

# APLIKASI MIKROKONTROLER AT89S52 SEBAGAI PENGENDALI PERALATAN ELEKTRONIK RUMAH TANGGA MELALUI *REMOTE CONTROL*

Eka Wahyudi<sup>1</sup>, Gina Kuswiani<sup>2</sup>

<sup>1,2</sup>Program Studi Diploma III Teknik Telekomunikasi, Purwokerto

<sup>1</sup>ekawahyudi@akatelsp.ac.id

## ABSTRAK

Intel, Motorola dan Atmel merupakan nama-nama yang sudah tidak asing lagi sebagai pembuat mikrokontroler, komponen yang dikenal dengan keandalan dan kecepatan kejanya. Program-program mikrokontroler yang ditulis dengan Bahasa Assembler umumnya berukuran kecil dan dapat dieksekusi dengan cepat sehingga banyak digunakan terutama di bidang industri. Sebenarnya kita dapat menggunakan mikrokontroler dengan *remote control* sebagai pengendalinya dalam kehidupan sehari-hari sebagai sakelar lampu jarak jauh. Tempat seperti vila, area perkantoran dan bandara sering kali memiliki lampu dalam jumlah banyak. Dengan alat yang dirancang ini, kita tidak perlu lagi mematikan lampu-lampu dalam jumlah banyak tersebut secara manual. Penggunaan mikrokontroler akan diterapkan dalam suatu alat yang dapat diaplikasikan atau digunakan pemilik rumah untuk berinteraksi dengan peralatan yang ada di rumah baik dari jarak jauh dalam suatu ruangan maupun dari jarak dekat. Alat ini dapat dihubungkan dengan berbagai peralatan elektronik rumah tangga misalnya lampu penerangan, pompa air, pemanas air, kulkas, *air conditioner* (AC) dan peralatan rumah tangga lainnya. Program yang dibuat menggunakan bahasa assembler. Adapun alat pengendali ini terdiri dari *remote control* dalam hal ini menggunakan *remote VCD 3.0* dan *infra red receiver module* berbagai rangkaian antara lain rangkaian mikrokontroler, *driver transistor* dan *relay*.

**Kata kunci** : mikrokontroler, *remote control*, *infra red*.

## 1. Pendahuluan

Seiring dengan perkembangan zaman, teknologi pun semakin berkembang. Sehingga menyebabkan peralatan elektronik dalam suatu rumah semakin canggih, banyak dan beragam. Dengan banyaknya alat elektronik tersebut, maka seseorang akan mengalami kesulitan untuk mengendalikan peralatan elektronik yang ada di rumah. Misalkan saja bila seseorang memiliki rumah besar dengan terdapat banyak ruangan di dalamnya, tentu saja setiap hari harus menyalakan dan mematikan atau sebagai fungsi *on* dan *off* alat-alat elektronik tersebut. Sebagai contoh yaitu menyalakan beberapa lampu penerangan, *Air Conditioner* (AC), TV dan peralatan rumah tangga lainnya. Dari penjelasan di atas dapat disimpulkan bahwa agar lebih mempermudah seseorang dalam

mengendalikan alat-alat elektronik dalam suatu rumah maka perlu dibangun sebuah sentral untuk mengendalikan (menyalakan dan mematikan atau *on* dan *off*) alat-alat elektronik rumah tangga, dimana sistem dari sentral otomatis tersebut dapat dioperasikan dengan menggunakan perangkat yang didalamnya dikendalikan oleh suatu mikrokontroler dan dilengkapi dengan sebuah *remote control* dimana terdapat beberapa modul penerima (*receiver*) sensor dari *remote control* tersebut yang ditempatkan di beberapa ruangan. Peranan *remote control* ini sangat penting karena akan mempermudah seseorang untuk mengoperasikan alat-alat elektronik yang ada di rumahnya dari jarak jauh dalam satu ruangan tertentu.

## 1.1 Rumusan Masalah

Permasalahan utama yang akan dibahas adalah bagaimanakah cara membuat alat kendali dengan mikrokontroler AT89S52 yang dapat mengendalikan (menyalakan dan mematikan atau *on* dan *off*) peralatan elektronik rumah tangga melalui *remote control*?

## 1.2 Maksud dan Tujuan

Penulisan melakukan penelitian dengan tema aplikasi *remote control* ini dimaksudkan untuk dapat membuat alat kendali dengan mikrokontroler AT89S52 yang dapat mengendalikan (menyalakan dan mematikan atau *on* dan *off*) peralatan elektronik rumah tangga melalui *remote control*.

