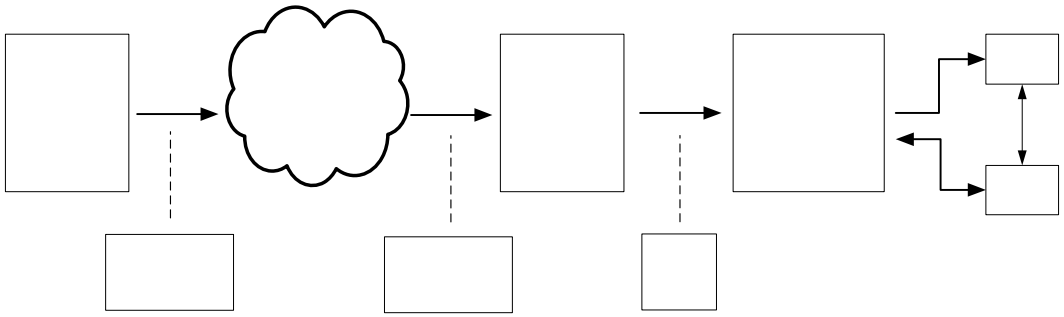
BAB III
PERANCANGAN SISTEM

3.1 Diagram Blok Sistem

Pada bab perancangan ini penulis menggunakan arsitektur jaringan *client/server* yang saling terhubung dengan jaringan LAN melalui *ethernet*. Pengiriman data awal dari komputer *client* ke komputer *server* melalui jaringan LAN kemudian *server* akan mengirim data tersebut melalui RS232 yang akan diterima dan diolah oleh mikrokontroler sehingga dapat mengendalikan lampu gedung. Untuk pengecekan status lampu gedung akan dikirimkan oleh sensor LDR (*Light Dependent Resistor*) yang akan diterima oleh komputer *server* kemudian dikirim ke komputer *client.*

Untuk lebih jelasnya dapat dilihat pada diagram blok seperti terlihat pada gambar 3.1. Dari setiap blok mempunyai fungsi dan cara kerja yang berbeda – beda yang akan membentuk suatu sistem yang diharapkan.



Gambar 3.1. *Diagram Blok Sistem*

.

Prinsip kerja pada diagram blok adalah sebagai berikut :

1. PC *(Personal Computer) client* terdapat aplikasi kendali lampu gedung menggunakan Visual Basic 6.0. PC *client* berfungsi untuk mengirim data ke *server* untuk menghidupkan dan mematikan lampu sekaligus menerima status lampu dari PC *server*.
2. *Ethernet Card* sebagai media transmisi data dari komputer *client* ke komputer *server* maupun sebaliknya.
3. LAN *(Lokal Area Netwrk)* berfungsi sebagai penghubung antara PC *client* dan *server.*

4. PC (P*ersonal Computer) server* terdapat juga aplikasi kendali lampu gedung menggunakan Visual Basic 6.0. PC *serve*r berfungsi untuk menerima data dari PC client untuk menghidupkan dan mematikan lampu sekaligus mengirim status lampu yang diterima dari mikrokontroler ke komputer *client*.
5. RS-232 sebagai alat untuk menghubungkan PC *server* dengan mikrokontroler sehingga dapat berkomunikasi.
6. *Mikrokontroler AT89S51* berfungsi untuk mengolah data dan mengendalikan alat.
7. Sensor LDR memberikan status lampu gedung ke PC *server* melalui RS232 kemudian dikirim ke PC *client.*

3.2. Perancangan Perangkat Keras

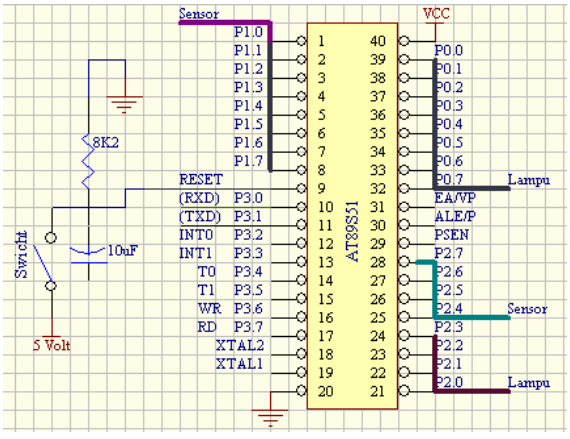
Pada perancangan perangkat keras penulis membagi menjadi dua bagian yaitu perancangan perangkat keras pada mikrokontroler dan perancangan perangkat keras pada jaringan

3.2.1 Perancangan Perangkat Keras Pada Mikrokontroler

Perancangan perangkat keras pada mikrokontroler meliputi Mikrokontroler AT89S51, Rangkaian RS232, Driver Relay, Driver Sensor LDR dan Power Supply.

3.2.1.1 Mikrokontroller AT89S51

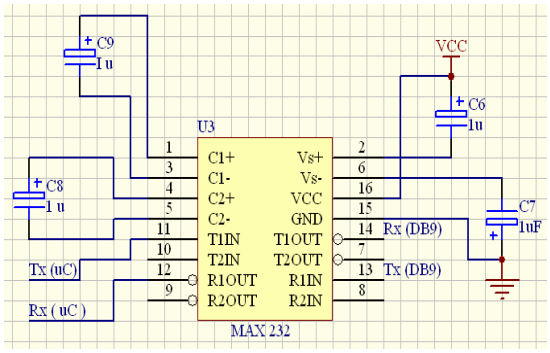
Rangkaian ini merupakan pusat pengolahan data dan pusat pengendalian alat. Data yang diterima oleh PC *server* dari PC *client* akan diolah oleh mikrokontroler sehingga dapat mengendalikan lampu gedung. Mikrokontroler hanya dapat menyalakan dan mematikan lampu gedung yang disertai dengan adanya pembacaan status lampu dari sensor cahaya LDR. Rangkaian Mikrokontroler dapat dilihat pada gambar 3.2



