebagai berikut, perubahan tegangan V_s menghasilkan perubahan pada arus LED dimana perubahan arus akan masuk ke foto transistor. Di sisi lain hal tersebut akan menghasilkan perubahan tegangan pada terminal *Collector-Emitter*. Oleh sebab itu

tegangan sinyal akan digandeng dari rangkaian *input* ke rangkaian *output*. Gambar 1 memperlihatkan rangkaian *optocoupler*.



Gambar 1 Optocoupler dengan LED dan foto transistor

Tegangan *output* V_O bagian *optocoupler* dapat dihitung dengan persamaan:

$$V_O = V_{CC} - V_{CE} \quad (1)$$

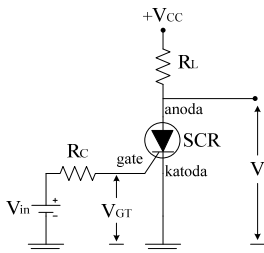
Dimana V_{CC} adalah tegangan catu untuk transistor dan V_{CE} adalah tegangan transistor dari *collector* ke *emitter*.

Sedangkan besarnya arus yang mengalir melalui *optocoupler* LED adalah:

$$I_{LED} = \frac{V_S - V_{LED}}{R_A} \quad (2)$$

2.4 Silicon Controlled Rectifier

Huruf *Silicon Controlled Rectifier* (SCR) adalah komponen yang terbuat dari bahan *semiconductor silicon* dan mempunyai fungsi sebagai pengendali ataupun sebagai *switch*. SCR juga sering disebut *thyristor*. Gambar 2 memperlihatkan rangkaian SCR dasar.



Gambar 2 Rangkaian Silicon Controlled Rectifier (SCR) dasar

Tegangan input (V_{in}) yang dibutuhkan untuk memicu (men-*trigger*) sebuah SCR adalah :

$$V_{in} = V_{GT} + I_{GT}R \quad (3)$$

dimana V_{GT} dan I_{GT} adalah tegangan dan arus pemicu *gate*.

Setelah SCR dalam kondisi ON maka SCR akan tetap ON meskipun V_{in} dikurangi menjadi nol. Satu-satunya cara untuk me-*reset* SCR adalah dengan mengurangi V_{CC} ke nilai yang lebih rendah. Karena arus *holding* (I_H) mengalir melalui hambatan beban (R_L) sehingga tegangan catu (V_{CC}) untuk kondisi OFF harus lebih kecil dari :

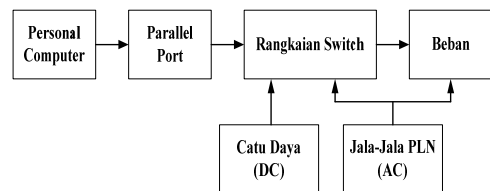
$$V_{CC} = 0,7 + (I_H R_L) \quad (4)$$

dimana I_H adalah arus *holding* yang mempertahankan SCR agar tetap ON.

3. Desain, Hasil dan Pembahasan

3.1 Desain

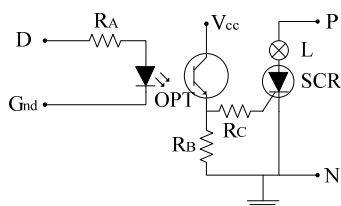
Secara keseluruhan rangkaian yang didesain sesuai dengan blok diagram seperti pada Gambar 3. Bagian *Personal Computer* (PC) merupakan pusat pengendali alat. Di dalam PC terdapat program yang dibuat menggunakan bahasa pemrograman Ezy Pascal dan berfungsi untuk mengaktifkan atau menonaktifkan beban. Dengan program yang dibuat, PC akan mengirimkan bit-bit kendali melalui *parallel port* kepada rangkaian *switch* sehingga beban dapat dikendalikan. Untuk memudahkan pengamatan maka pada bagian beban dipasang 8 buah lampu pijar, masing-masing lampu mempunyai daya 5 Watt dan tegangan 220 Volt, yang digunakan sebagai indikator beban.



