

**SISTEM KENDALI PERANGKAT LISTRIK
MENGUNAKAN PC MELALUI PORT PARALEL**



Oleh :

Chandra Aan Setiawan

NIM. M3304005

TUGAS AKHIR

**ditulis dan diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Ahli Madya Ilmu Komputer**

**PROGRAM DIPLOMA III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA**

2007

**SISTEM KENDALI PERANGKAT LISTRIK
MENGUNAKAN PC MELALUI PORT PARALEL**



Oleh :

Chandra Aan Setiawan

NIM. M3304005

TUGAS AKHIR

**ditulis dan diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Ahli Madya Ilmu Komputer**

**PROGRAM DIPLOMA III ILMU KOMPUTER
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM
UNIVERSITAS SEBELAS MARET
SURAKARTA**

2007

TUGAS AKHIR
SISTEM KENDALI PERANGKAT LISTRIK
MENGGUNAKAN PC MELALUI PORT PARALEL

yang disusun oleh

CHANDRA AAN SETIAWAN

NIM. M3304005

dibimbing oleh

Pembimbing I,

Pembimbing II,

Nuryani, M.Si

NIP. 132 258 048

Dra. Respatiwan, M.Si

NIP. 132 258 050

telah dipertahankan di depan Dewan Penguji
pada hari _____, tanggal Juli 2007
dan dinyatakan telah memenuhi syarat.

Anggota Tim Penguji

Tanda Tangan

1. Nuryani, M. Si

1

2. Dra. Respatiwan, M.Si

2

3. Hartono, S.Si

3

Surakarta, Juli 2007

Disahkan oleh

Fakultas MIPA

Dekan,

Ketua Program Studi DIII

Ilmu Komputer,

Prof. Drs. Sutarno, M.Sc.PhD

NIP. 130 906 776

Irwan Susanto, DEA

NIP. 132 134 694

ABSTRAK

Chandra Aan Setiawan, 2007, **SISTEM KENDALI PERANGKAT LISTRIK MENGGUNAKAN PC MELALUI PORT PARALEL**, Program DIII Ilmu Komputer Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sebelas Maret Surakarta.

Komputer atau disebut PC (*Personal Computer*) sudah berada hampir di setiap rumah, gedung atau perkantoran. Kebanyakan komputer lebih sering digunakan untuk keperluan pengetikan, memutar film, mendengarkan musik dan untuk permainan atau *game*. Pada saat sebagian orang juga dapat menghabiskan waktu seharian di depan komputer mereka. Hal ini dapat membuat orang malas untuk mengerjakan hal lain saat sibuk di depan komputer, misalnya saja untuk menghidupkan atau mematikan lampu, pendingin ruangan (*Air Conditioner*), atau perangkat elektronik lainnya.

Pada Tugas Akhir ini dirancang sistem kendali perangkat listrik dengan PC (*personal Computer*) dengan port paralel, yaitu suatu sistem yang berfungsi sebagai otomatisasi perangkat listrik menggunakan *timer*. Sistem ini mempunyai 2 blok program utama yaitu sistem yang berdasar *timer* waktu, dan sistem yang berdasarkan *timer* waktu dan tanggal.

Sebagai perantara antara PC dengan perangkat listrik AC (*Alternating Current*) menggunakan driver saklar elektronik yang dirancang menggunakan *optocoupler* MOC3020 atau MOC3041 dan *triac* BT137, yang dapat bekerja dengan baik apabila mendapat masukan 5 Volt DC dari komputer. Sedangkan pemrogramannya menggunakan Delphi 7.0 yang menggunakan komponen *suipack* untuk desain *form*. Agar dapat mengakses *port* paralel dengan baik digunakan sebuah *file library*, yaitu "*inpout32.dll*".

MOTTO

- ❖ **Sesungguhnya perbuatan sekecil apapun masih lebih berharga dari pada keinginan yang paling besar. (Larry Eisenberg).**
- ❖ **Sesungguhnya didalam ketekunan akan menghasilkan sesuatu yang besar.**
- ❖ **Carilah ilmu walaupun harus sampai ke negeri cina.**

PERSEMBAHAN

Kupersembahkan untuk

- ☞ Ibundaku tersayang,
Semoga karya ini dapat memberikan sedikit kebahagiaan.
- ☞ Keluarga Besarku, terima kasih atas dukungannya.
- ☞ Dik Rina EW, terima kasih atas motivasi dan perhatiannya!!!
- ☞ Teman-teman seperjuangan DIII Teknik Komputer 2004.
- ☞ Semua yang tidak dapat saya sebutkan satu per satu!!!

KATA PENGANTAR

Alhamdulillah robbil alamin, segala puji syukur kehadiran Allah SWT, yang telah memberikan iman, kekuatan, kecerdasan, semangat yang tinggi, dan berkah-Nya tugas akhir ini dapat terselesaikan tanpa halangan suatu apapun.

Selama penyusunan tugas akhir penyusun mendapatkan bantuan dari berbagai pihak, maka dalam kesempatan ini penulis mengucapkan terima kasih yang sebesar-besarnya kepada pihak-pihak sebagai berikut :

1. Bapak Irwan Susanto, DEA selaku Ketua Program DIII Ilmu Komputer Fakultas MIPA UNS.
2. Bapak Nuryani, M.Si dan Ibu Dra. Respatiwulan, M.si, selaku dosen pembimbing I dan pembimbing II yang telah banyak memberikan bimbingan dan masukan dalam penelitian tugas akhir ini.
3. Ibuku tercinta yang senantiasa mendo'a-kan ku.
4. Teman-teman seperjuangan DIII Teknik Komputer 2004.
5. Rekan-rekan dan semua pihak yang berkenan membantu menyelesaikan tugas akhir ini.

Semoga segala bantuan yang telah diberikan kepada penulis mendapatkan balasan dari Allah SWT, serta sepenuhnya menyadari bahwa tanpa bantuan beliau-beliau maka laporan ini tidak akan mendapat hasil yang baik. Semoga tugas akhir ini dapat menjadi manfaat bagi semua pihak.

Surakarta, Juli 2007

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
MOTTO	iv
PERSEMBAHAN	v
KATA PENGANTAR	vi
DAFTAR ISI	vii
DAFTAR TABEL	ix
DAFTAR GAMBAR	x
BAB I PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah.....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian.....	3
1.5 Manfaat Penelitian	3
BAB II LANDASAN TEORI DAN TINJAUAN PUSTAKA	
2.1 Port Paralel.....	4
2.1.1 Pengenalan Port Paralel	4
2.1.2 Diagram Pin Port paralel	4
2.2 Pengertian Sistem	6
2.3 Pengertian Database	7
2.4 Pengertian Delphi	8
2.5 Penertian Resistor	8
2.5.1 Lambang dan Bangun Resistor	8
2.5.2 Sifat Resistor dan Pengertian Resistivitas	9
2.6 Pengertian Dioda	10

2.6.1 Lambang dan Bentuk Komponen Dioda	10
2.6.2 Sifat Dioda	11
2.7 Pengertian Transistor	11
2.7.1 Definisi Transistor	11
2.7.2 Transistor Sebagai Saklar	12
2.8 Optocoupler
	14
2.9 Triac	15
BAB III METODE PENELITIAN	
3.1 Perancangan Hardware dan Software	17
3.2 Library Research.....	22
3.3 Alat Penelitian	23
BAB IV HASIL PENELITIAN DAN PEMBAHASAN	
4.1 Perancangan Hardware.....	24
4.1.1 Catu Daya	
24	
4.1.2 Driver Saklar Elektronik.....	25
4.2 Perancangan Software	26
4.2.1 Splash Form.....	27
4.2.2 Main Form	28
4.2.3 FTanggal Form.....	29
4.2.4 Form Manual	30
4.2.5 Form TestKoneksi	30
4.2.6 Form EditDenah	31
4.2.7 Form About	31
4.3 Perancangan Database	31
4.3.1 Main Form	32
4.3.2 Form FTanggal	
	33
4.3.3 Form EditDenah	34

4.4 Evaluasi Pengujian Sistem	35
4.4.1 Pengujian Perangkat Keras	35
4.4.2 Pengujian Sistem Kendali Perangkat Listrik	38
4.5 Keunggulan dan Kelemahan Sistem	42
4.5.1 Keunggulan	42
4.5.2 Kelemahan	42
BAB V HASIL PENELITIAN DAN PEMBAHASAN	
5.1 Kesimpulan.....	44
5.2 Saran	44
DAFTAR PUSTAKA	45
LAMPIRAN.....	46

DAFTAR TABEL

Tabel 2.1	Tabel Fungsi Pin <i>Port</i> Paralel DB-25.....	5
Tabel 4.1	Tabel THidup.....	32
Tabel 4.2	Tabel TMati	32
Tabel 4.3	Tabel TTglHidup	33
Tabel 4.4	Tabel TTglMati.	34
Tabel 4.5	Tabel TDenah	34
Tabel 4.6	Pengujian <i>device</i> dengan masukan 0 volt dan 5 volt	36
Tabel 4.7	Pengujian <i>port</i> paralel	38
Tabel 4.8	Hasil pengujian sistem <i>timer</i> waktu	39
Tabel 4.9	Hasil pengujian sistem <i>timer</i> tanggal dan waktu	40
Tabel 4.10	Hasil pengujian sistem <i>form</i> Fmanual	41

DAFTAR GAMBAR

Gambar 2.1	Pin konektor DB-25.....	5
Gambar 2.2	Lambang dan bentuk komponen resistor.....	9
Gambar 2.3	Kurva hubungan resistor, arus dan voltase.....	9
Gambar 2.4.	Lambang dan bentuk komponen dioda.....	11
Gambar 2.5	Kurva karakteristik dan daerah kerja transistor.....	12
Gambar 2.6	Transistor sebagai saklar.....	13
Gambar 2.7	Dasar rangkaian <i>optocoupler</i>	14
Gambar 2.8	Simbol <i>Triac</i>	16
Gambar 3.1	Skema sistem kendali perangkat listrik.....	17
Gambar 3.2	Diagram alir <i>Main Form</i>	19
Gambar 3.3	Diagram alir <i>form FTanggal</i>	20
Gambar 3.4	Diagram alir <i>form Fmanual</i>	21
Gambar 3.5	Diagram alir <i>form TestKoneksi</i>	22
Gambar 4.1	Skema catu daya.....	24
Gambar 4.2	Skema rangkaian internal saklar elektronik	25
Gambar 4.3	<i>Splash Form</i>	27
Gambar 4.4	<i>Main form</i>	28
Gambar 4.5	<i>Form Ftanggal</i>	29
Gambar 4.6	<i>Form manual</i>	30
Gambar 4.7	<i>Form TestKoneksi</i>	30
Gambar 4.8	<i>Form EditDenah</i>	31
Gambar 4.9	<i>Form About</i>	31
Gambar 4.10	<i>Driver</i> saklar elektronik dengan delapan keluaran	35
Gambar 4.11	Skema pin <i>port</i> paralel dan led.....	37
Gambar 4.12	Program tes port.....	37
Gambar 4.13	<i>Setting timer</i> waktu	39
Gambar 4.14	<i>Setting tanggal</i> dan waktu.....	40
Gambar 4.15	Tombol <i>ON/OFF</i> pada <i>form FManual</i>	41

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, komputer atau disebut PC (*Personal Computer*) sudah berada hampir di setiap rumah, gedung atau perkantoran. Kebanyakan komputer lebih sering digunakan untuk keperluan pengetikan, memutar film, mendengarkan musik dan untuk permainan atau *game*.

Pada era sekarang ini sebagian besar orang bekerja berjam-jam di depan komputer, bahkan sebagian orang juga dapat menghabiskan waktu seharian di depan komputer mereka. Hal ini membuat orang malas untuk mengerjakan hal lain saat sibuk di depan komputer, misalnya saja untuk menghidupkan atau mematikan lampu, pendingin ruangan (*Air Conditioner*), atau perangkat elektronik lainnya.

Padahal komputer juga bisa digunakan untuk keperluan pengontrolan peralatan listrik rumah tangga seperti lampu, kipas angin dan lain-lain dengan memanfaatkan *Port Paralel (Port printer)* pada komputer tersebut. Sehingga apabila sedang sibuk bekerja di depan komputer tidak perlu meninggalkan pekerjaannya hanya untuk menghidupkan atau mematikan lampu dan peralatan yang lain. *Port* merupakan satu set instruksi atau perintah sinyal dimana *microprocessor* atau CPU (*Central Processing Unit*) menggunakannya untuk memindahkan data dari atau ke piranti lain. Penggunaan umum *port* adalah untuk berkomunikasi dengan *printer, modem, keyboard* dan *display*. Kebanyakan *port* komputer adalah berupa kode digital, di mana tiap-tiap sinyal atau *bit* adalah berupa kode *biner* 1 atau 0.

Tugas akhir ini memanfaatkan fungsi dari *port* paralel dengan membuat *software* dan *hardware* sistem kendali perangkat listrik yang digunakan sebagai otomatisasi perangkat listrik AC (*Alternating Current*) menggunakan *timer*. Dalam pembuatan *software*, program aplikasi tersebut dibuat dengan menggunakan bahasa pemrograman Delphi dan *hardware* berupa rangkaian saklar

elektronik (*electronic switch*) mempunyai nama lain yaitu relai benda padat (*Solid State Relay*) atau SSR. Selain itu, saklar elektronik juga dikenal sebagai *opto-triac* yang nantinya rangkaian ini digunakan sebagai pengatur arus daya tinggi. Adapun untuk implementasinya bisa diterapkan pada perangkat listrik yang ada di rumah.