Gambar 3.2. *Rangkaian sistem minimum*

3.2.1.2 Rangkaian RS232

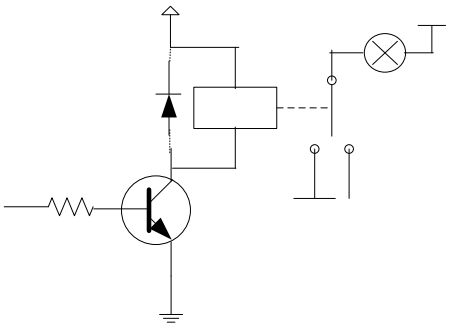
Sebuah komputer dapat berkomunikasi dengan alat menggunakan rangkaian RS232. Rangkaian ini digunakan untuk mengkonversi nilai tegangan yang akan disesuaikan nilai tegangan yang keluar dari komunikasi serial. Komponen utama pada rangkaian ini adalah IC MAX232. IC ini akan mengkonversi tegangan atau kondisi logika TTL dari hardware agar sesuai dengan tegangan pada komputer ataupun sebaliknya. IC MAX232 memiliki 16 pin. Selain itu terdapat juga komponen tambahan yaitu beberapa kapasitor yang di pasang pada pin-pin tertentu dengan kapasitas sebesar 1uF. Komponen ini berfungsi sebagai *charge pump* yang menyuplai muatan ke bagian pengubah tegangan. Ini dioperasikan dengan catu daya 5 Volt.



Gambar 3.3. *Rangkaian RS232*.

3.2.1.3 Driver Lampu

Driver lampu ini berfungsi untuk mengendalikan beban yang dapat digunakan lampu. Jenis transistor pada rangkaian ini menggunakan tipe NPN (2N3904). Rangkaian ini dikendalikan langsung oleh mikrokontroler. Port yang digunakan yaitu port P0 (pin P0.0,P0.1,P0.2,P0.3,P0.4,P0.5,P0.6 dan P0.7) dan port P2 (pin P2.0,P2.1,P2.2 dan P2.3) untuk menyalakan dan mematikan lampu.



Gambar 3.4. *Driver Lampu*

Pada perancangan driver relay ini penulis menggunakan perhitungan untuk mendapatkan nilai hambatan pada basis (R_B). Berikut adalah perhitunganya :

Mencari nilai R_B pada transistor :

$$R_B = \frac{V_B - V_{BE}}{I_B} = \frac{5V - 0{,}85V}{1mA}$$

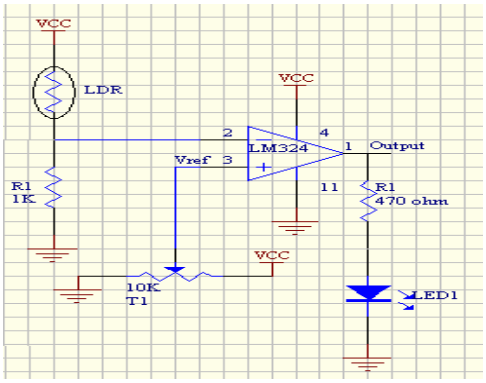
$$R_B = \frac{4{,}15V}{1mA} = 4{,}15K\ Ohm$$

Hasil dari perhitungan nilai R_B mendekati nilai 4K7, Sehingga dalam rangkaian relay ini penulis mengggunakan hambatan pada basis sebesar adalah 4K7.

Keterangan : I_B = Arus basis diambil dari data sheet transistor 2N3904

V_{BE} = Tegangan saturasi pada Basis-Emiter di ambil dari data sheet transistor 2N3904.

3.2.1.4 Sensor LDR (*Light Dependent Resistor*)

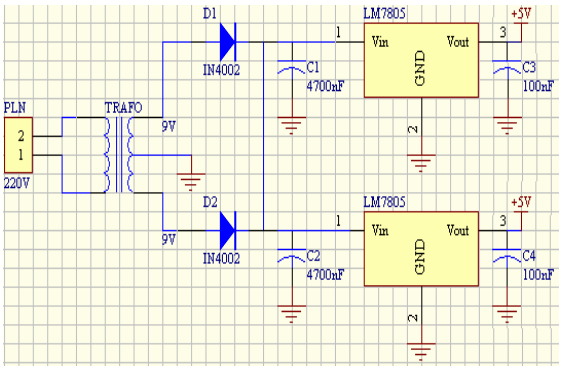


Gambar 3.5. *Driver Sensor LDR*

Rangkaian driver sensor LDR ini berfungsi untuk membaca cahaya sebagai status lampu yang diterima oleh PC *server* untuk kemudian dikirim ke PC *client*. Setiap lampu dipasang masing-masing satu LDR. Pembuatan driver sensor ini menggunakan rangkaian komparator, yaitu menggunakan IC LM324. Bila sensor terkena cahaya terang maka keluaran dari komparator akan berlogika low (0), dan jika sensor tidak terkena cahaya maka keluaran dari komparator berlogika high (1). Pada gambar 3.5 merupakan skematik rangkaian sensor LDR menggunakan komparator IC LM324. Rangkaian sensor ini dikendalikan langsung oleh mikrokontroler.

3.2.1.5 Power Supply (Sumber tegangan)

Rangkaian power supply atau catu daya memberikan supply tegangan pada alat pengendali. Rangkaian ini masih mendapatkan sumber tegangan sebesar 220 V AC (tegangan bolak- balik). Tegangan ini akan diturunkan melalui trafo sebagai penurun tegangan. Kemudian tegangan tersebut akan disearahkan oleh dioda menjadi tegangan DC (tegangan searah). Keluaran dari dioda akan masuk ke IC regulator yang akan menstabilkan tegangan. Pada rangkaian catu daya terdapat kapasitor yang berfungsi membuang noise pada tegangan DC.