Gambar 3 Blok diagram alat pengontrol beban

Keterangan Pada penelitian ini, standar *parallel port* yang digunakan adalah *Standard Printer Port* (SPP), konektor DB-25 dengan label LPT1 yang beralamat \$378 - \$37F. Register data yang beralamat \$378 digunakan untuk mengeluarkan data pada jalur data *parallel port* (pin 2 sampai dengan pin 9) sehingga dapat mengontrol 8 beban sekaligus. Indikator beban menggunakan 8 buah lampu pijar sebagai indikator berfungsi atau tidaknya aplikasi sistem yang dirancang. Catu daya yang digunakan adalah catu DC (*Direct Current*) sebesar 5 Volt digunakan untuk memberi catu kepada *optocoupler* dan kaki *gate* SCR sebagai *trigger* sedangkan tegangan AC (*Alternating Current*) jala-jala PLN 220 Volt digunakan untuk memberi catu ke lampu melalui anoda-katoda SCR.

Rangkaian *switch* terdiri atas *Silicon Controlled Rectifier* (SCR) dan *optocoupler*. SCR yang digunakan adalah jenis 2P4M dan tipe *optocoupler* yang dipakai adalah 817B1. Gambar 4 memperlihatkan rangkaian *switch* keseluruhan yang dibuat. Rangkaian *switch* akan menerima data input dari PC melalui *parallel port* sebanyak 8 saluran dari D₀ sampai D₇ dan masing-masing saluran akan mengeluarkan level tegangan 1 (*high*) atau level tegangan 0 (*low*).



Gambar 4 Rangkaian *switch* yang dibuat

Tegangan *output* bagian *optocoupler* dapat dihitung dari persamaan (1). Apabila D₀ bernilai 1 (*high*) maka LED pada *optocoupler* (OPT) akan menyala, sehingga transistor pada OPT akan saturasi. Hal ini akan menyebabkan arus mengalir dari V_{CC} ke *ground* melalui R_B, maka besarnya tegangan *output* v_o akan menjadi 5 Volt. Apabila D₀ bernilai 0 (*low*) maka LED pada

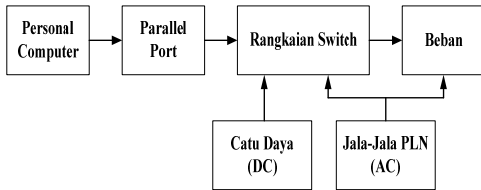
OPT tidak akan menyala, sehingga transistor pada OPT dalam keadaan *cut off*. Hal tersebut menyebabkan tidak ada arus yang mengalir dari V_{CC} ke *ground*. Pada keadaan ini V_{CC} akan mempunyai tegangan sebesar 0 Volt.

Tegangan *output parallel port* pada saat *high* adalah sebesar 5 Volt, tegangan LED (V_{LED}) adalah sebesar 2,5 Volt dan resistor V_{LEI} bernilai 100 Ω, maka dari persamaan (2) akan diketahui arus yang mengalir melalui LED *optocoupler* adalah sebesar 0,025 Ampere.

Arus dari jala-jala PLN akan mengalir apabila kaki *gate* SCR diberi *trigger* yang berasal dari V_{LEI}. Tegangan V_{LEI} merupakan tegangan yang berasal dari *output optocoupler*. Dengan V_{LED} adalah tegangan *gate trigger* yang besarnya 0,8 Volt dan V_{LEI} adalah arus *gate trigger* yang besarnya 200 μ Ampere maka dari persamaan (3) besarnya tegangan input V_{LEI} yang dibutuhkan untuk men-*trigger* SCR adalah sebesar 1 Volt