## 1.3 Manfaat Penulisan

Manfaat yang diperoleh dalam penelitian pada jurnal ini antara lain adalah :

1. Untuk mengembangkan ilmu pengetahuan baik untuk bahasa pemrograman Assembler maupun pada sistem mikrokontroler.
2. Dapat membantu bagi siapa saja yang ingin mempelajari lebih jauh tentang sistem mikrokontroler.
3. Dapat dijadikan bahan referensi bagi para perancang elektronika maupun penggemar mikrokontroler.

## 1.4 Tinjauan Pustaka/Dasar Teori

### 1.4.1 Arsitektur AT89S52

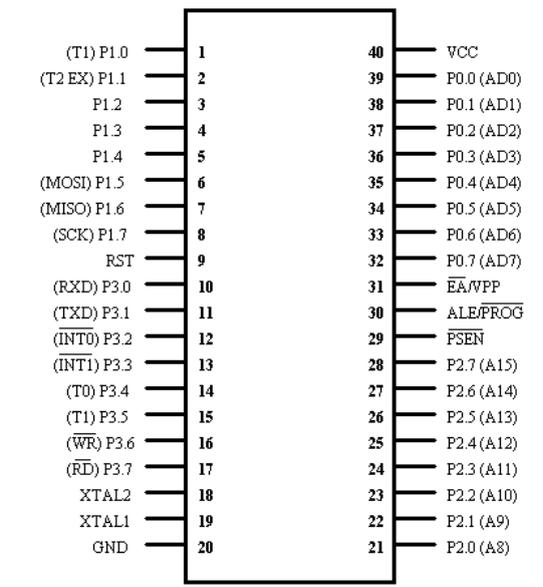
Mikrokontroler AT89S52 merupakan salah satu keluarga dari MCS-51 keluaran Atmel dengan 8K

*byte Flash PEROM (Programmable and Erasable Read Only Memory)*, AT89S52 merupakan memori dengan teknologi *nonvolatile memory*, isi memori tersebut dapat diisi ulang ataupun dihapus berkali-kali. Jenis mikrokontroler ini pada prinsipnya dapat digunakan untuk mengolah data per bit ataupun data 8 bit secara bersamaan.

Sebuah mikrokontroler dapat bekerja bila dalam mikrokontroler tersebut terdapat sebuah program yang berisi instruksi-instruksi yang akan digunakan untuk menjalankan sistem mikrokontroler tersebut. Instruksi-instruksi dari sebuah program pada tiap jenis mikrokontroler mempunyai beberapa perbedaan, misalkan saja pada mikrokontroler Atmel berbeda dengan instruksi pada mikrokontroler Motorola. Pada prinsipnya program pada mikrokontroler dijalankan secara bertahap, jadi pada program itu sendiri terdapat beberapa set instruksi dan tiap instruksi tersebut dijalankan secara bertahap atau berurutan.

Mikrokontroler AT89S52 memiliki 40 pin, 32 pin diantaranya untuk keperluan kanal paralel. Satu kanal paralel terdiri dari 8 pin, dengan demikian 32 pin tersebut membentuk 4 kanal paralel, yang masing-masing dikenal dengan Port 0, Port 1, Port 2 dan Port 3. nomor dari masing-masing jalur (pin) dari kanal paralel mulai dari 0 sampai 7, pin pertama kanal 0 disebut

dengan P0.0 dan jalur terakhir untuk kanal 3 adalah P3.7 adapun susunan pin-pin pada mikrokontroler AT89S52 ditunjukkan pada Gambar 1. beberapa pin pada mikrokontroler AT89S52 mempunyai lebih dari satu fungsi dan tidak semua fungsi dapat digunakan secara bersamaan, tergantung program dan aplikasi yang dibuat.



Gambar 1. IC AT89S52

### 1.4.2 Remote control

Pengendalian tanpa kabel dengan menggunakan infra merah (*infra red remote control*) ada beberapa jenis sistem pengkodean, dikarenakan tidak ada standar yang mengikat tiap produsen peralatan elektronik, misal Philips menggunakan standar RC5 dengan panjang kode 13 bit dan 14 bit dengan frekuensi carrier 36 Khz, Sony menggunakan kode 12 bit dengan frekuensi carrier 38 Khz atau 40 Khz. Dalam pembahasan ini *remote* yang digunakan adalah *remote* konvensional