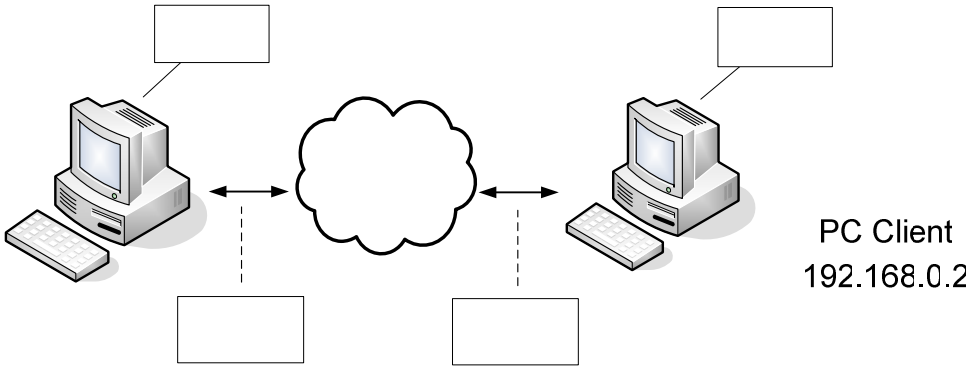
Gambar 3.6. *Rangkaian Power Supply*

3.2.2 Perancangan Hardware Pada Jaringan

Untuk perancangan hardware jaringan meliputi 2 buah PC *(Personal Computer)*, 2 buah Ethernet Card, kabel UTP (*Unishield Tranmission Pair*) dan *Connector* RJ-45.

3.2.2.1 PC (*Personal Computer*)

Pada perancangan hardware jaringan penulis menggunakan dua buah komputer yang terdiri dari PC *client* dan PC *server* dan saling terhubung melalui jaringan LAN. Masing-masing PC terdapat program aplikasi Visual Basic. Proses pertukaran data melalui *ethernet* dan untuk komunikasi antara *client* dan *server* melalui Winsock *(Windows Socket)* yang tersedia pada Visual Basic 6.0. Untuk sistem operasi menggunakan sistem operasi *Microsoft® Windows XP Professional* pada *client* dan *Microsoft® Windows 2000* pada PC *server*. Pada PC *client* menggunakan IP 192.168.0.2 dan PC *server* menggunakan IP 192.168.0.1. Gambar 3.6 dapat lihat PC *client* dan PC *server* saling terkoneksi.



Gambar 3.7. *Client dan server saling terkoneksi*

3.2.2.2 UTP (*Unshielded Twisted Pair*)

Media transmisi data yang digunakan pada tugas akhir ini menggunakan kabel. Jenis kabel yang digunakan adalah kabel UTP *(Unishield Twisted Pair)* yang sering di pakai pada sistem jaringan. Penggunaan kabel berdasarkan yang dikeluarkan oleh *Telecommunications Industri Association / Electronic Industry Association (TIA/EIA)* yaitu tipe 568A dan tipe 568B. Untuk penyambungan kabel *client* dan *server* menggunakan teknik penyambungan *cross–over* (silang). Ujung kabel satu menggunakan tipe 568A dengan urutan warna yang terlihat pada tabel 3.1 (a). Dan ujung kabel satunya menggunakan tipe 568B dengan urutan warna yang terlihat pada tabel 3.1 (b). Kedua ujung kabel tersebut di hubungkan pada konektor RJ-45. Kemudian dipasang pada masing-masing LAN Card yang tersedia pada kedua PC.

Ethernet

Pin	Warna
1	Putih Hijau
2	Hijau
3	PutihOrange
4	Biru
5	Putih biru
6	Orange
7	Putih Coklat
8	Coklat

(a)

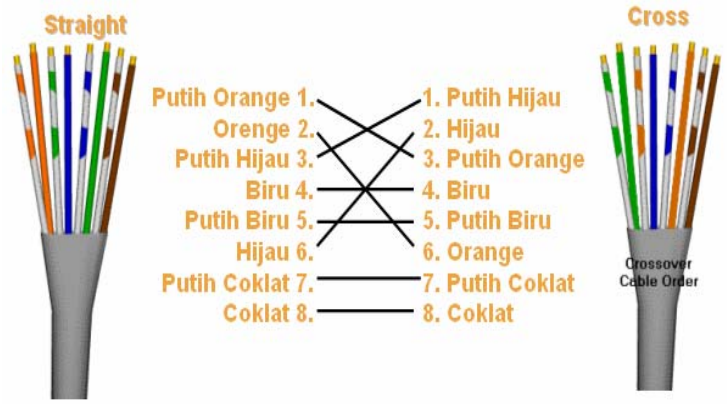
Tabel 3.1. (a)*Kabel UTP Tipe 568A*

Pin	Warna
1	Putih Orange
2	Orange
3	Putih Hijau
4	Biru
5	Putih biru
6	Hijau
7	Putih Coklat
8	Coklat

(b)

Tabel 3.1. *(b)Kabel UTP Tipe 568B*

Pada gambar 3.8 terlihat konfigurasi urutan warna kabel UTP tipe 568A dan 568B.



Tipe 568A Tipe 568B

Gambar 3.8. *Konfigurasi urutan warna kabel UTP secara silang (cross)*

3.2.2.3 Ethernet Card atau Kartu Jaringan

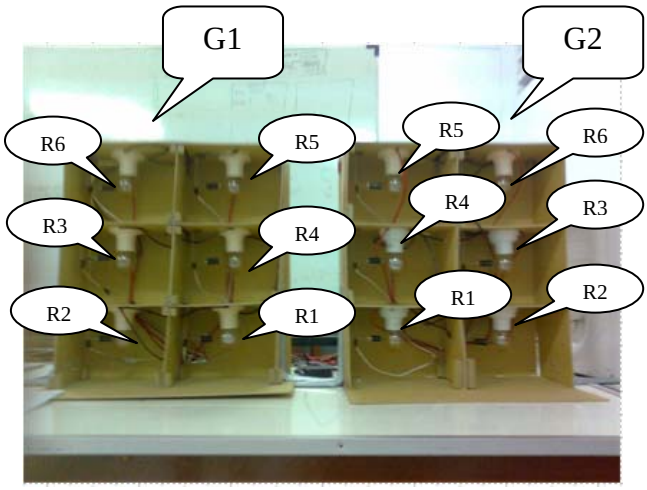
Kartu jaringan atau *Network Interface Card* (NIC) merupakan komponen utama dari sebuah komputer yang berfungsi sebagai media pertukaran data antara komputer *client* dan *server* melalui suatu jaringan yang terintegrasi. Kartu ini berupa sebuah kartu ekspansi yang dipasang salah satu slot ekspansi pada mainboard komputer. Jenis NIC yang digunakan yaitu SIS 900 PCI Fast Ethernet. Gambar 3.9 menunjukkan gambar *Network Interface Card* (NIC).