Setelah SCR ON maka SCR akan tetap hidup meskipun tegangan V_{CC} dikurangi menjadi nol. Untuk me-*reset* SCR adalah dengan cara mengurangi arusnya sehingga lebih kecil dari pada arus *holding* V_{CC} yang biasanya dilakukan dengan mengurangi tegangan jala-jala ke nilai yang rendah. Besarnya arus *holding* adalah 3 mA dan V_{CC} sebesar 9.680 Ω. Dengan demikian besarnya tegangan jala-jala yang dibutuhkan untuk membuat SCR OFF sesuai dengan persamaan (4) adalah 29,74 Volt. Tegangan jala-jala PLN sendiri berupa arus bolak-balik Secara keseluruhan rangkaian yang didesain sesuai dengan blok diagram seperti pada Gambar 3. Bagian *Personal Computer* (PC) merupakan pusat pengendali alat. Di dalam PC terdapat program yang dibuat menggunakan bahasa pemrograman Ezy Pascal dan berfungsi untuk mengaktifkan atau menonaktifkan beban. Dengan program yang dibuat, PC akan mengirimkan bit-bit kendali melalui *parallel port* kepada

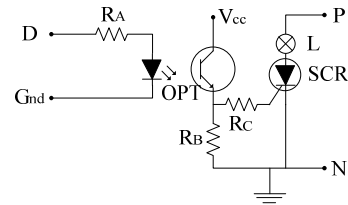
rangkaian *switch* sehingga beban dapat dikendalikan. Untuk memudahkan pengamatan maka pada bagian beban dipasang 8 buah lampu pijar, masing-masing lampu mempunyai daya 5 Watt dan tegangan 220 Volt, yang digunakan sebagai indikator beban.



Gambar 3 Blok diagram alat pengontrol beban

Keterangan Pada penelitian ini, standar *parallel port* yang digunakan adalah *Standard Printer Port* (SPP), konektor DB-25 dengan label LPT1 yang beralamat \$378 - \$37F. Register data yang beralamat \$378 digunakan untuk mengeluarkan data pada jalur data *parallel port* (pin 2 sampai dengan pin 9) sehingga dapat mengontrol 8 beban sekaligus. Indikator beban menggunakan 8 buah lampu pijar sebagai indikator berfungsi atau tidaknya aplikasi sistem yang dirancang. Catu daya yang digunakan adalah catu DC (*Direct Current*) sebesar 5 Volt digunakan untuk memberi catu kepada *optocoupler* dan kaki *gate* SCR sebagai *trigger* sedangkan tegangan AC (*Alternating Current*) jala-jala PLN 220 Volt digunakan untuk memberi catu ke lampu melalui anoda-katoda SCR.

Rangkaian *switch* terdiri atas *Silicon Controlled Rectifier* (SCR) dan *optocoupler*. SCR yang digunakan adalah jenis 2P4M dan tipe *optocoupler* yang dipakai adalah 817B1. Gambar 4 memperlihatkan rangkaian *switch* keseluruhan yang dibuat. Rangkaian *switch* akan menerima data input dari PC melalui *parallel port* sebanyak 8 saluran dari D_0 sampai D_7 dan masing-masing saluran akan mengeluarkan level tegangan 1 (*high*) atau level tegangan 0 (*low*).



Gambar 4 Rangkaian *switch* yang dibuat

Tegangan *output* V_o bagian *optocoupler* dapat dihitung dari persamaan (1). Apabila D_0 bernilai 1 (*high*) maka LED pada *optocoupler* (OPT) akan menyala, sehingga transistor pada OPT akan saturasi. Hal ini akan menyebabkan arus mengalir dari V_{CC} ke *ground* melalui R_B , maka besarnya tegangan *output* V_o akan menjadi 5 Volt. Apabila D_0 bernilai 0 (*low*) maka LED pada OPT tidak akan menyala, sehingga transistor pada OPT dalam keadaan *cut off*. Hal tersebut menyebabkan tidak ada arus yang mengalir dari V_{CC} ke *ground*. Pada keadaan ini V_o akan mempunyai tegangan sebesar 0 Volt.