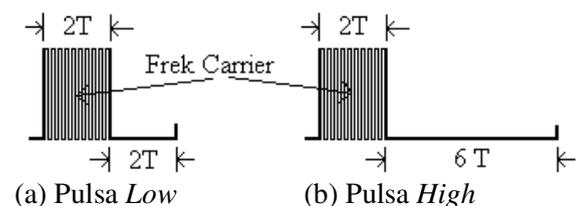
dengan standar produksi perusahaan NEC dengan panjang kode 32 bit dan 1 bit kode header serta frekuensi pembawa (carrier) 38 Khz. Sesuai dengan spesifikasi *remote control* dari NEC, frekuensi pembawa ( $F_c$ ) dan frekuensi pulsa ( $F_p$ ) sebagai penentu lebar periode untuk keadaan data *high* atau *low* dari data yang dipancarkan untuk frekuensi osilator ( $F_{osc}$ ), telah ditentukan sebagai berikut:

$$F_c = \frac{F_{osc}}{12} \quad \dots\dots(1)$$

$$F_p = \frac{F_{osc}}{192} \quad \dots\dots(2)$$

Sedangkan lebar pulsa carrier ( $T_c$ ) dari data *low* '0' dan data *high* '1' seperti pada Gambar 2. Dapat dihitung menggunakan persamaan:

$$T_c = \frac{1}{F_p} \quad \dots\dots(3)$$



Gambar 2. Lebar Pulsa pada *Remote control*

Untuk format kode dengan standar perusahaan NEC, urutan pulsa data serial adalah sebagai berikut:

H C0 C1 C2 C3 C4 C5 C6 C7 C'0 C'1  
C'2 C'3 C'4 C'5 C'6 C'7 D0 D1 D2  
D3 D4 D5 D6 D7 D'0 D'1 D'2 D'3  
D'4 D'5 D'6 D'7

dengan H = *Header pulse*  
C0 – C7 = *Custom bit code* (8 bit)

C'0 – C'7 = *Inversion Custom* (8 bit)  
D0 – D7 = *Data code* (8 bit)  
D'0 – D'7 = *Inversion Data code* (8 bit)

*Header* merupakan pulsa awal yang menandakan *start kode*, yaitu memberitahukan perangkat lain (*host*) untuk siap menerima kode yang akan dikirimkan oleh *remote control*. *Custom* adalah sebagai kode untuk membedakan jenis *remote control* yang dipakai (dalam standar perusahaan NEC), pada sebuah *remote control* mempunyai nilai *custom code* yang sama untuk setiap tombolnya, dan yang membedakan kode dari tiap tombol adalah pada kode datanya.

### 1.4.3 Perangkat Bahasa Assembler

Didalam bahasa pemrograman Assembler terdapat berbagai macam mode pengalamatan (*Addressing Mode*) antara lain:

a) Mode Pengalamatan Segera (*Immediate Addressing Mode*)

Metode ini menggunakan konstanta, misalnya: MOV A, #20h. data konstan merupakan data yang menyatu dengan instruksi yang harus diawali dengan tanda #, pada contoh diatas mempunyai arti bahwa data konstantanya adalah 20h yang disalin ke akumulator A.

b) Mode Pengalamatan Langsung (*Direct Addressing Mode*)

Metode ini dipakai untuk menunjuk data yang berada di suatu lokasi (alamat) memori tempat data berada.

Misalnya: MOV A, 30h. Instruksi ini mempunyai arti bahwa data yang berada di dalam memori dengan lokasi 30h disalin ke akumulator. Sekilas instruksi ini mirip dengan instruksi sebelumnya, tetapi terdapat perbedaan pada tanda '#', jika pada instruksi pertama menggunakan tanda '#' yang menandai 20h sebagai data konstan, sedangkan dalam instruksi kedua tidak menggunakan tanda '#' yang berarti bahwa alamat 30h diartikan sebagai suatu lokasi memori.

c) Mode Pengalamatan Tidak Langsung (*Indirect Addressing Mode*)

Cara ini dipakai untuk mengakses data yang berada didalam memori tetapi lokasi memori tersebut secara tidak langsung dititipkan pada *register* lain. Misalnya: MOV A, @R0. Dalam instruksi ini *register* serbaguna R0 digunakan untuk menyimpan lokasi memori, sehingga instruksi ini mempunyai arti memori yang alamatnya tersimpan pada *register* R0 yang kemudian isinya disalin ke Akumulator A. tanda '@' dipakai untuk menandai lokasi memori yang tersimpan dalam R0.

d) Mode Pengalamatan *Register* (*Register Addressing Mode*)

Cara ini menggunakan *register* serbaguna mulai dari R0 sampai dengan R7, sehingga aksesnya akan lebih cepat. Contoh: MOV A, R6. Dalam instruksi tersebut mempunyai arti bahwa data yang berada dalam