1.2 Perumusan Masalah

Berdasarkan uraian latar belakang masalah, maka rumusan masalah yang dibahas adalah bagaimana membuat suatu program aplikasi *sistem* kendali perangkat listrik menggunakan PC melalui *port* paralel, dan pembuatan *interface* perangkat keras yang digunakan untuk menghubungkan PC dengan perangkat listrik.

1.3 Batasan Masalah

Pembatasan masalah dalam tugas akhir ini adalah memfokuskan pembuatan sistem kendali perangkat listrik dari segi *hardware* maupun *software*, sehingga mempunyai batasan masalah sebagai berikut :

1. Program aplikasi *sistem* kendali perangkat listrik dirancang menggunakan Delphi 7.0, sebagai bahasa pemrograman yang bersifat GUI (*Graphic User Interface*), dan menggunakan file *library* “inpout32.dll”, yaitu piranti untuk mempermudah pemrograman dalam mengakses *port* paralel, akan tetapi tidak dibahas struktur file “inpout32.dll”.
2. *Database Management System* menggunakan Ms. Access XP, digunakan untuk menyimpan *setting* waktu, *setting* tanggal dan waktu, serta nama ruangan.
3. Bersifat *single user*, yaitu hanya bisa dikontrol melalui satu komputer.
4. *Hardware* yang digunakan untuk penghubung PC dengan perangkat listrik menggunakan *optocoupler* dan *triac* sebagai saklar, mempunyai 8 (delapan) *output* yang memiliki arus dan tegangan AC (*Alternating Current*), dalam tugas akhir ini perangkat listrik yang digunakan adalah lampu pijar.

1.4 Tujuan Penelitian

Tujuan dari tugas akhir ini adalah merancang *software* dan *hardware* sistem kendali perangkat listrik dengan PC (*Personal Computer*) menggunakan *port* paralel. Sehingga mempermudah dalam kegiatan pengoperasian peralatan listrik AC (*Alternating Current*) menggunakan otomatisasi *timer* (waktu maupun waktu dan tanggal).

1.5 Manfaat Penelitian

Manfaat dari tugas akhir ini adalah untuk menambah pengetahuan tentang *port* paralel sebagai piranti untuk mengirimkan data, yang dapat digunakan sebagai sistem kendali perangkat listrik AC (*Alternating Current*), yang menggunakan *driver* saklar elektronik sebagai piranti pemisah antara listrik bertegangan tinggi dan tegangan rendah. Selain itu diharapkan dari hasil penulisan laporan tugas akhir ini dapat memberikan inspirasi untuk pengembangan sistem di masa yang akan datang.

BAB II

LANDASAN TEORI DAN TINJAUAN PUSTAKA

2.1. *Port* Paralel

2.1.1. Pengenalan *Port* Paralel

Menurut Budiharto (2004), *port* paralel ialah *port* data komputer yang digunakan untuk mentransmisi 8 *bit* data dalam satu detak. Standar *port* paralel yang terbaru ialah IEEE 1284 yang dikeluarkan tahun 1994. standar ini mendefinisikan lima mode operasi sebagai berikut :

- a. Mode Kompabilitas.
- b. Mode *Nibble*.
- c. Mode *Byte*.
- d. Mode ECP (*Enhanced Parallel Port*).
- e. Mode ECP (*Extended Capability Port*).

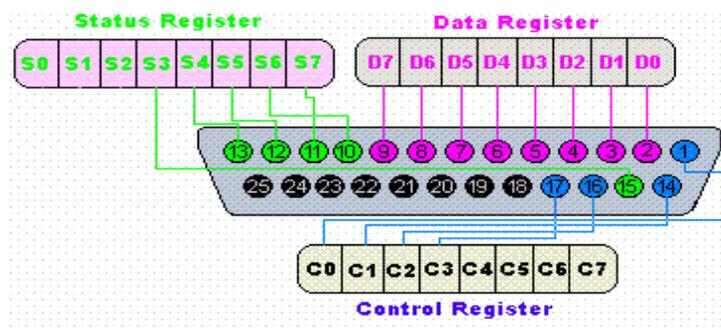
Tujuan standar yang baru tersebut ialah untuk mendesain *driver* dan peralatan yang baru dan kompatibel dengan peralatan yang lain serta standar paralel *port* sebelumnya (SPP) yang diluncurkan pada tahun 1981.

2.1.2. Diagram Pin *Port* Paralel

Menurut Sutadi (2002), ada dua macam konektor paralel *port*, yaitu 36 pin dan 25 pin. Konektor 36 pin dikenal dengan nama *Centronics* dan konektor 25 dikenal dengan nama DB-25. *Centronics* lebih dahulu ada dan digunakan daripada DB-25. DB-25 diperkenalkan oleh IBM (bersamaan dengan DB-9 untuk serial *port*) ketika memperkenalkan IBM PC pertama kali, yang bertujuan untuk menghemat tempat. Karena DB-25 lebih praktis, maka untuk

koneksitor *port* paralel pada komputer sekarang hanya digunakan sebagai konektor pada *printer* (dan atau piranti luar lainnya).

Layaknya komponen elektronika *port* paralel dibutuhkan dengan konektor betina dan jantan. Pada komputer konektor *port* paralel yang terpasang adalah DB-25 betina, sehingga kabel penghubung kabel keluar adalah DB-25 jantan. (Gambar 2.1 : Pin konektor DB-25).



Gambar 2.1 Pin konektor DB-25

Dari 25 pin konektor DB-25 tersebut, hanya 17 pin yang digunakan untuk saluran pembawa informasi dan yang berfungsi sebagai *ground* 8 pin seperti dalam Tabel 2.1. Ketujuh belas saluran informasi itu terdiri dari tiga bagian, yakni data 8 *bit*; status 5 *bit*; dan kontrol 4 *bit*. *Bit* kontrol dan status berfungsi dalam “jabat tangan” dalam proses penulisan data ke *port* paralel

Tabel 2.1. Tabel Fungsi Pin *Port* Paralel DB-25

DB-25	In/Out	Nama Sinyal	Register Bit
1	Out	nStrobe	C0
2	Out	Data 0	D0
3	Out	Data 1	D1
4	Out	Data 2	D2
5	Out	Data 3	D3
6	Out	Data 4	D4
7	Out	Data 5	D5

8	<i>Out</i>	Data 6	D6
9	<i>Out</i>	Data 7	D7
10	<i>In</i>	<i>nACK</i>	S 6 +
11	<i>In</i>	<i>BUSY</i>	S 7 -
12	<i>In</i>	<i>PaperEnd</i>	S 5 +
13	<i>In</i>	<i>Select</i>	S 4 +
14	<i>Out</i>	<i>nAutoFeed</i>	C 1 -
15	<i>In</i>	<i>nError</i>	S 3 +
16	<i>Laut</i>	<i>nInit</i>	C 2 +
17	<i>Laut</i>	<i>nSelectIn</i>	C 3 -
18-25	← →	<i>Ground</i>	

2.2. Pengertian Sistem

Banyak pendapat para ahli tentang sistem yaitu sebagai berikut :

- a. Sistem adalah seperangkat unsur yang melakukan sesuatu kegiatan atau beberapa tujuan dengan mengolah data suatu energi dalam jangka waktu tertentu guna menghasilkan informasi dan energi. (Mudrick dan Ross : dalam Jogiyanto, 2001).
- b. Bahwa unsur-unsur sistem terdiri dari :
 - Masukan (*Input*) yang memprosesnya dengan perhitungan, penggabungan unsur data, pemutakhiran jumlah (*up-dating account*), dll.
 - Keluaran (*Output*).
 (Scot : Dalam Anwar, 2003).
- c. Bahwa sebuah sistem harus mempunyai organisasi, hubungan, integrasi dan tujuan-tujuan yang sama.(Sidharta : Dalam Anwar, 2003).

Dari beberapa definisi di atas, maka dapat diambil suatu kesimpulan, yaitu sistem adalah totalitas himpunan bagian yang satu sama lain berinteraksi dan bersama-sama untuk mencapai tujuan atau sekelompok tujuan dalam suatu lingkungan. Sedangkan bagian sistem yang biasa disebut juga dengan sub-sistem, yang merupakan suatu kumpulan dari unsur tertentu, namun dalam mencapai tujuan semua sub-

sistem bekerja dalam mencapai tujuan dan keharmonisan dan keteraturan yang pasti.

2.3. Pengertian *Database*

Menurut Kadir (2004), *database* merupakan suatu bentuk pengelolaan data yang ditujukan agar pengaksesannya terhadap data dapat dilakukan dengan mudah. Sistem yang ditujukan untuk menangani *database* biasa disebut DBMS (*Database Management System*). Dengan menggunakan DBMS, pemakai dapat melakukan hal-hal seperti berikut dengan mudah :

- Menambah data
- Menghapus data
- Mengubah data
- Mencari data
- Menampilkan data dengan kriteria data
- Mengurutkan data

Sedangkan menurut Fatansyah (1999), *database* atau basis data terdiri dari 2 kata yaitu basis dan data. Basis dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul, sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia (pegawai, siswa, pembeli, pelanggan), barang, hewan, peristiwa, konsep, keadaan dan sebagainya, yang direkam dalam bentuk angka, huruf, simbol, teks, gambar, bunyi, atau kombinasinya. Basis data sendiri dapat diartikan dalam sejumlah sudut pandang seperti :

- a. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali secara cepat dan mudah.

- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundansi*) yang tidak perlu untuk memenuhi berbagai kebutuhan.
- c. Kumpulan *file*, tabel atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

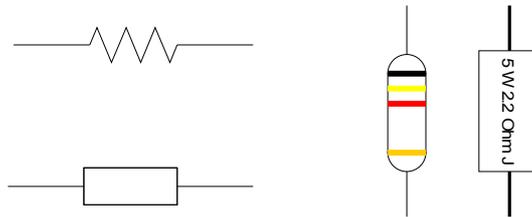
2.4. Pengertian Delphi

Menurut Kadir (2004), Delphi merupakan sebuah piranti pengembangan aplikasi berbasis windows yang dikeluarkan oleh Borland *International*. Perangkat lunak ini sangat terkenal di lingkungan pengembang aplikasi karena mudah untuk dipelajari dan dapat digunakan untuk menangani berbagai hal, dari aplikasi matematika, permainan (*games*), hingga *database*. Pada penanganan *database*, Delphi menyediakan fasilitas yang memungkinkan pemrogram dapat berinteraksi dengan *database* seperti dBase, Paradox, Oracle, MySQL, dan Access.

2.5. Pengertian Resistor

2.5.1 Lambang dan Bangun Resistor

Menurut Blocher (2003), dalam rangkaian elektronika, resistor dilambangkan dengan lambang yang terdapat pada Gambar 2.2, Lambang dan bentuk komponen resistor. Resistor terdapat dalam berbagai bentuk, tetapi paling sering berbentuk silinder kecil dengan satu sambungan pada masing-masing ujung. Silinder ini diberi lingkaran warna sebagai kode warna untuk menunjukkan sifatnya.



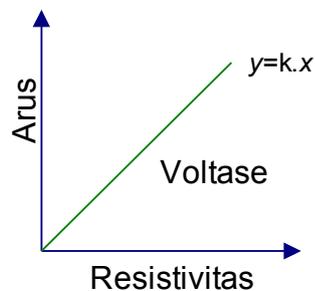
Gambar 2.2 Lambang dan bentuk komponen resistor

2.5.2 Sifat Resistor dan Pengertian Resistivitas

Menurut Blocher (2003), pada resistor terdapat hubungan berbanding lurus atau hubungan *linear* antara voltase dan arus, seperti diperlihatkan dalam Gambar 2.3, yang secara rumus terdapat bentuk :

$$y=k.x$$

Bentuk ini voltase bisa dipilih sebagai variabel y dan arus sebagai variabel x . dalam hal ini konstanta k disebut resistivitas R dan terdapat persamaan sebagai berikut :



Gambar 2.3 Kurva hubungan resistor, arus dan voltase.

$$V=R.I \Leftrightarrow R=\frac{V}{I}$$

Persamaan diatas disebut juga "hukum Ohm". Satuan dari resistivitas adalah Ohm, di singkat dengan huruf Yunani omega besar (Ω). 1 Ohm = 1 Ω adalah resistivitas yang terdapat kalau voltasenya

sebesar 1 volt menghasilkan arus sebesar 1 Ampere. Berarti untuk satuan dari resistivitas terdapat persamaan satuan.

$$[R] = \Omega = \frac{V}{A}$$

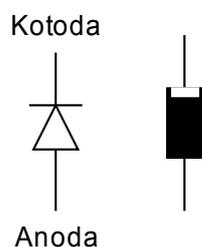
Resistivitas juga disebut sebagai tahanan dan besar resistivitas menunjukkan berapa kuat suatu komponen (misalnya resistor) menahan arus. Kalau resistivitas besar, berarti daya untuk menahan arus juga besar, sehingga arus menjadi kecil atau voltase harus besar untuk mendapatkan arus tertentu.

Besarnya resistivitas ditulis pada resistor dengan memakai lingkaran berwarna sebagai kode warna. Lingkaran pertama menunjukkan angka pertama dan lingkaran kedua menunjukkan angka kedua, lingkaran ketiga menunjukkan berapa banyak nol harus ditambahkan pada kedua angka sebelumnya.