Gambar 3.9. *Network Interface Card (NIC)*

3.2.2.4 Bentuk Gedung

Pada perancangan penulis membuat 2 gedung masing-masing gedung dengan 6 ruangan dan disertai dengan sensor LDR pada setiap ruangan. Pada gedung1 terdapat ruang1, ruang2, ruang3, ruang4, ruang5, dan ruang6. Untuk gedung2 juga memiliki ruangan seperti gedung1 dengan nama yang sama. Pada gambar 3.10 terlihat bentuk gedung.



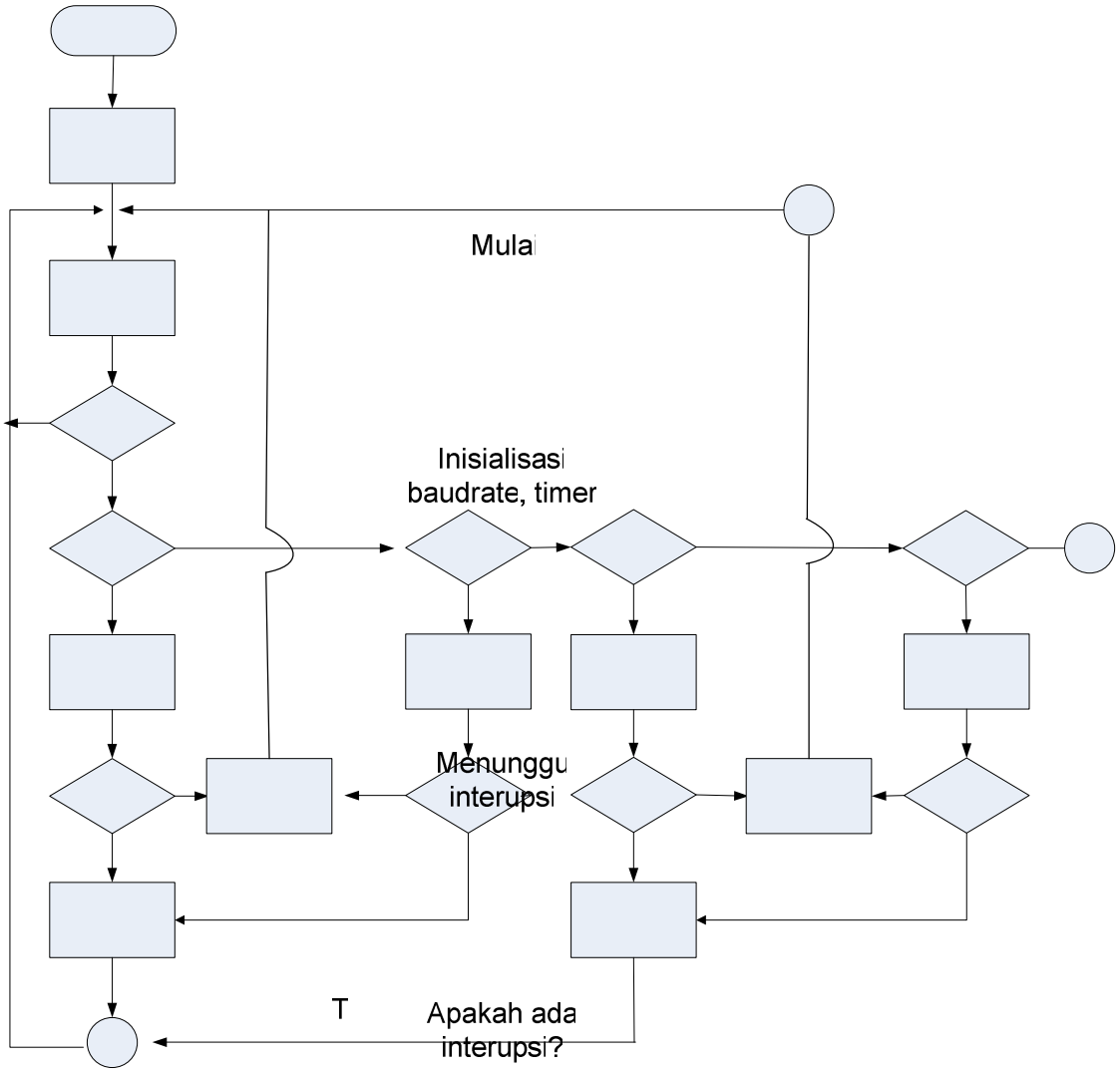
Gambar 3.10. *Bentuk Gedung*

3.3 Perancangan Perangkat Lunak (Software)

3.3.1 Perancangan Pada Mikrokontroler AT89S51

Pemrograman mikrokontroler AT89S51 menggunakan bahasa Assembler, dimana set intruksinya sesuai dengan set intruksi MCS-51. Instruksi-instruksi ini ditulis pada program aplikasi *Windows* yaitu *notepad* dan disimpan dalam file ber-*extension* *.asm. File assembler ini kemudian di*compile* ke file obyek ber-*extension* *.Hex. File obyek ini kemudian dimasukkan ke dalam *flash* mikrokontroler melalui alat *Flash PEROM Microcontroller*.

Sebelum mencoba untuk merancang program, terlebih dahulu melakukan proses pembuatan diagram alirnya. Gambar 3.11 merupakan diagram aliran untuk mikrokontroler.