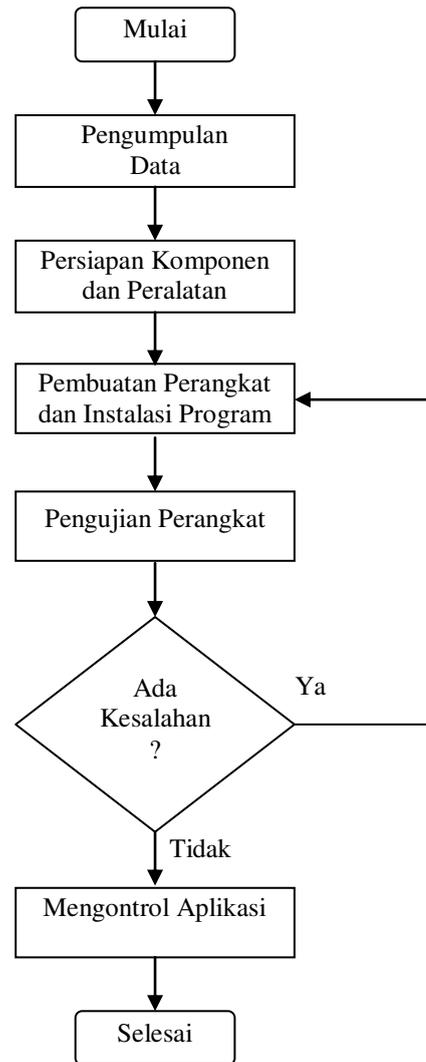
register R6 dipindahkan ke Akumulator R6.

e) Mode Pengalamatan Kode Tidak Langsung (*Code Indirect Addressing Mode*)

Metode ini digunakan untuk menyebutkan data yang tersimpan di dalam memori program yang dilakukan secara tidak langsung. Contoh: `MOVC A, @A+DPTR`. Pada instruksi ini perintah `MOV` diganti dengan perintah `MOVC`, instruksi `MOVC` digunakan untuk memori program. Tanda '@' digunakan untuk menandai `A+DPTR` yang berfungsi untuk menyatakan lokasi memori yang isinya disalin ke Akumulator A, dalam hal ini nilai yang tersimpan dalam `DPTR (Data Pointer Register 2 byte)` ditambah dengan nilai yang tersimpan dalam Akumulator A (1 byte) yang sama dengan lokasi memori program yang diakses.

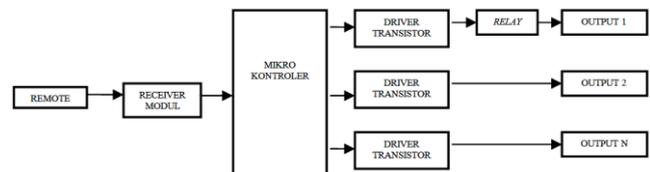
## 2. Metodologi Penelitian

Metodologi yang digunakan dalam penelitian ini adalah metode eksperimental yang bertujuan untuk mendapatkan rancangan alat yang akan dibuat dengan cara mencari, memodifikasi dan menguji rangkaian-rangkaian elektronika seperti yang ditunjukkan pada Gambar 3.



Gambar 3. Metodologi Penelitian

Gambaran umum mengenai proses pengendalian peralatan elektronik rumah tangga melalui *remote control* dapat dilihat pada blok diagram berikut ini:



Gambar 4. Diagram Blok Alat Pengendali

Adapun dalam perancangan alat pengendali peralatan elektronik berbasis mikrokontroler AT89S52 dibagi menjadi beberapa bagian utama, antara lain: Rangkaian Kendali, Mikrokontroler,

Osilator dan *Reset, Receiver Module* IRM8510

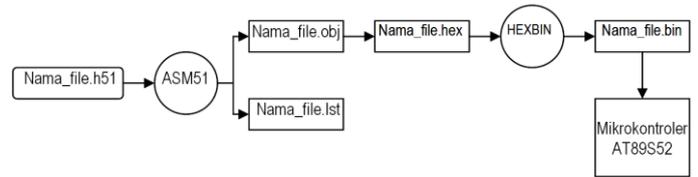
## 2.2 Perancangan Software

Untuk mengoperasikan alat yang telah dibuat dibutuhkan program dengan bahasa assembler yang telah dikompiler menjadi data *binary* yang kemudian diisikan pada mikrokontroler dari rangkaian pengontrol.