2.6. Pengertian Dioda

2.6.1 Lambang dan Bentuk Komponen Dioda

Menurut Blocher (2003), satu komponen lain yang paling penting dalam elektronika adalah dioda. Dalam skema rangkaian, dioda dilambangkan seperti Gambar 2.4 sebelah kiri. Dari lambang sudah dilihat bahwa arah arus mempengaruhi sifat dari dioda. Satu sisi dari dioda disebut anoda, yang lain katoda. Katoda ada pada ujung depan segitiga.



Gambar 2.4. Lambang dan bentuk komponen dioda

2.6.2 Sifat Dioda

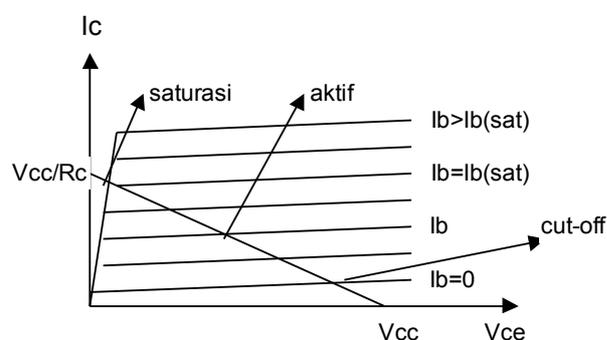
Menurut Blocher (2003), sebagai pendekatan pertama bisa dikatakan bahwa dioda mengijinkan arus untuk mengalir ke satu arah saja. Ketika anoda mendapatkan voltase yang lebih positif daripada katoda, maka arus bisa mengalir dengan bebas. Dalam situasi ini dikatakan dioda dibias maju. Kalau voltase dibalikkan, berarti katoda positif terhadap anoda, arus tidak bisa mengalir kecuali suatu arus yang sangat kecil. Dalam situasi ini dikatakan dioda dibias balik atau dibias mundur. Arus yang mengalir ketika dioda dibias balik disebut arus balik atau arus bocor dari dioda dan arus itu begitu kecil sehingga dalam kebanyakan rangkaian bisa diabaikan. Arus bisa mengalir kearah segitiga dalam lambang skema rangkaian. Supaya arus bisa mengalir ke arah maju, voltase harus sebesar ≈ 0.7 V pada dioda Silikon (disingkat dengan Si) dan ≈ 0.3 V pada dioda Germanium (disingkat dengan Ge) dan votase yang lebih besar lagi untuk LED.

2.7. Pengertian Transistor

2.7.1 Definisi Transistor

Menurut Malvino (1994), Transistor adalah sebuah komponen aktif yang terdiri dari tiga terminal yaitu *basis*, kolektor

dan *emiter*. Transistor akan bekerja apabila terdapat arus yang mengalir pada basis, artinya untuk transistor NPN bila basis diberi tegangan positif lebih dari tegangan bias basis emiter maka transistor akan melewatkan arus dari kolektor ke emiter, jika tegangan basis dinaikkan maka akan tercapai kondisi saturasi (jenuh) dan sebaliknya bila basis diberi tegangan negatif maka transistor akan *cut-off*. Sedangkan untuk transistor PNP bekerja pada kondisi sebaliknya.

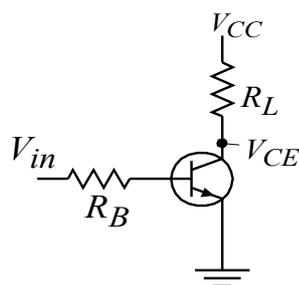


Gambar 2.5 Kurva karakteristik dan daerah kerja transistor.

Transistor memiliki 3 (tiga) daerah kerja, yaitu daerah aktif, daerah saturasi dan daerah *cut-off*, seperti yang ditunjukkan pada Gambar 2.5

2.7.2 Transistor Sebagai Saklar

Menurut Malvino (1994), prinsip pengoperasian transistor sebagai saklar yaitu dengan mengoperasikannya pada dua keadaan ekstrim, yaitu dalam keadaan kerja penuh (saturasi) dan keadaan tidak bekerja sama sekali (*cut-off*). Perubahan dari keadaan satu ke keadaan yang lainnya dapat berupa perubahan tegangan maupun perubahan arus.



Gambar 2.6 Transistor sebagai saklar.

Gambar 2.6 menunjukkan bagaimana suatu transistor dioperasikan sebagai saklar. Dalam keadaan $V_{in} = 0$ dan $I_B = 0$ yang berarti tidak ada sinyal masukan, maka transistor akan berada dalam kondisi mati. Pada keadaan ini tidak ada arus yang mengalir melalui beban R_L . Kondisi seperti ini dapat disamakan dengan sebuah saklar yang sedang terbuka karena tegangan antara kolektor dengan emitor besarnya mendekati V_{CC} . Karena I_C mendekati nol (harga sama dengan arus bocor I_{ceo}), maka jatuh tegangan pada R_L dapat diabaikan.

Jika V_{in} diberikan cukup besar sehingga I_B juga cukup besar, maka transistor akan berubah dari keadaan *cut-off* menuju keadaan saturasi. Keadaan saturasi adalah keadaan dimana arus I_C mencapai keadaan maksimum sehingga kenaikan I_B tidak lagi menyebabkan kenaikan I_C . Kondisi seperti ini dapat disamakan dengan sebuah saklar yang sedang tertutup (*on*).

Arus yang mengalir pada keadaan ini pada beban R_L adalah :

$$I_{R_L} = I_{C(sat)} = \frac{V_{CC} - V_{CE(sat)}}{R_L}$$

Karena $V_{CE(sat)} \cong 0$, maka I_{R_L} dapat didekati dengan persamaan :

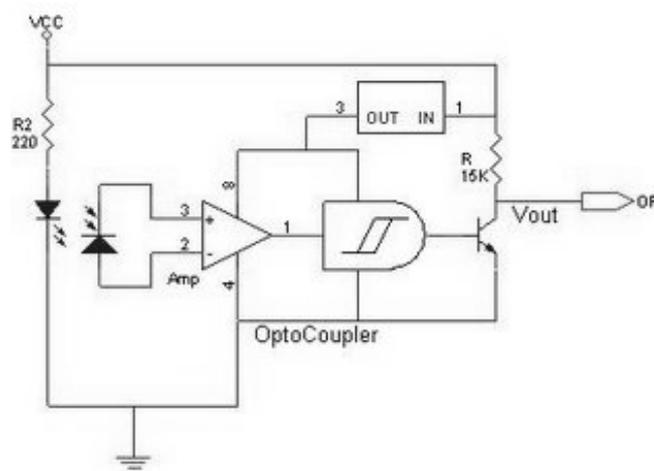
$$I_{R_L} = I_{C(sat)} = \frac{V_{CC}}{R_L}$$

Tegangan masukan V_{in} yang menyebabkan transistor mencapai keadaan saturasi adalah:

$$V_{in} = I_{B(sat)}R_B + V_{BE}$$

2.8. Optocoupler

Optocoupler merupakan piranti elektronika yang berfungsi sebagai pemisah antara rangkaian *power* dengan rangkaian kontrol. *Optocoupler* adalah salah satu jenis komponen yang memanfaatkan sinar sebagai pemicu *on/off*-nya. *Opto* berarti *optic* dan *coupler* berarti pemicu. Sehingga bisa diartikan bahwa *optocoupler* merupakan suatu komponen yang bekerja berdasarkan picu cahaya *optic opto-coupler* termasuk dalam sensor, dimana terdiri dari dua bagian yaitu *transmitter* dan *receiver*. Dasar rangkaian dapat ditunjukkan seperti pada Gambar 2.7 Dasar rangkaian *optocoupler*



Gambar 2.7 Dasar rangkaian *optocoupler*

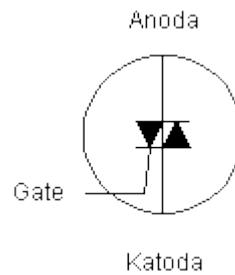
Bagian pemancar atau *transmitter* dibangun dari sebuah led *infra* merah untuk mendapatkan ketahanan yang lebih baik daripada menggunakan led biasa. Sensor ini bisa digunakan sebagai isolator dari rangkaian tegangan rendah kerangkaian tegangan tinggi. Selain itu juga bisa dipakai sebagai pendeteksi adanya penghalang antara *transmitter* dan *receiver* dengan memberi ruang uji dibagian tengah antara led dengan *photo* transistor. Penggunaan ini bisa diterapkan untuk mendeteksi putaran motor atau mendeteksi lubang penanda disket pada *disk drive* komputer. Penggunaan dari optocoupler tergantung dari kebutuhannya. Ada berbagai macam bentuk, jenis, dan *type*. Seperti MOC 3040 atau 3020, 4N25 atau 4N33 dan sebagainya.

Pada umumnya semua jenis optocoupler pada lembar datanya mampu dibebani tegangan sampai 7500 Volt tanpa terjadi kerusakan atau kebocoran. Biasanya dipasaran *optocoupler* tersedianya dengan *type* 4NXX atau MOC XXXX dengan X adalah angka *part valuenya*. Untuk *type* 4N25 ini mempunyai tegangan isolasi sebesar 2500 Volt dengan kemampuan maksimal led dialiri arus *forward* sebesar 80 mA. Namun besarnya arus led yang digunakan berkisar antara 15mA - 30 mA dan untuk menghubungkannya dengan tegangan +5 Volt diperlukan tahanan pembatas (Sumber : <http://elektronika-elektronika.blogspot.com>).

2.9. Triac

Boleh dikatakan SCR adalah *thyristor* yang *uni-directional*, karena ketika *ON* hanya bisa melewatkan arus satu arah saja yaitu dari anoda menuju katoda. Struktur *Triac* sebenarnya adalah sama dengan dua buah SCR yang arahnya bolak-balik dan kedua *gate*-nya disatukan.

Simbol triac ditunjukkan pada Gambar 2.8. *Triac* biasa juga disebut *thyristor bi-directional*. Selain itu *triac* bekerja mirip seperti SCR yang paralel bolak-balik, sehingga dapat melewati arus dua arah.



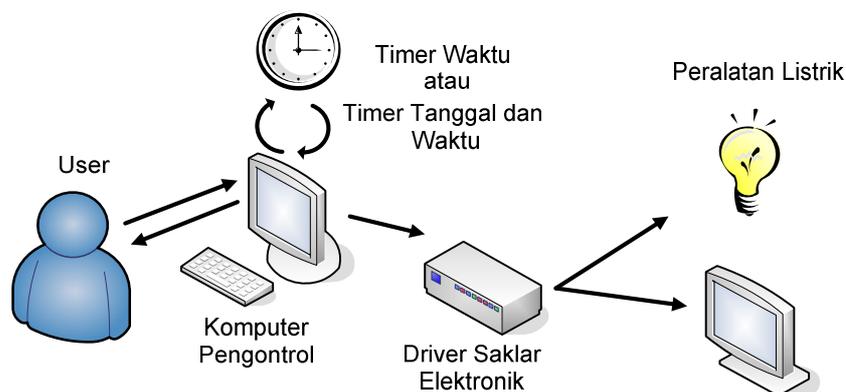
Gambar 2.8 Simbol *Triac*

(Sumber : <http://electroniclab.com> : 2003-2005)

BAB III METODE PENELITIAN

3.1 Perancangan *Hardware* dan *Software*

Perancangan sistem kendali perangkat listrik mencakup keseluruhan kebutuhan *software* maupun *hardware*. Gambar 3.1 Menunjukkan skema sistem kendali perangkat listrik secara umum.



Gambar 3.1. Skema sistem kendali perangkat listrik secara umum

Berdasarkan gambar 3.1 Skema sistem kendali perangkat listrik terlihat bahwa sistem mempunyai beberapa blok fungsional yaitu :

- a. *User*, merupakan pengguna yang akan mengatur kerja sistem.
- b. Komputer pengontrol, digunakan sebagai jalannya program aplikasi yang mengontrol *driver* saklar elektronik yang menggunakan sistem *timer*.
- c. *Driver* saklar elektronik, berfungsi sebagai saklar yang menghubungkan antara PC dengan perangkat listrik AC (*Alternating Current*) dengan komputer. *Driver* saklar elektronik ini mempunyai 8 (delapan) keluaran.
- d. Peralatan listrik dapat berupa lampu pijar, televisi, kipas angin, atau peralatan yang lain, tetapi dalam tugas akhir ini perangkat listrik yang digunakan adalah lampu pijar.

Perancangan *hardware* meliputi pembuatan *driver* saklar lampu yang menghubungkan PC dengan peralatan listrik yang mempunyai tegangan dan arus AC (*Alternating Current*). Perancangan *software* meliputi program sistem kendali

perangkat listrik yang dibuat menggunakan Delphi 7.0 dan mempunyai *database* yang dibuat di Microsoft Access XP.

A. Perancangan *Hardware* (Perangkat Keras).

Perancangan perangkat keras berupa penyusunan komponen-komponen elektronika menjadi satu kesatuan sistem rangkaian yang bisa bekerja sesuai dengan fungsinya. Dalam perancangan *hardware* terdapat 2 bagian yang penting, antara lain :

1. Catu Daya

Catu daya merupakan suatu rangkaian elektronika yang digunakan untuk mengubah dari arus dan tegangan bolak-balik menjadi searah. Keluaran dari catu daya ini menggunakan IC 7805 sehingga mempunyai keluaran tegangan DC sebesar 5 volt untuk mensuplai *driver* saklar elektronik.

2. *Driver* Saklar Elektronik

Driver saklar elektronik merupakan *interface* yang menghubungkan antara PC (*Personal Computer*) dengan perangkat listrik AC melalui konektor *port* paralel. Perancangan *driver* elektronik menggunakan *optocoupler* MOC3020 (atau MOC3041) dan *triac* BT137 sebagai komponen isolasi antara tegangan tinggi (AC) dan tegangan rendah (DC).