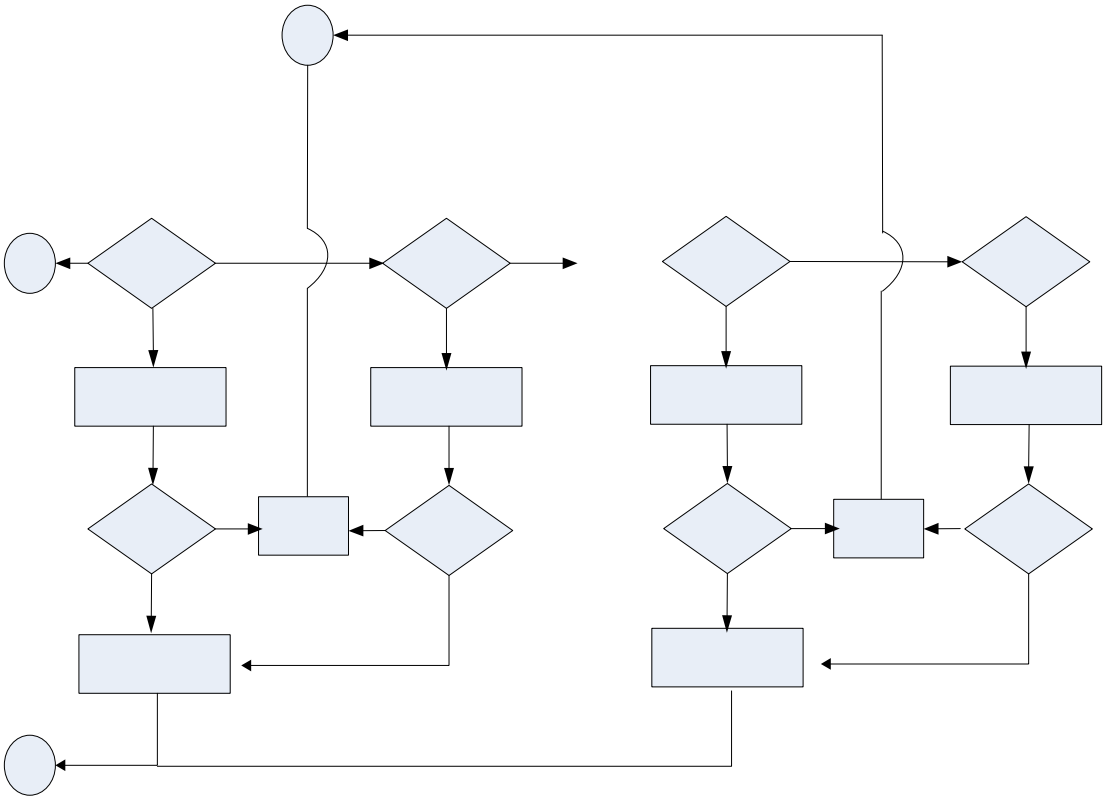


Gambar 3.11. *Flowchart program mikrokontroler*

Flowchart pada gambar 3.11 terdiri atas beberapa proses yang akan dilakukan, yaitu proses inisialisasi, proses menunggu sebuah perintah (interupsi), pemeriksaan apabila ada perintah (interupsi), proses baca sensor. Penjelasan flowchart pada gambar 3.11 adalah sebagai berikut.

Pertama dilakukan inisialisasi variabel (*timer, baudrate*), setelah itu proses menunggu sebuah perintah adanya interupsi. Jika ada interupsi maka isi akumulator akan di bandingkan. Apakah isi akumulator ≠ ‘a’ jika ya maka dibandingkan lagi. Jika tidak hapus pin P0.0 maka gedung 1 ruang 1 akan nyala, lalu dicek status lampu (nyala/mati), jika nyala maka kirim ‘B’ ke R0 , jika mati maka kirim ‘A’ ke R0 lalu kembali menunggu interupsi. Kemudian dibandingkan lagi apakah isi akumulator ≠ ‘b’ jika ya maka dibandingkan lagi.

Jika tidak set pin P0.0 maka gedung 1 ruang 1 akan mati, lalu dicek status lampu (nyala/mati). Jika nyala maka kirim ‘B’ ke R0 , jika mati maka kirim ‘A’ ke R0 kemudian kembali menunggu interupsi. Setelah itu dibandingkan lagi apakah isi akumulator ≠ ‘c’ jika ya maka dibandingkan lagi. Jika tidak hapus pin P0.1 maka gedung 1 ruang 2 akan nyala, lalu dicek status lampu (nyala/mati). Jika nyala maka kirim ‘D’ ke R0 , jika mati maka kirim ‘C’ ke R0 lalu kembali menunggu interupsi. Dibandingkan lagi apakah isi akumulator ≠ ‘d’ jika ya maka dibandingkan lagi. Jika tidak set pin P0.1 maka gedung 1 ruang 2 akan mati, lalu dicek status lampu (nyala/mati). Jika jika nyala maka kirim ‘D’ ke R0 , jika mati maka kirim ‘C’ ke R0 kemudian kembali menunggu interupsi. Demikian program seterusnya untuk lampu berikutnya dan untuk pengecekan status lampu. Setiap interupsi akan selalu diperiksa dan melakukan perintah selanjutnya jika interupsi tersebut sesuai. Dibawah ini merupakan lanjutan dari program mikrokontroler dari gambar 3.11. Adapun lanjutan program mikrokontroler yang dapat dilihat pada gambar 3.12



Gambar 3.12. *Lanjutan flowchart program mikrokontroler*

Keterangan :