Adapun tiap tahap penyusunan perangkat lunak adalah sebagai berikut:

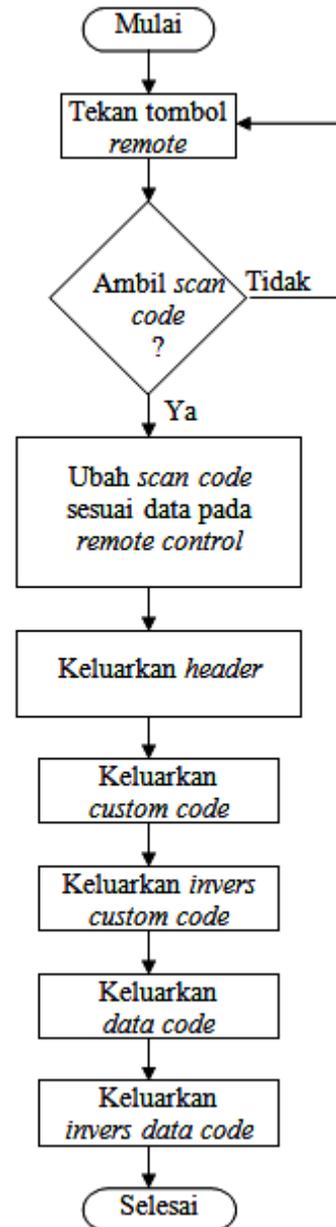
- Menyusun diagram alir (*flow chart*) program rangkaian kendali.
- Membuat perangkat lunak berdasarkan diagram alir yang telah disusun dengan menggunakan bahasa assembly.
- Perangkat lunak yang telah disusun, di-*compile* menggunakan *software* MCS-51 Macroassembler V.2.2 Intel Corporation menjadi data file Object (*\_.OBJ*) dan list (*\_.LST*).
- Apabila terjadi kesalahan, memeriksa kembali dengan membuka file yang berekstensi (*\_.LST*) untuk mengetahui pada bagian *upcode* yang mana terjadi kesalahan. Dan mengulangi kembali ke tahap b.
- Bila sudah tidak terjadi kesalahan, meng-*compile* data file Object (*\_.OBJ*) ke bentuk data file Hexadesimal (*\_.HEX*).
- Kemudian data file (*\_.HEX*) di-*compile* menjadi data file Binary (*\_.BIN*).
- Setelah semua proses selesai, program tersebut dimasukkan/ diisikan ke dalam PEROM internal dari tiap mikrokontroler sesuai dengan tiap bagian rangkaian yang dibuat.

Proses selengkapnya ditunjukkan pada Gambar 5.



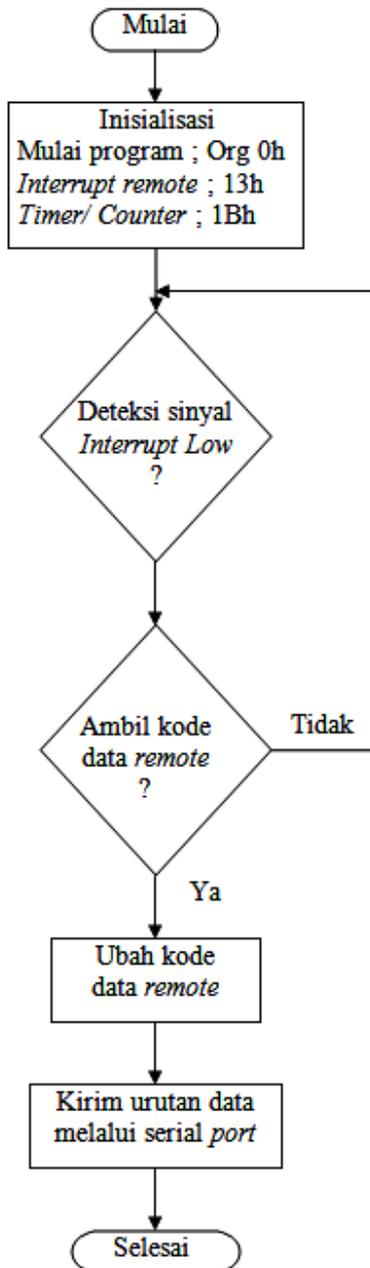
Gambar 5. Alur *Compiler* Bahasa Assembler

2.2.1 *Flow chart* Program pada Rangkaian Alat Pendeteksi *Data Remote*



Gambar 6. *Flow chart* Program pada Rangkaian Alat Pendeteksi *Data Remote*

## 2.2.2 Flow chart Program pada Rangkaian Pengendali



Gambar 7. Flow chart Program pada Rangkaian Pengendali.

## 2.3 Batasan Masalah

Pembahasan dalam jurnal ini berkisar pada prinsip pengiriman sinyal dari *remote control* dengan menggunakan *infra red*, Bahasa pemrograman yang digunakan adalah bahasa Assembler, Alat elektronik yang digunakan sebagai sample adalah tiga

buah lampu penerangan dimana masing-masing peralatan elektronik dikontrol oleh rangkaian *driver* Transistor yang dihubungkan dengan *Relay*.