Tegangan *output parallel port* pada saat *high* adalah sebesar 5 Volt, tegangan LED (V_{LED}) adalah sebesar 2,5 Volt dan resistor R_A bernilai 100 Ω , maka dari persamaan (2) akan diketahui arus yang mengalir melalui LED *optocoupler* adalah sebesar 0,025 Ampere.

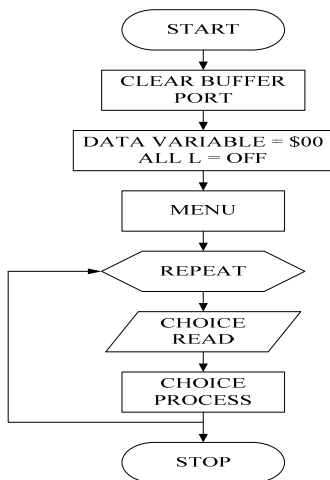
Arus dari jala-jala PLN akan mengalir apabila kaki *gate* SCR diberi *trigger* yang berasal dari V_{in} . Tegangan V_{in} merupakan tegangan yang berasal dari *output optocoupler*. Dengan V_{GT} adalah tegangan *gate trigger* yang besarnya 0,8 Volt dan I_{GT} adalah arus *gate trigger* yang besarnya 200 μ Ampere maka dari persamaan (3) besarnya tegangan input V_{in} yang dibutuhkan untuk men-*trigger* SCR adalah sebesar 1 Volt

Setelah SCR ON maka SCR akan tetap hidup meskipun tegangan V_{in} dikurangi menjadi nol. Untuk me-*reset* SCR adalah dengan cara mengurangi arusnya sehingga

lebih kecil dari pada arus *holding* I_H yang biasanya dilakukan dengan mengurangi tegangan jala-jala ke nilai yang rendah. Besarnya arus *holding* adalah 3 mA dan R_L sebesar 9.680Ω . Dengan demikian besarnya tegangan jala-jala yang dibutuhkan untuk membuat SCR OFF sesuai dengan persamaan (4) adalah 29,74 Volt. Tegangan jala-jala PLN sendiri berupa arus bolak-balik sehingga untuk membuat SCR OFF adalah dengan cara memberikan tegangan input V_{in} lebih kecil dari pada 1 Volt.

3.2 Flowchart Program

Program dimulai dari blok *start* yang artinya program dalam keadaan dieksekusi. Selanjutnya program akan menghapus semua data dalam *buffer parallel port* dengan cara memberikan nilai \$00 kepada *parallel port*. Keadaan ini akan mengakibatkan semua beban dalam keadaan OFF. Berikutnya adalah memberikan variabel data dengan nilai \$00 dan variabel L1 samapai dengan L2 dengan nilai OFF. Setelah semua data dalam kondisi awal selanjutnya program akan menampilkan menu pilihan. Menu pilihan dibuat untuk panduan dalam layar monitor sehingga proses untuk mematikan atau menghidupkan beban menjadi lebih mudah. Gambar 4 menunjukkan *flow chart* aplikasi program yang telah dilaksanakan.



Gambar 4. Flowchart program

3.3 Pengaktifan Bit-Bit Parallel Port

Pengendalian aktif atau non aktif beban pada rangkaian *switch* dilakukan oleh program yang terdapat pada PC. Program akan mengirimkan data yang akan mengaktifkan bit-bit pada register data *parallel port* dari D_0 sampai dengan D_7 . Menggunakan program yang dibuat dalam bahasa pemrograman Ezy Pascal, pengaktifan bit-bit register data dilakukan dengan perintah: `port[$378] := $nilai;`

`Port[$378]` adalah alamat untuk *parallel port* LPT1, sedangkan `$nilai` merupakan sebuah nilai dalam format heksadesimal. Tabel 1 menunjukkan hubungan antara beban yang akan diaktifkan dengan nilai heksadesimal yang akan diberikan untuk pengaktifan bit-bit *parallel port*.