B. Perancangan *Software* (Perangkat Lunak).

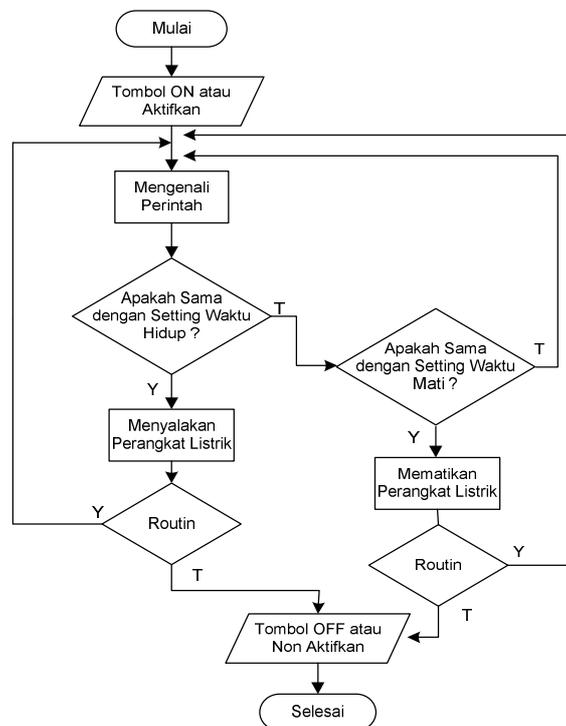
Perancangan *software* dimaksudkan agar komputer dapat bertugas mengirimkan data ke *driver* saklar elektronik. *Software* sistem kendali perangkat listrik dibuat menggunakan bahasa pemrograman Delphi 7.0 yang berbasis GUI (*Graphic User Interface*), dan menggunakan komponen *suipack* untuk desain *form* program aplikasi serta penyimpanan *setting* waktu dan tanggal menggunakan Microsoft Access XP. Sedangkan untuk mempermudah pemrograman dalam mengakses *port* paralel digunakan file *library* "inpout32.dll".

Program aplikasi sistem kendali perangkat listrik dirancang mempunyai beberapa *form*, yaitu : *splash form*, *main form*, *form* FTanggal,

form FManual, *form* TestKoneksi, *form* EditDenah, dan *form* About. Sedangkan beberapa *form* yang berfungsi untuk menyalakan perangkat listrik dan melakukan pengecekan terhadap status sambungan *port* paralel yang terdapat pada sistem antara lain :

1. Main Form

Main form mempunyai fungsi *timer* untuk menghidupkan atau mematikan *device* (perangkat listrik) berdasarkan *setting* waktu hidup dan mati. Sehingga apabila diaktifkan kemudian jam di komputer mempunyai nilai yang sama dengan *setting* waktu hidup, maka *device* (perangkat listrik) tersebut akan menyala, begitu juga sebaliknya apabila jam di komputer sama dengan *setting* waktu mati, maka *device* tersebut juga akan mati. Gambar 3.2 menunjukkan diagram alir *main form*.



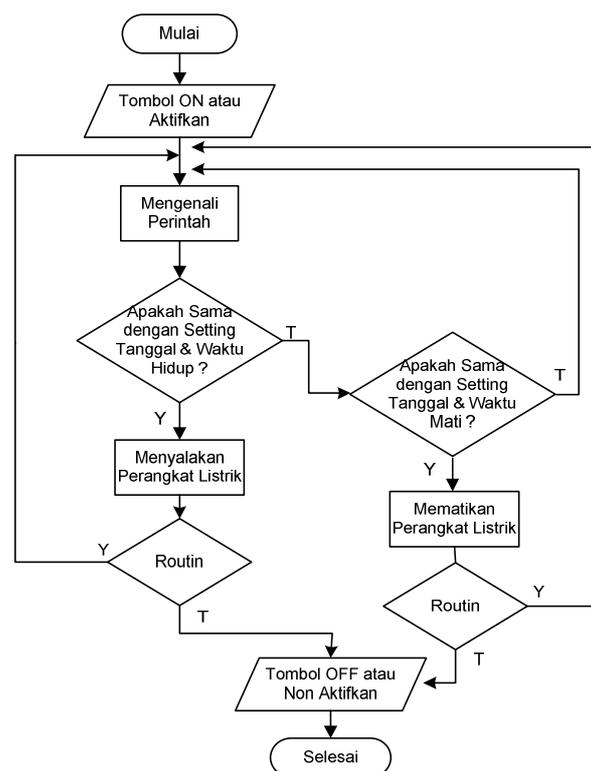
Gambar 3.2. Diagram alir *Main form*

Berdasarkan Gambar 3.2. Diagram alir *Main form*, sistem diaktifkan dengan perintah tombol *ON* atau tombol aktifkan (*semua device*), maka sistem akan mengenali perintah dengan mendeklarasikan *setting* waktu hidup dan mati. Apabila *setting* waktu hidup mempunyai nilai yang sama dengan jam di komputer, maka sistem akan menyalakan

perangkat listrik, tetapi apabila *setting* jam mati mempunyai nilai yang sama dengan jam di komputer, maka sistem akan mematikan perangkat listrik. Hal tersebut berarti jika jam di komputer tidak mempunyai nilai yang sama dengan *setting* sistem tidak memberikan respon (perangkat listrik tidak menyala). Sebelum tombol *OFF* atau non aktif dieksekusi sistem masih melakukan pengulangan.

2. Form FTanggal

Form FTanggal mempunyai fungsi *timer* untuk menghidupkan atau mematikan *device* (perangkat listrik) berdasarkan *setting* tanggal dan waktu hidup serta *setting* tanggal dan waktu mati. Sehingga apabila diaktifkan kemudian tanggal dan jam di komputer mempunyai nilai yang sama dengan *setting* tersebut, maka *device* (perangkat listrik) tersebut akan menyala, begitu juga sebaliknya apabila kemudian tanggal dan jam di komputer sama dengan *setting* waktu mati, maka *device* (perangkat listrik) tersebut juga akan mati. Gambar 3.3 menunjukkan diagram alir *main form*.

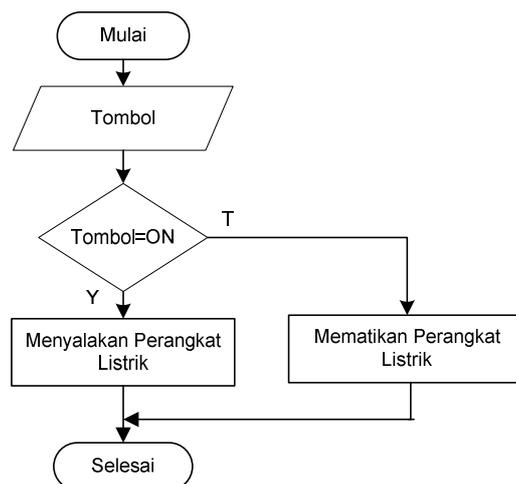


Gambar 3.3. Diagram alir *form* FTanggal

Berdasarkan Gambar 3.3. Diagram alir *form* FTanggal, sistem diaktifkan dengan perintah tombol *ON* atau tombol aktifkan (semua *device*), maka sistem akan mengenali perintah dengan mendeklarasikan *setting* tanggal dan waktu hidup serta *setting* tanggal dan waktu mati. Apabila *setting* tanggal dan waktu hidup mempunyai nilai yang sama dengan tanggal dan jam di komputer, maka sistem akan menyalakan perangkat listrik, tetapi apabila *setting* tanggal dan waktu mati, mempunyai nilai yang sama dengan jam di komputer, maka sistem akan mematikan perangkat listrik. Hal tersebut berarti jika tanggal dan jam komputer tidak mempunyai nilai yang sama dengan *setting* sistem tidak memberikan respon (perangkat listrik tidak menyala). Sebelum tombol *OFF* atau non aktif dieksekusi sistem masih melakukan pengulangan.

3. *Form* FManual

Form FManual mempunyai fungsi menghidupkan dan mematikan perangkat listrik melalui tombol *ON* dan *OFF*, tanpa adanya fasilitas *timer*. Gambar 3.4 menunjukkan diagram alir *form* FManual.

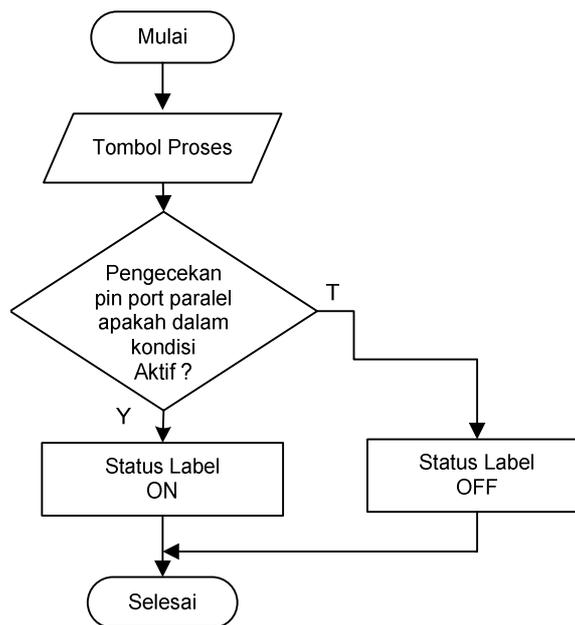


Gambar 3.4 Diagram alir *form* Fmanual

Berdasarkan Gambar 3.4. Diagram alir *form* FTanggal, jika menekan tombol *ON*, maka perangkat listrik akan dihidupkan, tetapi jika perintah yang diberikan adalah *OFF*, maka perangkat listrik akan dimatikan.

4. *Form* TestKoneksi

Form TestKoneksi ini merupakan sebuah *form* yang digunakan untuk pengecekan status sambungan pada *port* paralel. Gambar 3.5 menunjukkan diagram alir dari *form* TestKoneksi.



Gambar 3.5 Diagram alir *form* TestKoneksi

Berdasarkan Gambar 3.4. Diagram alir *form* TestKoneksi, jika tombol proses dieksekusi akan melakukan pengecekan status pada *port* paralel, apabila *port* paralel dalam keadaan aktif (diasumsikan dalam keadaan ideal perangkat listrik juga hidup, apabila ternyata perangkat listrik tidak menyala maka berarti perangkat listrik tersebut mengalami kerusakan) maka *port* paralel di pin tersebut mempunyai label *ON*, begitu juga sebaliknya *port* paralel dalam keadaan tidak aktif (hal ini diasumsikan perangkat listrik mati) maka *port* paralel di pin tersebut mempunyai label *OFF*.

3.2 *Library Research*

Mempelajari kepustakaan yang berhubungan dengan pembuatan program sistem kendali perangkat listrik (Delphi, Ms. Access, dll) dan pembuatan perangkat keras (*port* paralel, *triac*, *optocoupler*, dll).

3.3 Alat Penelitian

Peralatan yang digunakan untuk menunjang dalam penelitian dan pengembangan sistem kendali perangkat listrik ini, antara lain sebagai berikut :

1. Seperangkat komputer dengan spesifikasi :
 - a. *Processor* : Celeron 1,8 GHz
 - b. *Memory* : DDR 256 MB
 - c. *VGA* : 128 MB
 - d. *Hardisk* : 20 GB
 - e. *Media masukan* : Papan ketik (*keyboard*) dan *mouse*
 - f. *Kartu suara* : Realtek AC '97
 - g. *Media Display* : Monitor CRT 17 "
2. *Software* Borland Delphi 7.0 yang digunakan sebagai *visual* pemrograman berbasis GUI (*Graphic User Interface*), serta menggunakan komponen *suipack* untuk desain *form*. Dalam pembuatan program digunakan *file library* "inpout32.dll" untuk mempermudah pemrograman akses *port* paralel.
3. Microsoft Access XP digunakan dalam pembuatan *database* untuk menyimpan *setting* waktu, tanggal dan nama ruang.
4. Komponen elektronika untuk merancang *catu daya* dan *driver* saklar elektronik, seperti resistor, transistor, transformator, dioda, kapasitor, *triac* BT137, *Optocoupler* MOC3041 (atau MOC3020), dll.

BAB IV

HASIL PENELITIAN DAN PEMBAHASAN

Hasil perancangan tugas akhir ini pada dasarnya dibagi menjadi dua bagian, yaitu perancangan perangkat keras (*hardware*) dan perancangan perangkat lunak (*software*). Perancangan perangkat keras berupa penyusunan komponen-komponen elektronika menjadi satu kesatuan sistem rangkaian yang bisa bekerja sesuai dengan fungsinya.

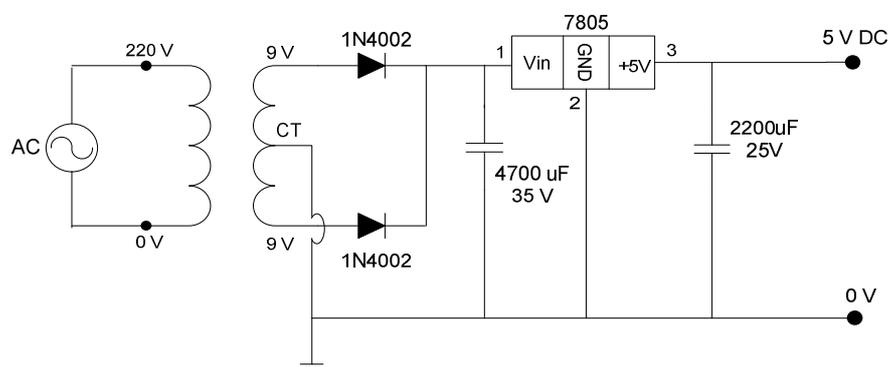
Sedangkan perancangan perangkat lunak menggunakan bahasa pemrograman Delphi 7.0 yang menghubungkan antara *driver* rangkaian dengan komputer, serta *database* sebagai media penyimpanan *setting* waktu, tanggal dan nama ruang.