## 3. Hasil dan Pembahasan

### 3.1 Pengujian Remote Control

Pengujian ini bertujuan untuk mengetahui frekuensi pembawa dan kebenaran nilai data (heksadesimal) dari *remote control*. Rangkaian alat penerima ini mempunyai sistem pemrograman seperti pada rangkaian penerima, tetapi nilai heksadesimal yang diterima ditampilkan menggunakan tujuh-segmen, alat pengujian ini sebelumnya telah dirancang dan dibuat.

Dari hasil pengujian nilai data pada *remote control*, menunjukkan adanya perbedaan data (nilai heksa) dari tiap tombol *remote*. Nilai ini dipakai untuk mengamati tiap rangkaian kontrol atau output. Sehingga tidak akan terjadi kesalahan dalam penekanan tombol *remote control* untuk menyalakan dan mematikan (*on* dan *off*) peralatan elektronik

### 3.2 Pengamatan Kerja Rangkaian Kendali Secara Elektronik

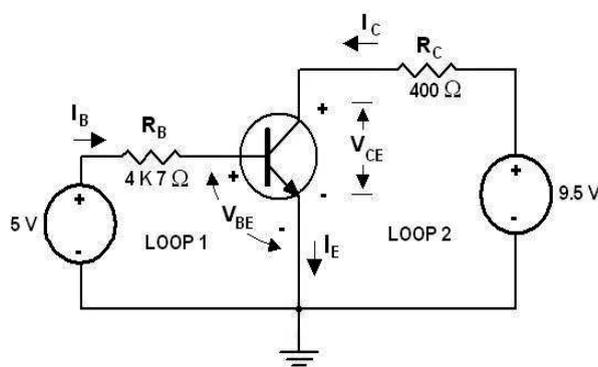
Rangkaian kendali terdiri dari komponen-komponen elektronik arus lemah dimana salah satunya adalah transistor. Transistor mempunyai 3 kutub yaitu Kolektor, Basis, dan Emitor. Aplikasi transistor disini bukanlah sebagai penguat, melainkan sebagai saklar elektronik, sehingga bila basis transistor menerima arus (berlogika tinggi) maka seolah-olah antara

kolektor dan emitor akan terhubung singkat, sedangkan bila arus basis 0 (berlogika rendah), maka antara kolektor dan emitor berperan sebagai resistor yang besarnya tak terhingga atau dikatakan terputus.

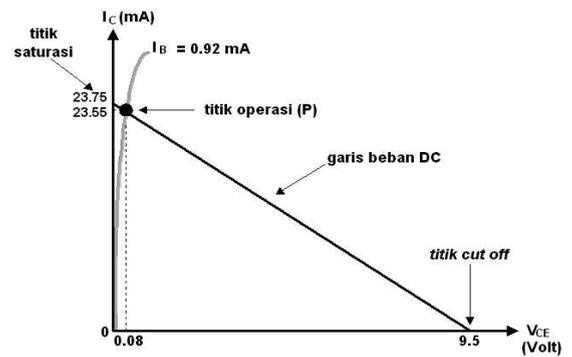
Untuk aplikasi saklar ini tentunya memerlukan perancangan yang didukung oleh parameter-parameter transistor seperti Tegangan Sumber ( $V_{CC}$ ), Tegangan Input Basis ( $V_{BB}$ ), Tegangan Kolektor-Emitor ( $V_{CE}$ ), dan sebagainya. Hal ini adalah untuk mendapatkan suatu transistor yang bekerja pada rentang tegangan yang telah diteditentukan.

Tabel 1. Hasil Pengukuran Parameter Transistor

Parameter	Besar	Satuan
Tegangan Sumber Kolektor ( $V_{CC}$ )	9.5	V
Tegangan Sumber Basis ( $V_{BB}$ )	5	V
Tegangan Kolektor-Emitor ( $V_{CE}$ )	80	mV
Tegangan Basis-Emitor ( $V_{BE}$ )	0.75	V
Arus Kolektor ( $I_C$ )	21.7	mA
Arus Basis ( $I_B$ )	0.89	mA



Gambar 8. Penyederhanaan Rangkaian



Gambar 9. Grafik Beban DC

Jika dibandingkan antara  $I_C$  dan  $I_B$  hasil perhitungan dengan  $I_C$  dan  $I_B$  hasil pengukuran ternyata hasilnya tidak jauh berbeda. Hal tersebut tidak terlalu berpengaruh pada transistor yang diaplikasikan sebagai saklar elektronik. Tetapi jika diaplikasikan dalam perangkat audio maka hal ini akan sangat mempengaruhi unjuk kerja sistem tersebut, yang mengakibatkan menurunnya kualitas sistem akibat titik P yang berada sangat dekat dengan daerah saturasi.