ON : Lampu dalam keadaan nyala

OFF: Lampu dalam keadaan mati

A

Tabel 3.2. *Port yang digunakan pada mikrokontroler*

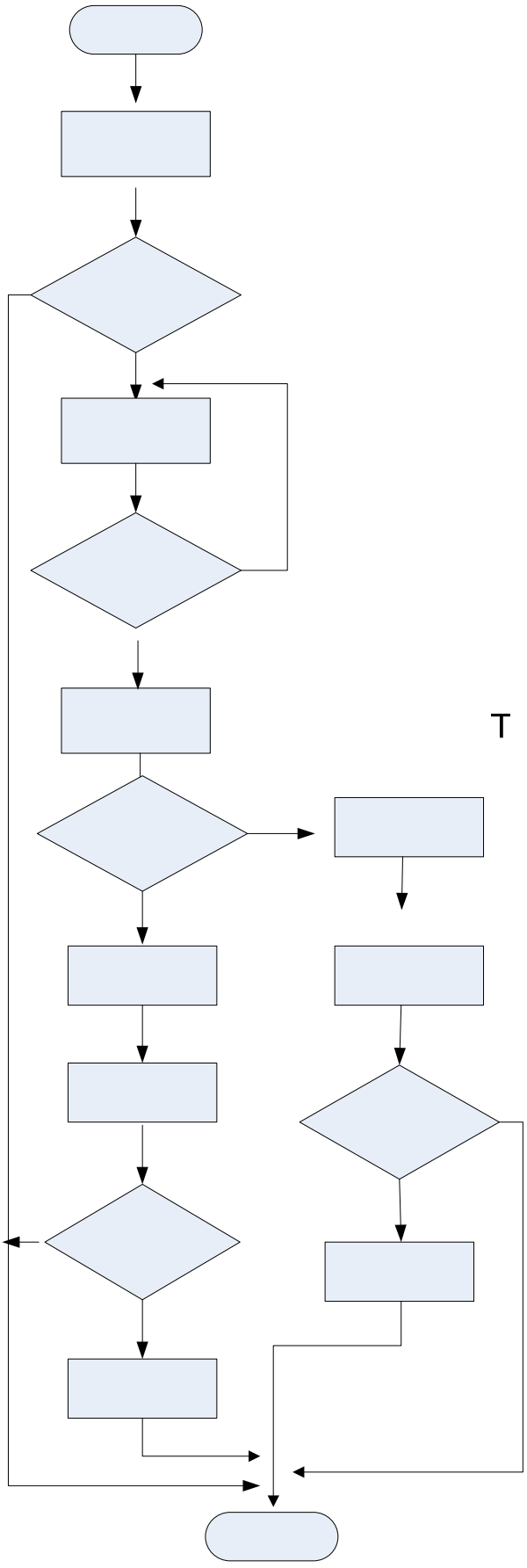
	Gedung dan Ruang	Port pada Mikrokontroler	Pin pada Mikrokontroler
1	Gedung1 ruang 1	Port P0	Pin P0.0
2	Gedung1 ruang 2		Pin P0.1
3	Gedung1 ruang 3		Pin P0.2
4	Gedung1 ruang 4		Pin P0.3
5	Gedung1 ruang 5		Pin P0.4
6	Gedung1 ruang 6		Pin P0.5
7	Gedung2 ruang 1		Pin P0.6
8	Gedung2 ruang 2		Pin P0.7
9	Gedung2 ruang 3	Port P2	Pin P2.0
10	Gedung2 ruang 4		Pin P2.1
11	Gedung2 ruang 5		Pin P2.2
12	Gedung2 ruang 6		Pin P2.3

3.3.2 Perangkat Lunak pada Visual Basic

Pada perancangan program Visual Basic penulis membagi menjadi 2 bagian yaitu perancangan program *client* dan perancangan program pada *server*. Agar program *client* dan program *server* dapat saling komunikasi dan bertukar data maka kedua program tersebut menggunakan *socket programming.* Pada sistem operasi yang berbasis *windows* telah disediakan Winsock atau Windows Socket API *(Application Programming Interface)* sebagai komunikasi antara *client* dan *server* yang terdapat pada Visual Basic.

3.3.2.1 Perancangan Program Client

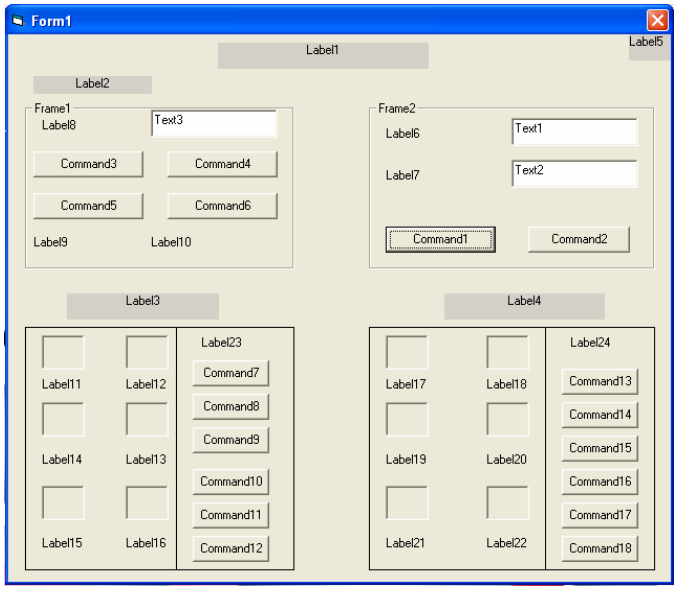
Sebelum merancang tampilan pada program *client* pertama yang dilakukan adalah membuat alur *flowchart* program. Pada program *client* hanya dapat mengirim dan menerima data dari ke PC *server*. Adapun alur *flowchart* program *client* yang dapat dilihat pada gambar 3.13.



Gambar 3.13. *Flowchart program pada client*

Berikut ini adalah penjelasan *flowchart* program *client*. Pertama melakukan set port yang akan digunakan untuk komunikasi dengan *PC server*. Kemudian masukkan IP tujuan *server* dengan benar, jika salah maka status tidak terhubung (koneksi ditutup). Jika IP benar maka status terhubung dengan PC *server*. Setelah terkoneksi ke *server* maka *client* memasukkan *user* dan *password server* untuk ijin pengontrolan aplikasi program *client*). Apabila *server* telah memverifikasi *user* dan *password*, maka aplikasi dapat dibuka atau digunakan oleh PC *client* (*user* dan *password server* benar).

Setelah membuat *flowchart* program *client* selanjutnya merancang tampilan untuk program *client*. Tampilan dirancang sedemikian rupa sehingga *user friendly* dan mudah untuk digunakan. Berikut ini adalah tampilan pada program *client* yang terlihat pada gambar 3.14.



Gambar 3.14. *Tampilan program client*

Adapun penjelasan untuk tampilan program client diatas sebagai berikut :

1. Label 1 berfungsi untuk label judul
2. Label 2 berfungsi untuk label *form client*
3. Label 3 dan 4 berfungsi untuk label gedung.
4. Label 5 berfungsi untuk keluar dari program.
5. Label 6 dan 7 berfungsi sebagai label untuk user dan password.
6. Label 8 berfungsi untuk label inputan IP.
7. Label 9 fan 10 berfungsi sebagai status koneksi
8. Label 11,12,13,14,15,16,17,18,19,20,21, dan 22 berfungsi untuk label ruangan.
9. Frame 1dan 2 berfungsi untuk melihat koneksi dan login *server*.