4.1. Perancangan *Hardware* (Perangkat Keras).

Perangkat keras untuk mendukung sistem yang akan dibuat antara lain catu daya dan *driver* saklar elektronik sebagai penghubung antara pemakai dan *device* perangkat listrik (lampu pijar, kipas angin, atau perangkat yang lain).

4.1.1. Catu Daya.

Suatu rangkaian dapat bekerja dengan baik apabila adanya catu daya, dalam hal ini daya atau energi listrik arus searah. Gambar 4.1 Skema catu daya, menunjukkan rangkaian yang digunakan untuk mensuplai *driver* saklar elektronik.



Gambar 4.1 Skema catu daya

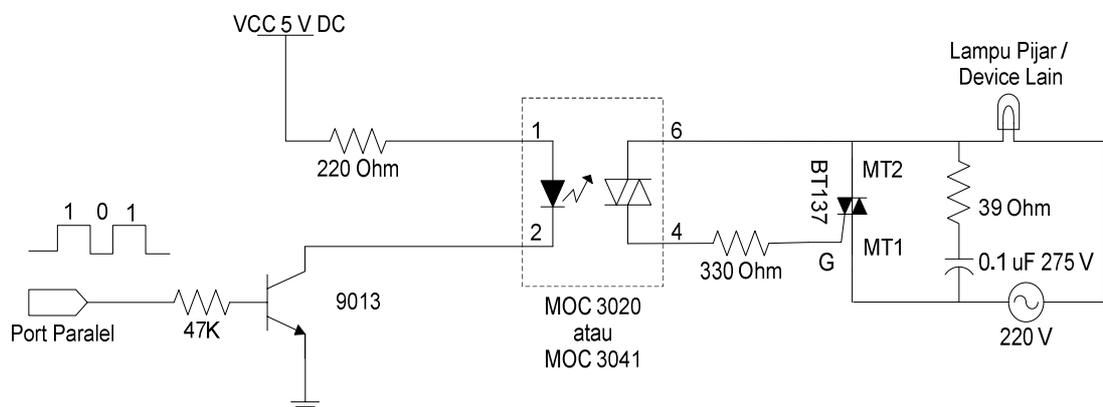
Gambar 4.1 menunjukkan rangkaian catu daya dari trafo digunakan tegangan 9 volt, tegangan ini digunakan untuk mensuplai IC 7805 pada pin 1. Pada pin ke-3 IC 7805 akan menghasilkan tegangan sebesar sebesar 5 volt. Keluaran sebesar 5 volt tersebut akan digunakan untuk mensuplai kebutuhan vcc *driver* elektronik.

4.1.2. *Driver Saklar Elektronik.*

Saklar elektronik (*electronic switch*) mempunyai nama lain yaitu relai benda padat (*Solid State Relay*) atau SSR. Selain itu, saklar elektronik juga dikenal sebagai *opto-triac*. Disebut *opto-triac* karena kontruksi dalamnya menggunakan gabungan antara *triac* dengan *optocoupler* sebagai komponen utama rangkaian pemicunya.

Optocoupler merupakan komponen elektronika yang digunakan untuk memindahkan informasi dari tegangan rendah ke tegangan tinggi. Sedangkan triac merupakan kependekan dari *triode alternating current switch* atau *saklar triode* untuk arus bolak-balik.

Biasanya SSR ini digunakan sebagai antarmuka antara rangkaian pengendali yang bertegangan DC (*direct current*) rendah dengan rangkaian jala-jala AC (*Alternating Current*) yang bertegangan jauh lebih tinggi.



Gambar 4.2 Skema rangkaian internal saklar elektronik

Berdasarkan Gambar 4.2, saklar elektronik mempunyai prinsip kerja sebagai berikut : keluaran dari *port* paralel PC akan mengaktifkan resistor 47 K Ω yang berfungsi sebagai pembatas arus untuk *basis* transistor. Transistor tersebut berfungsi sebagai saklar untuk *katode* dari resistor 220 Ω yang terhubung MOC3020 (atau MOC3041). Kemudian arus Vcc akan mengalir melalui resistor 220 Ω dan akan mengaktifkan MOC3041 melalui *anode*-nya (pin1). Resistor disini berfungsi sebagai pembatas arus untuk anoda MOC3041 (pin 1).

Keluaran MOC3041 akan digunakan untuk memicu *triac* BT137 yang dikewati arus AC (*Alternating Current*). *Triac* tersebut akan aktif jika MOC3041 mengeluarkan nilai logika 1 (maksimum 6 Volt DC). Apabila dibandingkan dengan relai elektromagnetik (*electromagnetic relay* – EMR), SSR mempunyai beberapa kelebihan antara lain :

- a. Lebih dapat diandalkan dan lebih tahan lama karena tidak mempunyai bagian yang bergerak untuk pensaklaran.
- b. Sesuai dengan rangkaian yang menggunakan rangkaian terpadu (*Integrated Circuit* atau IC).
- c. Tidak menimbulkan *interferensi* elektromagnetik.
- d. Lebih tahan terhadap kejutan dan getaran.
- e. Mempunyai waktu tanggapan yang lebih cepat.

Saklar elektronik yang digunakan dalam perangkat keras ini mempunyai tegangan DC masukan maksimum 6 volt dengan keluaran berupa tegangan AC maksimum 240 volt dan dengan arus maksimum 12 ampere.

4.2. Perancangan *Software* (Perangkat Lunak)

Perancangan *software* (perangkat lunak) sistem kendali perangkat listrik dimaksudkan agar komputer dapat bertugas untuk mengirimkan data ke *driver saklar* elektronik. Dalam membuat perangkat lunak ini menggunakan bahasa pemrograman Delphi 7.0, komponen *suipack* untuk desain *form*, dan Microsoft

Access XP untuk menyimpan *database setting* waktu dan tanggal, serta nama ruangan. Dalam mempermudah pemrograman dalam hal mengakses *port* paralel menggunakan *file library* "inpout32.dll".

Perancangan aplikasi sistem kendali perangkat listrik pada peralatan listrik AC dirancang menggunakan enam *form*. Ke-enam *form* tersebut yaitu:

1. *Splash form*, yaitu sebuah *form* yang tampil pertama kali dan menunjukkan nama program sekaligus NIM dan nama pembuat.
2. *Main*, merupakan *form* utama yang berguna untuk memanggil *form* lain selain digunakan sebagai sistem program berdasarkan *timer* waktu (hidup dan mati).
3. *FTanggal*, merupakan *form* yang digunakan untuk mengaktifkan kerja sistem berdasarkan *timer* waktu dan tanggal hidup, serta waktu dan tanggal mati.
4. *TestKoneksi*, *form* yang digunakan untuk tes sambungan pada *port* paralel, perangkat listrik dalam keadaan hidup atau mati.
5. *Manual*, merupakan *form* yang digunakan untuk menghidupkan atau mematikan *device* tanpa menggunakan fasilitas *timer*.
6. *FEdit*, merupakan *form* yang digunakan untuk merubah atau meng-*edit* nama ruang dan menyimpan nama ruang baru ke *database*.
7. *FAbout*, merupakan *form* yang berisi identitas pembuat.

4.2.1. *Splash Form*



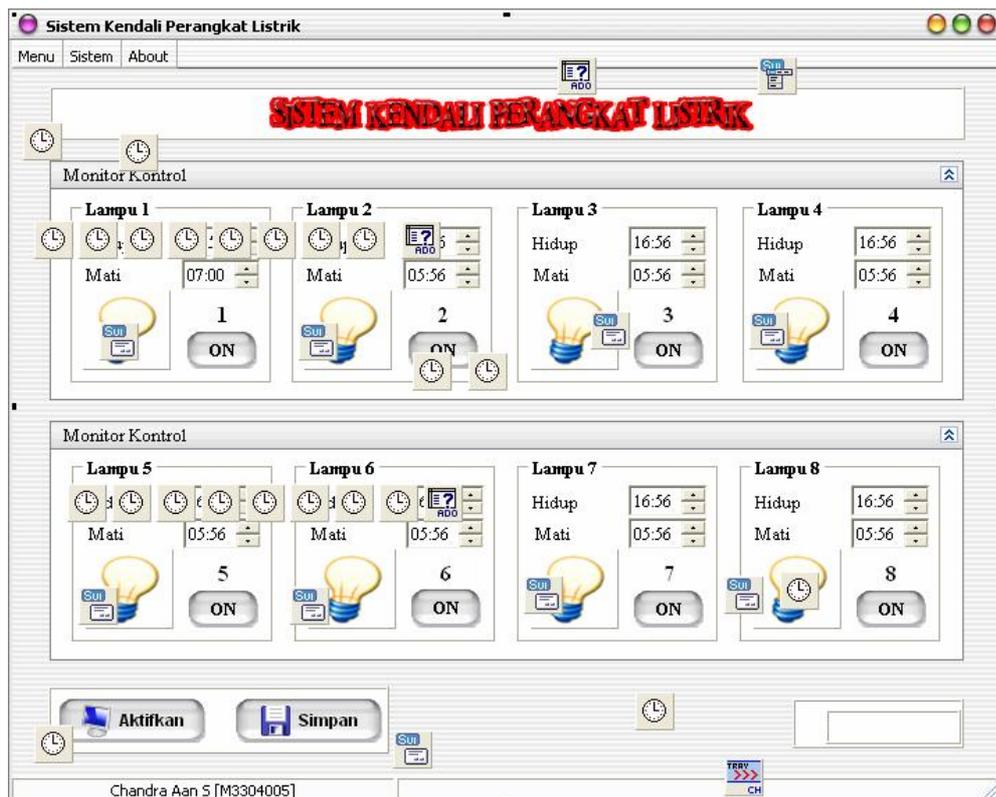
Gambar 4.3. Desain *Splash*

4.2.2. Main Form

Form ini merupakan *form* utama dari aplikasi sistem kendali perangkat listrik. Gambar 4.4 menunjukkan rancangan dari *Main form*. *Form* ini berisi berdelapan buah panel. Setiap panel memiliki komponen *TDateTime* untuk pengaturan jam hidup dan mati. *Status bar* dibagi menjadi dua bagian, bagian yang sebelah kiri untuk menampilkan nama dan NIM. Sedangkan sebelah kanan untuk menampilkan nama sistem operasi pada komputer.

Selain itu dalam *main form* ini terdapat *mainfilemenu*, yang digunakan untuk memanggil *form* lain. *Main form* memiliki beberapa tombol yaitu:

1. *ON perintah* untuk mengaktifkan *timer device*.
2. *Aktifkan* untuk mengaktifkan timer semua *device* (delapan keluaran).
3. *Non Aktifkan* digunakan mematikan semua *device*.
4. *Simpan* berfungsi untuk menyimpan *setting* waktu.

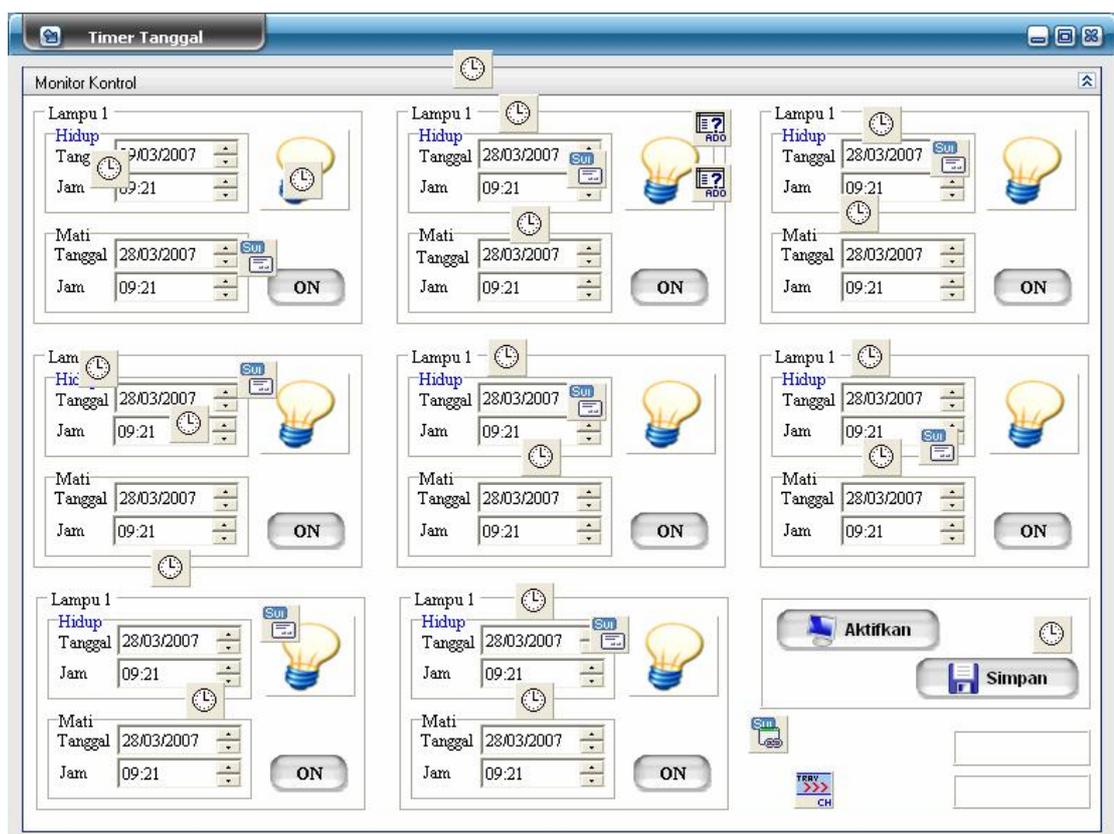


Gambar 4.4 Desain *Main form*

4.2.3. Form FTanggal

Form ini merupakan form yang digunakan untuk mengontrol *ON/OFF device* (perangkat listrik) berdasarkan *setting* waktu dan tanggal dari aplikasi sistem kendali perangkat listrik. Gambar 4.5 menunjukkan rancangan dari form *FTanggal*. Form ini berisi delapan buah panel. Setiap panel memiliki komponen *TDateTime* untuk pengaturan jam hidup dan mati, *TDateTime* bertipe tanggal untuk pengaturan tanggal hidup dan tanggal mati. Selain itu dalam form *FTanggal* memiliki beberapa tombol yaitu:

1. *ON perintah* untuk mengaktifkan *timer device* berdasarkan waktu dan tanggal.
2. *Aktifkan* untuk mengaktifkan *timer* semua *device* (delapan keluaran).
3. *Non Aktifkan* digunakan untuk mematikan *semua device*.
4. *Simpan* berfungsi untuk menyimpan *setting* waktu dan tanggal



Gambar 4.5 Desain form FTanggal

4.2.4. Form Manual

Form ini merupakan *form* yang digunakan untuk mengontrol *ON/OFF device* secara manual (langsung) atau tidak menggunakan fasilitas *timer*. Gambar 4.6 menunjukkan rancangan dari *Form Manual*. *Form* ini berisi berisi delapan buah panel. Setiap panel memiliki komponen tombol *ON/OFF* untuk mengaktifkan dan menonaktifkan *device*.

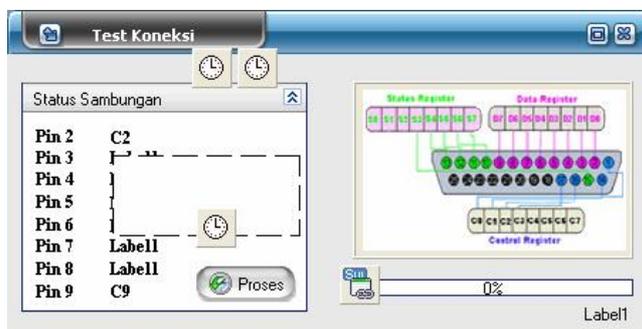


Gambar 4.6 *Form* manual

4.2.5. Form TestKoneksi

Form TestKoneksi ini merupakan sebuah *form* yang digunakan untuk pengecekan status sambungan pada *port* paralel. Pada *form* ini tombol proses digunakan untuk melakukan pengecekan. Gambar 4.7 menunjukkan rancangan dari *form* TestKoneksi.

Apabila pin 2-9 dari *port* paralel dalam kondisi aktif atau menyala maka didapat keterangan *ON*, sedangkan jika dalam kondisi mati maka keterangannya *OFF*.



Gambar 4.7. *Form* TestKoneksi

4.2.6. *Form EditDenah*

Form EditDenah ini merupakan sebuah *form* yang digunakan untuk *mengedit* atau merubah daftar nama ruangan. Pada *form* ini tombol *simpan* digunakan untuk menyimpan nama ruang ke *database*, sedangkan tombol *cancel* adalah tombol keluar dari *form*. Gambar 4.8 menunjukkan rancangan dari *form TestKoneksi*.

Gambar 4.8. *Form TestKoneksi*

4.2.7. *Form About*

Form about ini digunakan untuk menampilkan profil pembuat.



Gambar 4.9 *Form About*

4.3. Perancangan *Database*

Sistem kendali perangkat listrik ini dirancang menggunakan *database* untuk menyimpan setting waktu, tanggal, serta nama ruang. *Database* dibuat menggunakan Microsoft Access XP. Dalam sistem ini *database* dibagi untuk beberapa kebutuhan blok program, antara lain :

4.3.1. Main Form

Main form menggunakan *database* untuk menyimpan *setting* waktu hidup dan mati. beberapa tabel yang digunakan untuk menyimpan waktu hidup dan mati, antara lain :

A. Tabel THidup

Tabel TTglHidup merupakan tabel yang berfungsi untuk menyimpan nilai *setting* dari waktu (jam) hidup. Tabel 4.1 menunjukkan *field* dari tabel THidup.

Tabel 4.1 Tabel THidup

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
1	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 2 (D0)
2	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 3 (D1)
3	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 4 (D2)
4	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 5 (D3)
5	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 6 (D4)
6	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 7 (D5)
7	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 8 (D6)
8	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 9 (D7)

B. Tabel TMati

Tabel TMati merupakan tabel yang berfungsi untuk menyimpan nilai *setting* dari waktu (jam) mati. Tabel 4.2 menunjukkan *field* dari tabel TMati.

Tabel 4.2 Tabel TMati

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
1	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 2 (D0)
2	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 3 (D1)
3	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 4 (D2)
4	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 5 (D3)
5	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 6 (D4)
6	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 7 (D5)
7	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 8 (D6)
8	<i>Date/Time</i>	<i>Setting</i> waktu mati untuk pin ke 9 (D7)

4.3.2. Form FTanggal

Form Ftanggal menggunakan *database* untuk menyimpan *setting* waktu hidup dan mati, serta tanggal hidup dan mati. Tabel untuk menyimpan *setting* tersebut antara lain :

A. Tabel TTglHidup

Tabel TTglHidup merupakan tabel yang berfungsi untuk menyimpan nilai *setting* waktu dan tanggal hidup. Tabel 4.3 menunjukkan *field* dari tabel TTglHidup.

Tabel 4.3 Tabel TTglHidup

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
T1	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 2 (D0)
T2	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 3 (D1)
T3	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 4 (D2)
T4	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 5 (D3)
T5	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 6 (D4)
T6	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 7 (D5)
T7	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 8 (D6)
T8	<i>Date/Time</i>	<i>Setting</i> waktu hidup untuk pin ke 9 (D7)
TTgl1	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 2 (D0)
TTgl2	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 3 (D1)
TTgl3	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 4 (D2)
TTgl4	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 5 (D3)
TTgl5	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 6 (D4)
TTgl6	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 7 (D5)
TTgl7	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 8 (D6)
TTgl8	<i>Date/Time</i>	<i>Setting</i> tanggal hidup untuk pin ke 9 (D7)

B. Tabel TTglMati

Tabel TTglMati merupakan tabel yang berfungsi untuk menyimpan nilai *setting* waktu dan tanggal mati. Tabel 4.4 menunjukkan *field* dari tabel TTglMati.

Tabel 4.4 Tabel TTglMati

<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
<i>T1</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 2 (D0)</i>
<i>T2</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 3 (D1)</i>
<i>T3</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 4 (D2)</i>
<i>T4</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 5 (D3)</i>
<i>T5</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 6 (D4)</i>
<i>T6</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 7 (D5)</i>
<i>T7</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 8 (D6)</i>
<i>T8</i>	<i>Date/Time</i>	<i>Setting waktu mati untuk pin ke 9 (D7)</i>
<i>TTgl1</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 2 (D0)</i>
<i>TTgl2</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 3 (D1)</i>
<i>TTgl3</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 4 (D2)</i>
<i>TTgl4</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 5 (D3)</i>
<i>TTgl5</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 6 (D4)</i>
<i>TTgl6</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 7 (D5)</i>
<i>TTgl7</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 8 (D6)</i>
<i>TTgl8</i>	<i>Date/Time</i>	<i>Setting tanggal mati untuk pin ke 9 (D7)</i>

4.3.3. Form EditDenah

Form EditDenah menggunakan *database* untuk menyimpan nama ruang, Tabel 4.5 menunjukkan *field* dari tabel TDenah.

Tabel 4.5 Tabel TDenah

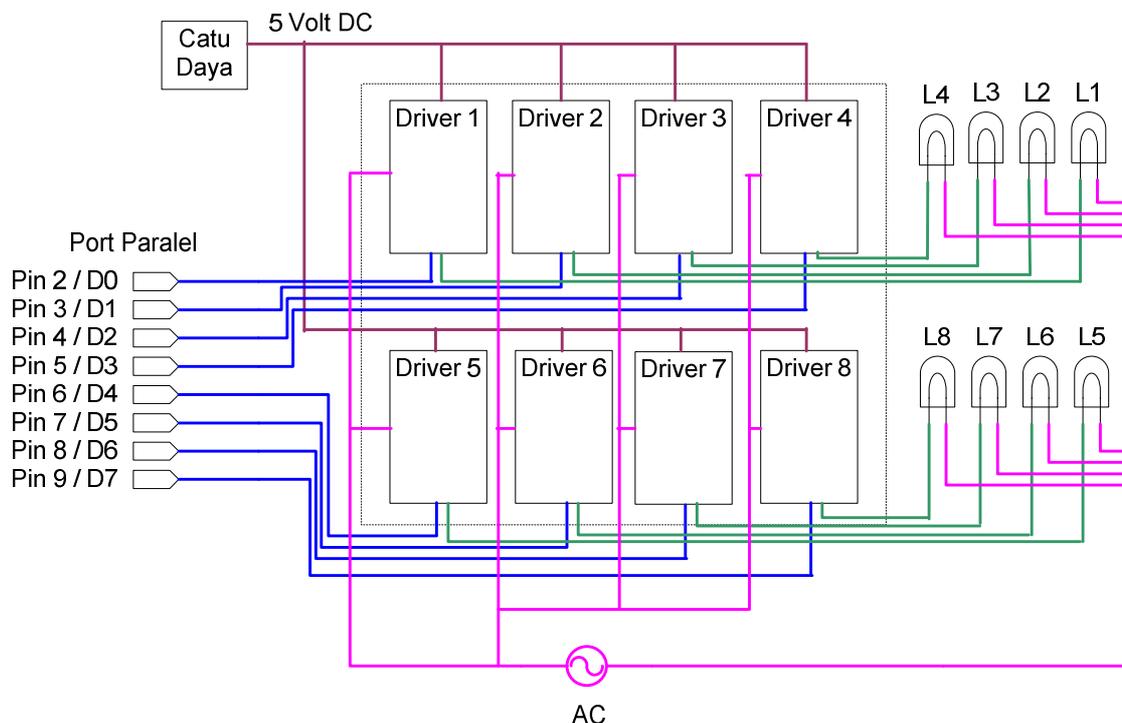
<i>Field</i>	<i>Type</i>	<i>Keterangan</i>
Kode *	<i>Text</i>	Sebagai <i>primary key</i>
1	<i>Text</i>	Nama ruang untuk <i>device</i> 1 (pin 2)
2	<i>Text</i>	Nama ruang untuk <i>device</i> 2 (pin 3)
3	<i>Text</i>	Nama ruang untuk <i>device</i> 3 (pin 4)
4	<i>Text</i>	Nama ruang untuk <i>device</i> 4 (pin 5)
5	<i>Text</i>	Nama ruang untuk <i>device</i> 5 (pin 6)
6	<i>Text</i>	Nama ruang untuk <i>device</i> 6 (pin 7)
7	<i>Text</i>	Nama ruang untuk <i>device</i> 7 (pin 8)
8	<i>Text</i>	Nama ruang untuk <i>device</i> 8 (pin 9)

4.4. Evaluasi Pengujian Sistem

4.4.1 Pengujian Perangkat Keras

4.4.1.1 Pengujian *Driver* Saklar Elektronik

Sistem kendali perangkat listrik ini mempunyai 8 (delapan) keluaran yang diuji menggunakan perangkat listrik lampu pijar masing-masing sebesar 10 watt, sehingga dibutuhkan juga *driver* saklar elektronik sebanyak 8 (delapan). Kedelapan *driver* akan aktif apabila mempunyai masukan maksimum 5 volt DC dari komputer. Gambar 4.10 menunjukkan skema *driver* saklar dengan delapan keluaran yang akan dihubungkan ke perangkat listrik, sedangkan Tabel 4.6 menunjukkan pengujian *driver* saklar dengan masukan 0 volt dan 5 volt dengan *device* lampu pijar.



Gambar 4.10 *Driver* saklar elektronik dengan delapan keluaran

Berdasarkan gambar 4.10 masukan berasal dari *port* paralel yang nantinya akan menghasilkan nilai *output* 5 volt dan 0 volt. Apabila *port* paralel mempunyai nilai 5 volt maka akan

mengaktifkan *driver* dan menyalakan *device* lampu, begitu juga sebaliknya apabila *port* bernilai 0 volt maka *device* lampu akan mati.

Pengujian *driver* saklar elektronik digunakan tegangan dari catu daya yang bernilai 5 volt dan 0 volt, hasil respon dari masing-masing *device* lampu dengan masukan 0 dan 5 volt terdapat dalam Tabel 4.6.