### 3.3 Daya Maksimum Perangkat Elektronik

Dengan arus operasi *relay* sebesar 21.7 mA maka dapat digunakan untuk memadamkan dan menyalakan perangkat elektronik dimana daya maksimum yang mampu dibebankan pada *relay* adalah:

$$P_{MAX} = V_{AC} \times I_{BEBAN\ MAX}$$

$$P_{MAX} = 220V \times 10A$$

$$P_{MAX} = 2200\ \text{Watt}$$

Arus maksimal beban yang dapat dilewatkan oleh *relay* tertera pada tutup komponen tersebut. Pengguna dapat memilih *relay* sesuai dengan besarnya arus yang akan masuk ke perangkat elektronik.

Untuk lebih aman sebaiknya menggunakan *relay* yang paling besar arus bebannya dan ditambah dengan *fuse* agar mencegah kelebihan arus yang melewati *relay*

### 3.4 Perbandingan Daya Antar Perangkat Elektronik

Arus yang masuk ke kolektor transistor tetap. Sedangkan arus yang beban yang akan dilewatkan oleh *relay* menuju perangkat elektronik berubah-ubah tergantung konsumsi daya dari perangkat elektronik tersebut. Berikut ini adalah perbandingan konsumsi daya dari beberapa perangkat elektronik

Tabel 2. Perbandingan Konsumsi Daya Perangkat Elektronik

PERANGKAT	I <sub>C</sub> (mA)	I <sub>BEBAN</sub> (A)	DAYA (WATT)
Lampu Pijar	21.7	0.182	40
Lampu Hemat Energi	21.7	0.091	20
Pompa Air	21.7	0.568	125
Kipas Angin	21.7	0.250	55
Rice Cooker	21.7	1.591	350
TV 21"	21.7	0.364	80
Blender	21.7	0.682	150
Mixer	21.7	0.909	200
Kulkas	21.7	0.936	206
Dispenser	21.7	1.727	380

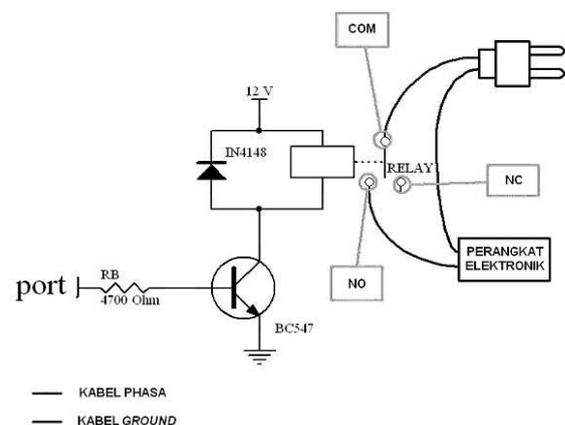
Daya dari masing-masing perangkat elektronik harus berada di bawah daya maksimum yang mampu dibebankan pada *relay*. Hal tersebut untuk mencegah agar *relay* tidak putus saat bekerja dan kendali perangkat dapat berjalan dengan lancar. Dari Tabel 2 terlihat bahwa daya yang diperlukan masih berada di bawah ambang batas daya beban maksimum

### 3.5 Penerapan Sistem Kendali pada Perangkat Elektronik

Untuk menerapkan sistem kendali ini tentunya harus direncanakan dengan matang dan juga saat instalasi harus didampingi oleh orang yang berpengalaman dalam bidang arus kuat sehingga mencegah terjadinya hal-hal yang tidak diinginkan. Untuk itu perlu direncanakan jalur tegangan AC yang akan masuk menuju ke rangkaian pengendali.

#### a. Perangkat yang Dihubungkan dengan Stop Kontak

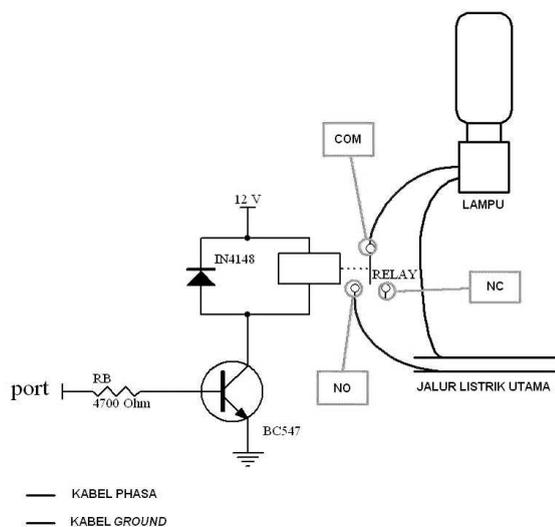
Untuk perangkat ini dapat diambil satu jalur menuju ke *relay* dimana dari salah satu kabel stop kontak akan masuk ke komponen *relay*. Contoh perangkat yang termasuk jenis ini yaitu Televisi, Mesin Cuci, AC, Kulkas, Pompa Air, dan perangkat-perangkat lainnya yang menggunakan stop kontak untuk mendapatkan sumber tegangan 220 VAC. Koneksi antara perangkat elektronik dalam kategori ini dapat dilihat pada Gambar 10.