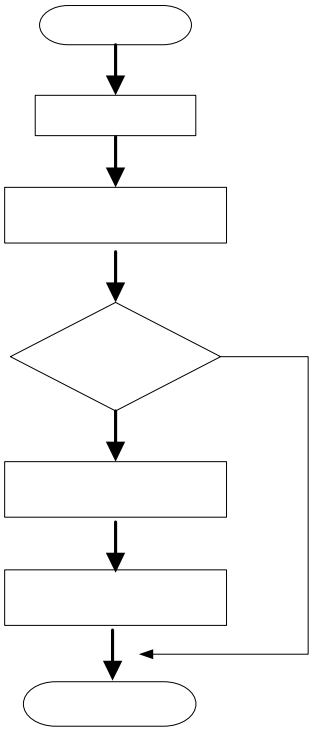
10. Label 23 dan 24 berfungsi untuk label gedung1 dan gedung2.
11. Tombol *Command 1*dan 2 berfungsi untuk proses *user* dan *password.*
12. Tombol *Command 3* dan *Command 4* berfungsi untuk *connect* dan *disconnect.*
13. Tombol *Command 5* dan *Command 6* berfungsi untuk tombol gedung 1 dan gedung 2.
14. Tombol *Command* 7,8,9,10,11,12,13,14,15,16,17, dan 18 untuk berfungsi untuk tombol lampu.
15. Text1, text2 berfungsi untuk inputan *user* dan *password.*
16. Text3 berfungsi untuk inputan IP.

3.3.2.2 Perancangan Program Server

3.3.2.2.1 Pemograman Winsock Control Server

Setelah merancang alur *flowchart* program *client* beserta tampilannya, selanjutnya merancang program *server* beserta tampilanya juga. Pada perancangan program *server* terdapat dua tipe komunikasi yaitu komunikasi Winsock *(Windows Socket)* sebagai penghubung dengan *client* dan komunikasi serial untuk menghubungkan dengan mikrokontroler.

Dibawah ini merupakan *flowchart* program *server* pada Winsock sekaligus penjelasan dari program tersebut.

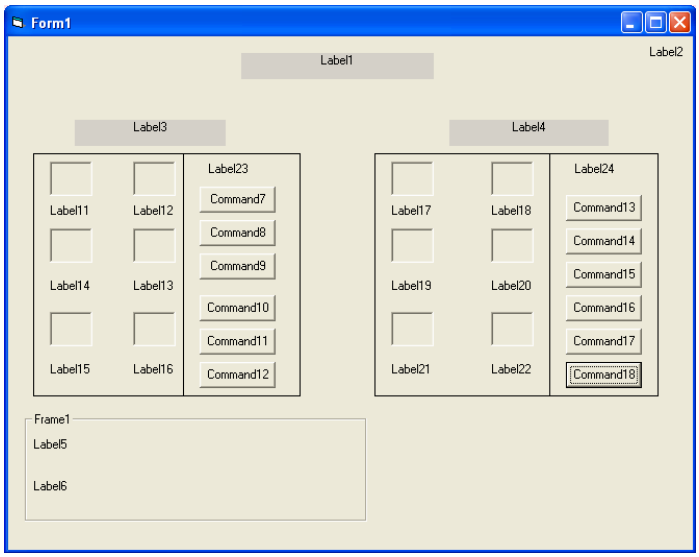


Gambar 3.15. *Flowchart program pada server*

Berikut adalah penjelasan dari *flowchart server* yang terdapat pada gambar 3.15. Pertama dilakukan set port untuk komunikasi dengan *client*. Port yang digunakan adalah 5000. Yang dilakukan *server* pertama kali adalah menunggu koneksi datang dari *client*. Jika ada koneksi dari *client* maka *server* akan menerima koneksi tersebut sehingga dapat melakukan komunikasi atau bertukar data dengan *client.* Jika tidak ada koneksi dari *client* maka koneksi di tutup.

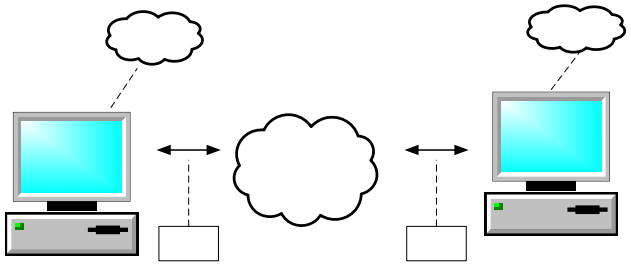
Untuk tampilan program *server* dapat dilihat pada gambar 3.16 dengan keterangan gambar sebagai berikut :

1. Label berfungsi untuk label form server.
2. Label 2 berfungsi untuk keluar dari program.
3. Label 3 dan 4 berfungsi untuk label gedung1 dan gedung 2
4. Label 5 dan 6 berfungsi untuk status koneksi.
5. Label 23 dan 24 berfungsi untuk label ruangan.
6. Tombol *Command* 7,8,9,10,11,12,13,14,15,16,17, dan 18 untuk berfungsi untuk tombol lampu.
7. Label 11,12,13,14,15,16,17,18,19,20,21, dan 22 berfungsi untuk label ruangan.



Gambar 3.16. *Tampilan program server*

Setelah *client* dan *server* terkoneksi melalui Winsock maka proses komunikasi dapat dilakukan. Berikut ini adalah gambaran *client server* terkoneksi dengan masing-masing program *winsock.*



Gambar 3.17.*Client server terkoneksi dengan program Winsock*

Program
Winsock Client

3.3.2.2.2 Pemograman Komunikasi Serial dengan Mikrokontroler

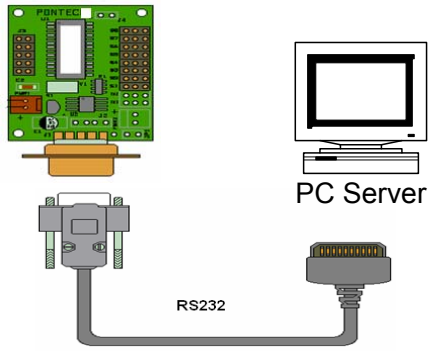
Pada pemograman komunikasi serial menggunakan komponen *MSComm* yang tersedia juga di Visual Basic. *MSComm* berfungsi sebagai *interface* untuk aplikasi program *server* dan mikrokontroler. Adapun program *interface* untuk komunikasi serial sebagai berikut :