Tabel 4.6 Pengujian *device* dengan masukan 0 volt dan 5 volt

<i>Device</i>	Respon	
	Masukan 0 volt	Masukan 5 volt
Lampu 1	Mati	Nyala
Lampu 2	Mati	Nyala
Lampu 3	Mati	Nyala
Lampu 4	Mati	Nyala
Lampu 5	Mati	Nyala
Lampu 6	Mati	Nyala
Lampu 7	Mati	Nyala
Lampu 9	Mati	Nyala

4.4.1.2 Pengujian *Port* Paralel

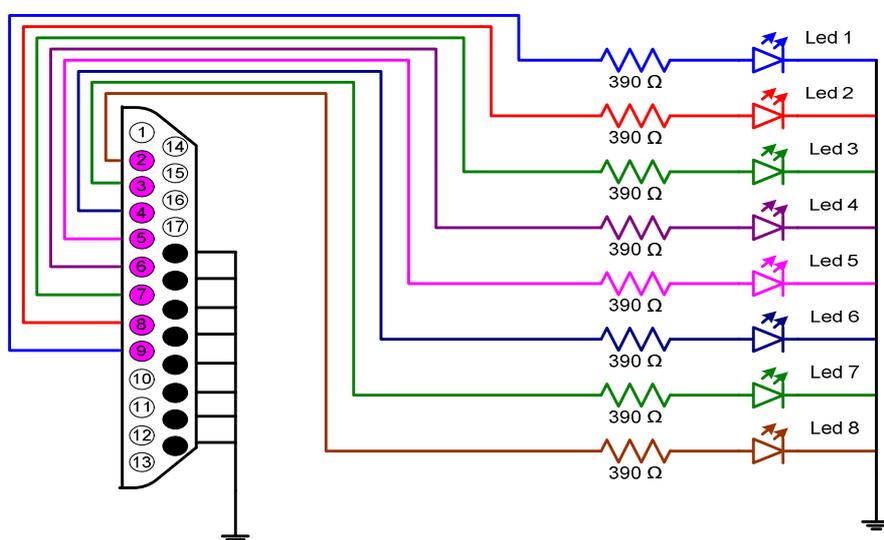
Pengujian *port* paralel menggunakan komponen led sebagai media pengganti *driver* saklar elektronik. Hal ini dimaksudkan untuk melakukan tes alamat pada pin ke 2-9, sebelum dirangkaikan ke blok *driver* saklar elektronik (8 *driver*). Fungsi yang digunakan untuk mempresentasikan nilai biner ke desimal yang digunakan sebagai alamat pengiriman data dari *port* paralel adalah sebagai berikut :

$$\text{Nilai desimal} = 2^{bit7} + 2^{bit6} + 2^{bit5} + 2^{bit4} + 2^{bit3} + 2^{bit2} + 2^{bit1} + 2^{bit0}$$

Berdasarkan fungsi diatas dalam pemrograman untuk mengakses *port* paralel alamat yang mempunyai nilai logika 1 (hidup) mempunyai nilai desimal +128, +64, +32, +16, +8, +4,

+2, +1, sehingga jika kedelapan nilai dijumlahkan menjadi +255 (mengaktifkan semua pin). Sedangkan untuk logika 0 (mati) mempunyai nilai desimal -128, -64, -32, -16, -8, -4, -2, -1, sehingga jika dijumlahkan menjadi -255 (mematikan semua pin).

Pin *port* paralel untuk mengaktifkan kedelapan led yaitu dari pin ke 2-9, dan pin ke 18-25 sebagai *ground* dapat dilihat pada gambar 4.11.



Gambar 4.11 Skema pin *port* paralel dan led

Pengujian *port* paralel digunakan program yang dibuat dengan delphi seperti Gambar 4.12, dengan metode mengirimkan alamat *bit* pada *port* paralel pada pin 2-9.



Gambar 4.12 Program tes *port*

Hasil pengujian *port* paralel menggunakan program *test port* dengan beberapa pengiriman nilai dapat dilihat pada Tabel 4.7 dengan asumsi tombol nyalakan mempunyai nilai *bit* plus(+) dan tombol matikan nilai *bit*nya bernilai minus(-).

Tabel 4.7. Pengujian *port* paralel

Nilai	Tombol Nyalakan								Tombol Matikan							
	L1	L2	L3	L4	L5	L6	L7	L8	L1	L2	L3	L4	L5	L6	L7	L8
1	P	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
2	-	P	-	-	-	-	-	-	-	-	-	-	-	-	-	-
4	-	-	P	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	P	P	-	-	-	-	-	-	-	-	-	-	-	-	-
10	-	P	-	P	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	P	-	-	-	-	-	-	-	-	-	-	-
24	-	-	-	P	P	-	-	-	-	-	-	-	-	-	-	-
32	-	-	-	-	-	P	-	-	-	-	-	-	-	-	-	-
56	-	-	-	P	P	P	-	-	-	-	-	-	-	-	-	-
64	-	-	-	-	-	-	P	-	-	-	-	-	-	-	-	-
72	-	-	-	P	-	-	P	-	-	-	-	-	-	-	-	-
86	-	P	P	-	P	-	P	-	-	-	-	-	-	-	-	-
128	-	-	-	-	-	-	-	P	-	-	-	-	-	-	-	-
255	P	P	P	P	P	P	P	P	-	-	-	-	-	-	-	-

Keterangan :

L = led (Led 1 sampai led 8)

- = led mati

P = led nyala

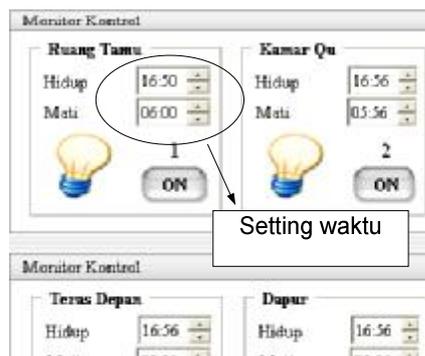
4.4.2 Pengujian Sistem Kendali Perangkat Listrik

Sistem dapat berjalan dengan baik atau tidak, maka dilakukan pengujian *sistem* secara keseluruhan dengan menggabungkan perangkat lunak dan perangkat keras. Pengujian dilakukan dengan cara memberikan perintah kemudian mengamati keberhasilan atas perintah yang diberikan.

4.4.2.1 Main Form

Main Form merupakan sistem otomatisasi perangkat listrik menggunakan *timer* waktu. Sedangkan *device* yang digunakan adalah lampu pijar 10 watt. Gambar 4.13

Menunjukkan *setting* waktu dan Tabel 4.8 adalah hasil pengujian kedelapan *device* dengan *settingan* waktu hidup dan mati.



Gambar 4.13 *Setting timer* waktu

Tabel 4.8 Hasil pengujian sistem *timer* waktu

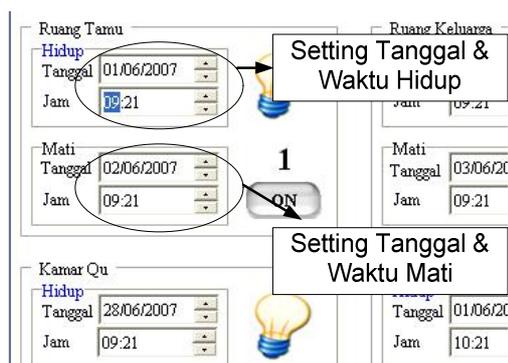
Device	Waktu Hidup	Respon	Waktu Mati	Respon
1	18:00	Nyala	05:00	Mati
2	17:00	Nyala	06:30	Mati
3	17:30	Nyala	06:00	Mati
4	19:20	Nyala	04:30	Mati
5	20:00	Nyala	00:00	Mati
6	07:00	Nyala	09:00	Mati
7	08:00	Nyala	08:30	Mati
8	13:00	Nyala	14:30	Mati

Berdasarkan hasil Tabel 4.8 perangkat listrik akan merespon berdasarkan *setting* waktu hidup atau mati, tentunya respon akan didapat jika *setting* waktu tersebut mempunyai nilai yang sama dengan jam di komputer.

4.4.2.2 *Form* FTanggal

Form FTanggal merupakan sistem otomatisasi perangkat listrik menggunakan *timer* waktu dan tanggal. Sedangkan *device* yang digunakan adalah lampu pijar 10 watt. Gambar 4.14

Menunjukkan *setting* waktu dan tabel 4.14 adalah hasil pengujian kedelapan *device* dengan *settingan* tanggal hidup dan mati, serta waktu hidup dan mati. Dalam *form* ini *setting* tanggal mati minimal harus lebih satu hari dari tanggal hidup.



Gambar 4.14 *Setting* tanggal dan waktu

Tabel 4.9 Hasil pengujian sistem *timer* tanggal dan waktu

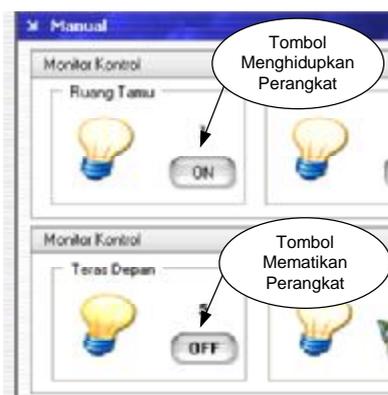
Device	Tanggal Hidup	Waktu Hidup	Respon	Tanggal Mati	Waktu Mati	Respon
1	31/05/2007	18:00	Nyala	01/06/2007	05:00	Mati
2	31/05/2007	17:00	Nyala	31/06/2007	18:30	Mati
3	01/06/2007	17:30	Nyala	05/06/2007	06:00	Mati
4	02/06/2007	19:20	Nyala	03/06/2007	04:30	Mati
5	05/06/2007	20:00	Nyala	06/06/2007	00:00	Mati
6	05/06/2007	07:00	Nyala	06/06/2007	09:00	Mati
7	06/06/2007	08:00	Nyala	08/06/2007	08:30	Mati
8	10/06/2007	13:00	Nyala	20/06/2007	14:30	Mati

Berdasarkan hasil Tabel 4.9 perangkat listrik akan merespon berdasarkan *setting* tanggal dan waktu (hidup atau mati), tentunya respon akan didapat jika *setting* tanggal dan waktu mempunyai nilai yang sama dengan tanggal dan jam di komputer.

4.4.2.3 *Form* FManual

Form FManual merupakan sistem menyalakan atau mematikan perangkat listrik tombol *ON/OFF*. Sedangkan *device*

(perangkat listrik) yang digunakan adalah lampu pijar 10 watt. Gambar 4.15 Menunjukkan *setting* waktu dan tabel 4.15 adalah hasil pengujian kedelapan *device* dengan *settingan* tanggal hidup dan mati, serta waktu hidup dan mati.



Gambar 4.15 Tombol *ON/OFF* pada *form* FManual

Tabel 4.10 Hasil pengujian sistem *form* FManual

Device	Respon	
	Tombol <i>ON</i>	Tombol <i>OFF</i>
1	Nyala	Mati
2	Nyala	Mati
3	Nyala	Mati
4	Nyala	Mati
5	Nyala	Mati
6	Nyala	Mati
7	Nyala	Mati
8	Nyala	Mati

Berdasarkan hasil Tabel 4.10 perangkat listrik akan merespon berdasarkan tombol yang dieksekusi, jika tombol *ON* maka perangkat listrik akan menyala, begitu juga sebaliknya jika tombol *OFF* yang dieksekusi maka perangkat listrik juga akan mati.

4.5. Keunggulan dan Kelemahan Sistem

4.5.1 Keunggulan

Sistem yang dapat di implementasikan dalam kehidupan sehari-hari, terdapat beberapa keunggulan atau kemudahan dari rancangan sistem yang dicapai, antara lain :

- a. Sistem kendali perangkat listrik ini dirancang berbasis GUI (*Graphic User Interface*), sehingga pengoperasian sistem sangat mudah.
- b. Sistem ini mempunyai cara kerja otomatisasi perangkat listrik menggunakan *timer*. Otomatisasi sistem terintegrasi kedalam tiga bagian, yaitu otomatisasi berdasarkan waktu (hidup atau mati), otomatisasi berdasarkan tanggal dan waktu (hidup atau mati), serta otomatisasi secara manual (perintah *ON/OFF* langsung). Sehingga untuk menghidupkan atau mematikan perangkat listrik tidak perlu bersusah-payah beranjak dari tempat kerja, dan secara tidak langsung sistem akan memberikan efisiensi terhadap pemakaian perangkat listrik.
- c. Sistem kendali perangkat listrik dapat di implementasikan pada perangkat listrik AC (*Alternating Current*), seperti lampu pijar, kipas angin, televisi, komputer, dan perangkat listrik lainnya.
- d. *Driver* saklar elektronik yang digunakan untuk menghubungkan PC dengan perangkat listrik dibuat menggunakan sistem *opto-triac* yang memiliki tingkat respon dan kestabilan sistem yang tinggi.

4.5.2 Kelemahan

Meskipun banyak kemudahan yang diperoleh dari hasil rancangan sistem kendali perangkat listrik ini, tentunya masih banyak kelemahan antar lain :

- a. Sistem yang dirancang masih bersifat *single user*, belum dapat digunakan sebagai sistem *client-server*.
- b. Karena sistem berjalan dalam PC, sehingga untuk memperoleh kegunaan dari sistem yaitu otomatisasi pada perangkat listrik, *driver*

saklar harus selalu terhubung dengan PC dan sistem program dijalankan (dieksekusi). Oleh sebab itu apabila PC tidak dinyalakan atau mati maka sistem juga tidak dapat berjalan sebagaimana mestinya.

- c. Sistem dirancang menggunakan pemrograman yang menggunakan *interface port* paralel, sehingga apabila dalam PC tidak terdapat komponen *port* paralel sistem tidak dapat diterapkan.