Gambar 10. Pemasangan Perangkat dengan Stop Kontak

b. Perangkat yang Sudah Terhubung dengan Jalur Listrik Utama

Contoh perangkat ini adalah lampu-lampu listrik yang sudah ada terpasang di rumah. Apabila ingin menghubungkannya dengan perangkat kendali, maka aliran listrik di rumah harus dipadamkan seluruhnya. Koneksi antara perangkat jenis ini dapat dilihat pada Gambar 11.



Gambar 11. Koneksi Lampu dengan Perangkat Kendali

## 4. Penutup

### 4.1 Kesimpulan

Setelah melakukan pembuatan alat antarmuka kanal paralel untuk melakukan pengendalian perangkat elektronika, dalam penyusunan laporan ini penulis berkesimpulan bahwa:

1. Sinar infra merah (*infra red*) yang dipancarkan oleh pemancar infra merah tentunya mempunyai aturan tertentu agar data yang dipancarkan dapat diterima dengan baik di *receiver*. Oleh karena itu baik di *transmitter* infra merah maupun *receiver* infra merah harus mempunyai aturan yang sama dalam mentransmisikan

(bagian *transmitter*) dan menerima sinyal tersebut kemudian mendekodekannya kembali menjadi data *biner* (bagian *receiver*).

2. Data yang dikirim oleh tiap tombol *remote control* berbeda-beda sesuai dengan jenis dan perusahaan pembuat *remote control* tersebut.

### 4.2 Saran

1. Jalur data yang dipakai untuk mengendalikan perangkat elektronik dapat dimaksimalkan menjadi 31 jalur yaitu dari Port 0 – Port 3 kecuali P3.0 di kaki 10 karena digunakan sebagai sehingga jumlah perangkat yang dapat dikendalikan menjadi lebih banyak.

2. Pada pemrograman bisa dirancang agar bisa menggunakan dua buah tombol *Remote control* untuk menyalakan dan mematikan alat elektronik atau hanya dengan satu tombol *on* dan *off* untuk masing-masing alat elektronik.

### Daftar Pustaka

- 1) Putra, Afgianto Eko. "Balajar Mikrokontroler AT89C51/52/55 Teori dan Aplikasi" Gava Media, Yogyakarta, 2002.
- 2) Wardhana, Lingga. "Mikrokontroler AVR Seri ATmega8535 Simulasi, Hardware, dan Aplikasi" ANDI, Yogyakarta, 2006.
- Nalwan, Paulus Andi. "Teknik Antarmuka dan Pemrograman Mikrokontroler AT89C51" PT Elex Media Komputindo, Jakarta, 2003.
- 3) Idris, Kamal. "Elektronika Komunikasi", Erlangga, Jakarta, 1984.

- 4) Malvino, Albert P. "Prinsip-Prinsip Elektronik", Salemba Teknika, Bandung, 2004.
- 5) Malvino, Albert P. "Prinsip-Prinsip Elektronik", Erlangga, Jakarta, 1981.
- 6) Suratman. "Kamus Elektronika", CV, Pustaka Grafika, Bandung, 2001.
- 7) "Data Sheet",  
<http://www.fairchildsemi.com/ds/>, 14 Januari 2007.
- 8) "Hand Out Infrared",  
[http://www.centrin.net.id/delta.com/Hand\\_Out\\_Infrared.pdf](http://www.centrin.net.id/delta.com/Hand_Out_Infrared.pdf), 14 Nopember 2005.
- 9) "Infrared Receiver",  
[http://www.innovativeelectronics.com/innovative\\_electronics/download\\_files/manual/Manual%20DT-IO%20IR%20Receiver.pdf](http://www.innovativeelectronics.com/innovative_electronics/download_files/manual/Manual%20DT-IO%20IR%20Receiver.pdf), 26 Agustus 2006.
- 10) "Relay",  
<http://www.kpsec.freeuk.com/components/relay.htm>, 5 Januari 2007.
- 12) Prestiliano, Jasson. "Strategi Bahasa Assembler", Gava Media, Jogjakarta, 2005.