PC Client
Visual Basic

MSComm1.CommPort = 1	; port serial yang digunakan adalah COM1
MSComm1.Settings = "9600,N,8,1"	; *Setting Comm* dengan baud rate 9600, tanpa bit paritas, jumlah data 8 bit dan jumlah *stop bit* adalah 1
MSComm1.PortOpen = True	; Membuka port serial *Comm*

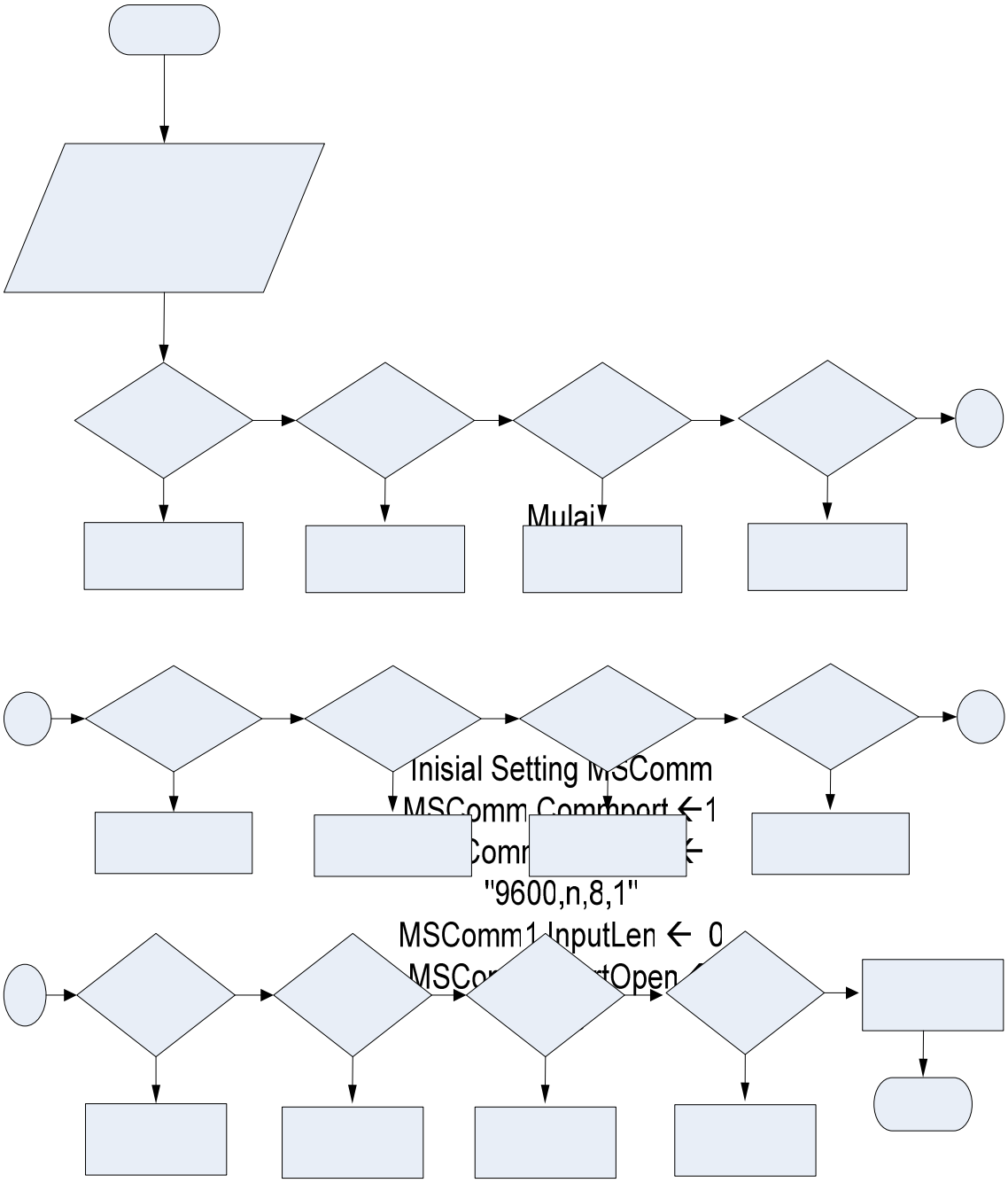
Ethernet
Computer

Berikut ini merupakan koneksi mikrokontroler dengan PC *server*, dimana PC *server* terhubung ke mikrokontroler melalui RS232. Adapun koneksi PC server dengan mikrokontroler yang dapat dilihat pada gambar 3.18



Gambar 3.18. *Koneksi PC server ke mikrokontroler*

Pada gambar 3.19 terlihat alur *flowchart* untuk menyalakan dan mematikan lampu pada Visual Basic. Pertama dilakukan inisialisasi *setting MSComm*, kemudian di lakukan pengecekan, Apakah gedung 1 ruang 1 ditekan jika ya, maka lampu gedung 1 ruang 1 menyala. Kemudian dicek lagi apakah gedung 1 ruang 2 ditekan jika ya maka gedung1 ruang 2 menyala. Demikian program seterusnya untuk ruangan berikutnya. Untuk lebih jelasnya dapat dilihat pada *flowchart* gambar 3.19



.

Gambar 3.19. *Flowchart Visual Basic untuk nyala mati lampu*

3.3.3 Proses Pengiriman Status Lampu

Pertama sensor LDR akan membaca status lampu (nyala atau mati) dan mengirimkan data status ke mikrokontroler kemudian data status tersebut akan diolah oleh mikrokontroler untuk kemudian diteruskan ke PC *server*. Dari PC *server* akan diteruskan juga ke PC *client* dengan cara mengirimkan karakter. Data berupa karakter dari dan ke *client* atau *server* menggunakan sebuah komponen kontrol yaitu Winsock tersedia pada Visual Basic. Setiap kondisi lampu (nyala atau mati) akan mengirimkan karakter yang berbeda untuk setiap lampunya. Adapun ketentuan yang digunakan penulis untuk membedakan status lampu (nyala atau mati) pada program Visual Basic dengan melihat pada *icon* gambar, jika status lampu nyala maka warna pada lampu kuning sebaliknya jika warna lampu putih maka status lampu dalam keadaan mati.