BAB V

PENUTUP

5.1. Kesimpulan

Setelah melakukan analisis dan pembahasan yang telah diuraikan pada bab sebelumnya, maka dapat diambil beberapa kesimpulan antara lain :

1. Telah selesai perancangan *hardware* dan *software* sistem kendali perangkat listrik menggunakan PC melalui *port* parallel. *Software* menggunakan GUI (*Graphic user Interface*) yang bersifat *user friendly*.
2. Dalam sistem kendali perangkat listrik, menggunakan prinsip otomatisasi menggunakan *timer* untuk menyalakan *device* perangkat listrik AC (*Alternating Current*). Penghubung antara PC dan perangkat listrik, menggunakan *driver* saklar elektronik (menggunakan *optocoupler* dan *triac*), yang dapat bekerja dengan baik apabila mendapat masukan +5 volt DC dari komputer.
3. Sistem kendali perangkat listrik ini dirancang menggunakan *database*. *Database* ini digunakan untuk menyimpan waktu hidup dan mati, serta tanggal hidup dan mati, selain itu juga untuk menyimpan nama ruang.

5.2. Saran

Berdasarkan permasalahan yang ada pada sistem kendali perangkat listrik yang dirancang, terdapat beberapa saran yaitu :

1. Aplikasi ini dapat dikembangkan untuk otomatisasi *device* (perangkat listrik) lebih dari 8 (delapan).
2. Apabila perangkat listrik yang digunakan adalah lampu pijar, dapat dikembangkan lagi sehingga lampu bisa diatur terang redupnya.
3. Sistem dapat dikembangkan sebagai aplikasi *client-serve*, sehingga pengontrolan dapat dilakukan dengan beberapa komputer.

DAFTAR PUSTAKA

- Anonim. 2005. <http://electroniqlab.com>.
- Anonim. 2007. <http://elektronika-elektronika.blogspot.com>.
- Anwar, M. K. 2003. *Aplikasi Sistem Informasi Manajemen bagi Pemerintah di Era Otonomi Daerah*. Malang.
- Blocher, R. 2003. *Dasar Elektronika*. Andi Yogyakarta. Yogyakarta.
- Budiharto, W. 2004. *Interfacing Komputer dan Mikrokontroler*. Elex Media Komputindo. Jakarta.
- Fatansyah. 2002. *Basis Data*. CV, Informatika. Bandung.
- Jogiyanto, H.M. 2001. *Analisa dan Desain Sistem Informasi, edisi kedua*. Andi Offset. Yogyakarta.
- Kadir, A. 2004. *Pemrograman Database dengan Delphi 7 Menggunakan Access dan ADO*. Andi Offset. Yogyakarta.
- Malvino. 1994. *Prinsip Prinsip elektronik*. Erlangga. Jakarta.
- Sutadi, D. 2002. *I / O Bus & Motherboard*. Andi Offset. Yogyakarta.

LAMPIRAN

Foto *Hardware* Sistem Kendali Perangkat Listrik



Driver Saklar Elektronik



Device 8 Lampu

LAMPIRAN

Perancangan *software* sistem kendali perangkat listrik dibuat menggunakan Delphi 7.0, untuk mempermudah pengaksesan *port* paralel digunakan file *library* “*inpout32.dll*”. *Procedure* untuk mengakses file “*inpout32.dll*” :

```
Procedure Out32 (PortAddress:smallint;Value:smallint);stdcall;
external'inpout32.dll';
function Inp32 (PortAddress:smallint):smallint;stdcall;external
'inpout32.dll';
```

Beberapa *procedure* yang terdapat dalam *main form* antara lain :

1. *Procedure timer* untuk menghidupkan *device* :

```
procedure TMain.THidup1Timer(Sender: TObject);
begin
if Jnyala1=strtotime (panel3.Caption) then
begin
datalampu:=datalampu+1;
out32 ($378,datalampu);
Thidup1.Enabled:=false;
lmp1.Glyph.LoadFromFile (extractfilepath (application.ExeName)+'\hid
up.bmp');
lmp1.Hint:='Nyala';
Tmatil.Enabled:=true;
end
end;
```

2. *Procedure timer* untuk mematikan *device* :

```
procedure TMain.TMatilTimer(Sender: TObject);
begin
if JMatil=strtotime (panel3.Caption) then
begin
datalampu:=datalampu-1;
out32 ($378,datalampu);
lmp1.Glyph.LoadFromFile (extractfilepath (application.ExeName)+'\mat
i.bmp');
lmp1.Hint:='Mati';
Thidup1.Enabled:=true;
TMatil.Enabled:=false;
end
end;
```

3. *Procedure tombol ON*

```
procedure TMain.Bt1Click(Sender: TObject);
begin
x:=Inp32 ($378);
if bt1.caption='ON' then
begin
T:=dtmatil.Time-dthidup1.Time;
if (T=strtotime ('00:00')) then
begin
dialog1.Text:='Maaf Setting Waktu'+gbl.caption+' Bermasalah-!Waktu
Hidup dan Waktu Mati tidak Boleh Sama !';
dialog1.ShowModal;
end
else begin
DH1:=timetostr (dthidup1.Time);
JNyalal:=strtotime (DH1);
```

```

    THidup1.Enabled:=true;
    DM1:=timetostr(DtMatil.Time);
    JMatil:=strtotime(DM1);
    dthidup1.Enabled:=false;
    dtmatil.Enabled:=false;
    btnon.Caption:='Non Aktifkan';
    Bt1.Caption:='OFF';
end;
end
else
begin
    if x mod 2=1 then
    begin
        THidup1.Enabled:=False;
        TMatil.Enabled:=false;
        dthidup1.Enabled:=true;
        dtmatil.Enabled:=true;
        datalampu:=datalampu-1;
        out32 ($378,datalampu);
    lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\Mat
    i.bmp');
        Bt1.caption:='ON';
    end
    else
    begin
        THidup1.Enabled:=False;
        TMatil.Enabled:=false;
        dthidup1.Enabled:=true;
        dtmatil.Enabled:=true;
    lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\Mat
    i.bmp');
        Bt1.caption:='ON';
    end;
end;
end;

```

4. Procedure tombol Simpan (untuk menyimpan *setting* waktu)

```

procedure TMain.suiButton2Click(Sender: TObject);
begin
    if (Application.MessageBox(' Anda Yakin Akan Menyimpan
    Setting Waktu Ini ? ',
        ' Perhatian ! ',MB_YESNO or MB_ICONINFORMATION)=IDYES)
    then
    begin
        with QHidup do
        begin
            sql.Clear;
            sql.Add('Update THidup set
            1="'+timetostr(dthidup1.Time)+'",'+2="'+timetostr(dthidup2.
            Time)+'",'+3="'+timetostr(dthidup3.Time)+'",'+4="'+timetos
            tr(dthidup4.Time)+'",'+5="'+timetostr(dthidup5.Time)+'",'+
            6="'+timetostr(dthidup6.Time)+'",'+7="'+timetostr(dthidup7.
            Time)+'",'+8="'+timetostr(dthidup8.Time)+'"');
            execsql;
        end;
        with QMati do
        begin
            sql.Clear;
            sql.Add('Update TMati set
            1="'+timetostr(dtMatil.Time)+'",'+2="'+timetostr(dtMati2.Ti
            me)+'",'+3="'+timetostr(dtMati3.Time)+'",'+4="'+timetostr(

```

```

dtMati4.Time)+'",'+5=""'+timetostr(dtMati5.Time)+'",'+6=""'+
timetostr(dtMati6.Time)+'",'+7=""'+timetostr(dtMati7.Time)+'
",'+8=""'+timetostr(dtMati8.Time)+'""');
execsql;
dlsimpan.Text:=' Seting Eaktu Telah Disimpan ';
dlsimpan.ShowModal;
end;
end;
end;

```

Beberapa *procedure* yang terdapat dalam *form* FTanggal antara lain :

1. *Procedure timer* untuk menghidupkan *device*

```

procedure TFTanggal.THiduplTimer(Sender: TObject);
begin
if
(Jnyala1=strtotime(panel3.Caption)) and (TglNyalal=strtodate(PTanggal.
l.Caption)) then
begin
datalampu:=datalampu+1;
out32 ($378,datalampu);
Thidupl.Enabled:=false;
lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\hid
up.bmp');
Tmatil.Enabled:=true;
end
end;

```

2. *Procedure timer* untuk mematikan *device*

```

procedure TFTanggal.TMatilTimer(Sender: TObject);
begin
if (JMatil=strtotime(panel3.Caption) and
(TglMatil=strtodate(PTanggal.Caption)) then
begin
datalampu:=datalampu-1;
out32 ($378,datalampu);
lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\mat
i.bmp');
THidupl.Enabled:=true;
TMatil.Enabled:=false;
end
end;

```

3. *Procedure tombol ON*

```

procedure TFTanggal.Bt1Click(Sender: TObject);
begin
if bt1.caption='ON' then
Begin
if (dttglhidupl.Date>dttglmatil.Date) then
begin
dialog1.Text:='Maaf Setting untuk '+denah1.caption+' Bermasalah
! Tanggal Mati Harus Lebih Lama dari Tanggal Hidup';
dialog1.ShowModal;
end
else
if (dttglhidupl.Date=dttglmatil.Date) and
(dtjamhidupl.Time>dtjammatil.Time) then
begin
dialog1.Text:='Maaf Setting untuk '+denah1.caption+'
Bermasalah ! Jam Mati Harus Lebih Lama dari Jam Hidup';
dialog1.ShowModal;
end
else

```

```

    if (dttglhidup1.Date=dttglmatil.Date) and
(dtjamhidup1.Time=dtjammatil.Time) then
    begin
        dialog1.Text:='Maaf Setting untuk '+denah1.caption+'
Bermasalah ! Jam Hidup Tidak Boleh Sama Dengan Jam Mati';
        dialog1.ShowModal;
    end
    else
    begin
        DJ1:=timetostr(dtJamHidup1.Time);
        DT1:=datetostr(dttglhidup1.Date);
        JNyalal:=strtotime(DJ1);
        TglNyalal:=strtoDate(DT1);
        THidup1.Enabled:=true;
        DJM1:=timetostr(DtJamMatil.Time);
        DTM1:=Datetostr(DtTglMatil.Time);
        JMatil:=strtotime(DJM1);
        TglMatil:=strtoDate(DTM1);
        Denah1.Enabled:=false;
        Bt1.Caption:='OFF';
        btaktif.Caption:='Non Aktifkan';
        Bt1.Visible:=false;
    end;
    end
    else
    begin
        Denah1.Enabled:=true;
        TMatil.Enabled:=False;
        THidup1.Enabled:=false;
        lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\Mat
i.bmp');
        Bt1.caption:='ON';
    end;
end;
end;

```

4. Procedure tombol Simpan (untuk menyimpan *setting* tanggal dan waktu)

```

procedure TFTanggal.Bt1Click(Sender: TObject);
begin
x:=Inp32($378);
if bt1.caption='ON'then
    Begin
        if (dttglhidup1.Date>dttglmatil.Date)then
        begin
            dialog1.Text:='Maaf Setting untuk '+denah1.caption+' Bermasalah
! Tanggal Mati Harus Lebih Lama dari Tanggal Hidup';
            dialog1.ShowModal;
        end
        else
            if (dttglhidup1.Date=dttglmatil.Date) and
(dtjamhidup1.Time>dtjammatil.Time) then
            begin
                dialog1.Text:='Maaf Setting untuk '+denah1.caption+'
Bermasalah ! Jam Mati Harus Lebih Lama dari Jam Hidup';
                dialog1.ShowModal;
            end
            else
                if (dttglhidup1.Date=dttglmatil.Date) and
(dtjamhidup1.Time=dtjammatil.Time) then
                begin
                    dialog1.Text:='Maaf Setting untuk '+denah1.caption+'
Bermasalah ! Jam Hidup Tidak Boleh Sama Dengan Jam Mati';
                    dialog1.ShowModal;
                end
            end
        end
    end
end;

```

```

else
begin
DJ1:=timetostr(dtJamHidup1.Time);
DT1:=datetostr(dttglhidup1.Date);
JNyalal:=strtotime(DJ1);
TglNyalal:=strtoDate(DT1);
THidup1.Enabled:=true;
DJM1:=timetostr(DtJamMatil.Time);
DTM1:=Datetostr(DtTglMatil.Time);
JMatil:=strtotime(DJM1);
TglMatil:=strtoDate(DTM1);
  dttglhidup1.Enabled:=false;
  dtjamhidup1.Enabled:=false;
  dttglmatil.Enabled:=false;
  dtjammatil.Enabled:=false;
Bt1.Caption:='OFF';
btaktif.Caption:='Non Aktifkan';
  end;
end
else
begin
  if x mod 2=1 then
begin
  THidup1.Enabled:=False;
  TMatil.Enabled:=false;
  dttglhidup1.Enabled:=true;
  dtjamhidup1.Enabled:=true;
  dttglmatil.Enabled:=true;
  dtjammatil.Enabled:=true;
  datalampu:=datalampu-1;
  out32 ($378,datalampu);
  lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\Mat
i.bmp');
  Bt1.caption:='ON';
  end
  else
begin
  THidup1.Enabled:=False;
  TMatil.Enabled:=false;
  dttglhidup1.Enabled:=true;
  dtjamhidup1.Enabled:=true;
  dttglmatil.Enabled:=true;
  dtjammatil.Enabled:=true;
  dtjammatil.Enabled:=true;
  lmp1.Glyph.LoadFromFile(extractfilepath(application.ExeName)+'\Mat
i.bmp');
  Bt1.caption:='ON';
  end;
end;
end;
end;

```