Tabel 1. Pengaktifan bit-bit register data *parallel port*

Beban (lampu)	$D_7 : D_0$	Heksadesimal
1	0000:0001	01
2	0000:0010	02
3	0000:0100	04
4	0000:1000	08
5	0001:0000	10
6	0010:0000	20
7	0100:0000	40
8	1000:0000	80

Berdasarkan *flowchart* yang dibuat, langkah pertama adalah membuat kondisi beban dalam keadaan non aktif (*reset*) setelah menjalankan atau mematikan program. Bagian ini disebut bagian *clear buffer port*. Perintah proses menghapus *parallel port* adalah:

`port[$378] := data_awal;`

Ini berarti memberikan nilai \$00 kepada *parallel port* sehingga seluruh beban dalam keadaan non aktif.

3.4 Hasil Pengujian

Pengujian bagian *parallel port* dimaksudkan untuk mengetahui *output* tegangannya. Untuk mengetahui *output* tegangan tersebut dilakukan dengan cara mengeksekusi program dan kondisi alat

terpasang. Dari hasil pengukuran didapatkan bahwa pada saat diberi logika 1 (*high*) maka setiap *port* yang diukur ($D_0 - D_7$) menghasilkan tegangan sebesar 4,42 Volt. Sedangkan saat diberi logika 0 (*low*) maka setiap *port* yang diukur ($D_0 - D_7$) menunjukkan tegangan sebesar 0 Volt. Pengujian catu daya (DC) menunjukkan output tegangan sebesar 4,38 Volt. Pada rangkaian *switch* menggunakan SCR memiliki tegangan *input* sebagai penggerakannya, hasil pengukuran tegangan *input* menunjukkan bahwa pada saat diberi logika 1 (*high*), tegangan V_{in} pada setiap saluran *port* masing-masing akan menunjukkan tegangan sebesar 4,36 Volt. Sedangkan jika diberi logika 0 (*low*), tegangan V_{in} pada setiap saluran *port* masing-masing akan menunjukkan tegangan sebesar 0 Volt.

Untuk mengetahui program yang telah dibuat berfungsi sesuai dengan yang diharapkan maka dilakukan pengujian dalam 2 kondisi. Kondisi pertama adalah saat semua indikator beban (lampu) dalam keadaan OFF. Kondisi ini untuk menguji apakah pilihan untuk mengaktifkan beban dapat bekerja dengan baik. Pengertian baik di sini adalah bahwa indikator yang berupa lampu dapat menyala ketika mendapat perintah aktif. Hasilnya adalah semua beban dapat diaktifkan (ON). Kondisi kedua adalah saat semua indikator beban (lampu) dalam keadaan ON atau artinya dalam kondisi lampu menyala. Kondisi ini untuk menguji apakah pilihan untuk menonaktifkan beban, program dapat bekerja dengan baik yang ditunjukkan dengan indikator lampu yang menjadi mati. Hasil pengujian menunjukkan semua beban dapat dinonaktifkan (OFF) atau semua lampu dapat dimatikan

4. Kesimpulan

Berdasarkan hasil penelitian dan fakta yang diperoleh dari pengujian maka dapat disimpulkan bahwa aplikasi yang dibuat dapat melakukan fungsinya sebagai kontrol *switch* yang mampu mengendalikan beban dengan baik atau dengan kata lain indikator

beban yang berupa lampu dapat dinyalakan atau dimatikan sesuai dengan perintah program yang dibuat. Untuk pengembangan penelitian selanjutnya perlu dilakukan pengujian dengan meningkatkan panjang kabel hingga jarak jangkauan aplikasi maksimumnya dapat diketahui dan dicari solusi pemecahannya.

5. Daftar pustaka

- [1] Fitzgerald A.E, Higginbotham, David E, Grabel, Arvin., *Basic Electrical Engineering*. 5th ed, McGraw-Hill, Auckland, 1981.
- [2] Malvino, A.P, *Electronic Principles*, 5th ed, Glencoe, Macmillan, New York, 1993.
- [3] Thompson and Cooser P, *A Hardware Programming with Pascal*, Telemechanics, USA, 2003.
- [4] www.ezypascal.com