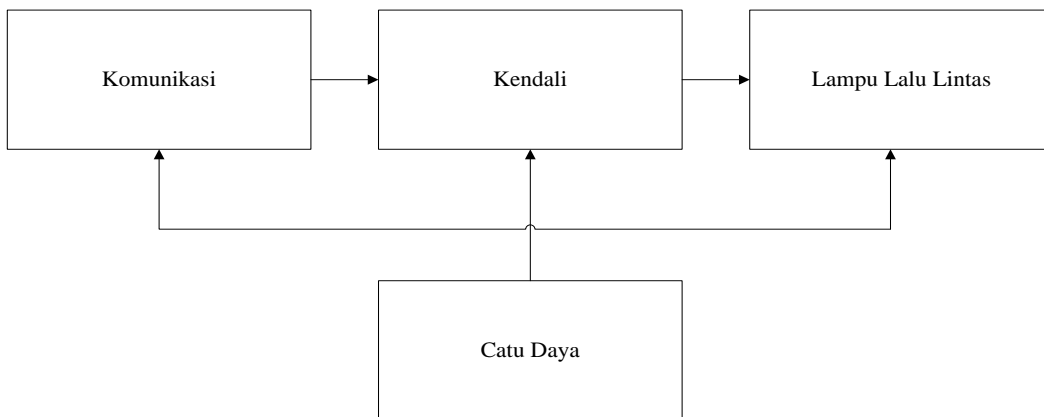


BAB III

PERANCANGAN DAN PEMBUATAN ALAT

3.1 Blok Diagram Sistem Pengendali Lampu Lalu Lintas

Blok sistem pada perencanaan perancangan sistem pengendali lampu lalu lintas secara garis besar terbagi menjadi tiga bagian utama, yaitu bagian catu daya, kendali sistem, lampu lalu lintas dan sub komunikasi.



Gambar 3.1. Blok Diagram Sistem Pengendali Lampu Lalu Lintas

Dari gambar 3.1. Sub sistem yang digunakan dalam perancangan pengendali lampu lalu lintas yaitu :

3.1.1 Sistem Komunikasi

Dalam sub sistem catu daya yang digunakan berupa adaptor yang didalamnya terdapat rangkaian trafo jenis *step down*. Adaptor ini akan mengubah arus AC menjadi arus DC dan menurunkan tegangan PLN menjadi tegangan 9 volt yang nantinya akan digunakan sebagai tegangan masukan dalam rangkaian pengendali lampu lalu lintas.

3.1.2 Kendali

Sistem kendali pada perancangan lampu lalu lintas menggunakan mikropengendali ATmega8535 yang terhubung langsung ke *Real Time Clock* (RTC) dan sebagai outputnya berupa *Light Emitting Diode* (LED).

3.1.3 Sistem Lampu Lalu Lintas

Perancangan lampu lalu lintas yang akan dibuat menggunakan 12 LED sebagai output akhir dengan simulasi perempatan, yang dapat diatur untuk nyala

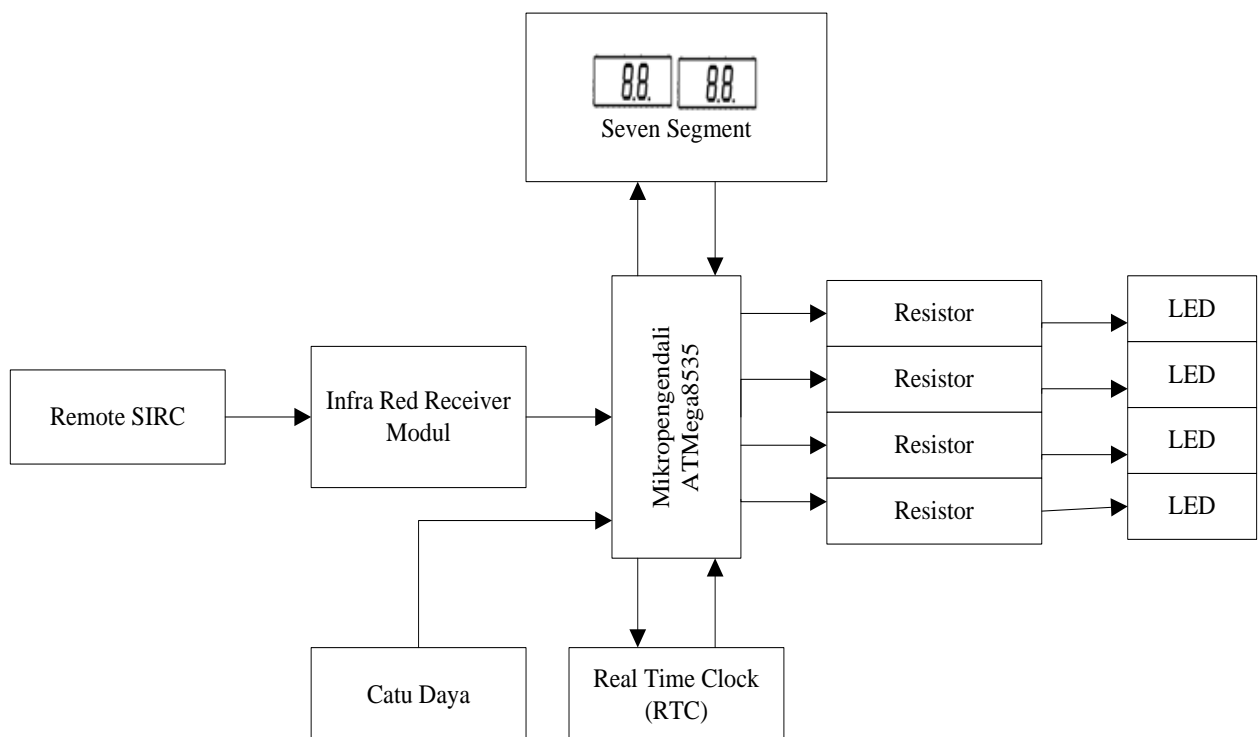
lampu dengan lima kondisi yaitu kondisi jam sibuk pagi jam 05.00 WIB – jam 08.00 WIB, kondisi jam sibuk siang dari jam 08.00 WIB – jam 17.00 WIB, kondisi jam sibuk malam jam 17.00 WIB – jam 19.00 WIB, kondisi jam normal malam 19.00 WIB – jam 23.00 WIB, kondisi jam malam yaitu pada jam 23.00 WIB – 05.00 WIB.

3.1.4 Catu Daya

Dalam sub sistem komunikasi yang digunakan adalah perangkat sensor infra merah yang digunakan sebagai tx di dalam perangkat *remote control* dan sebagai rx yaitu *IR receiver* pada mikropengendali ATmega 8535.

3.2 Perancangan Perangkat Keras

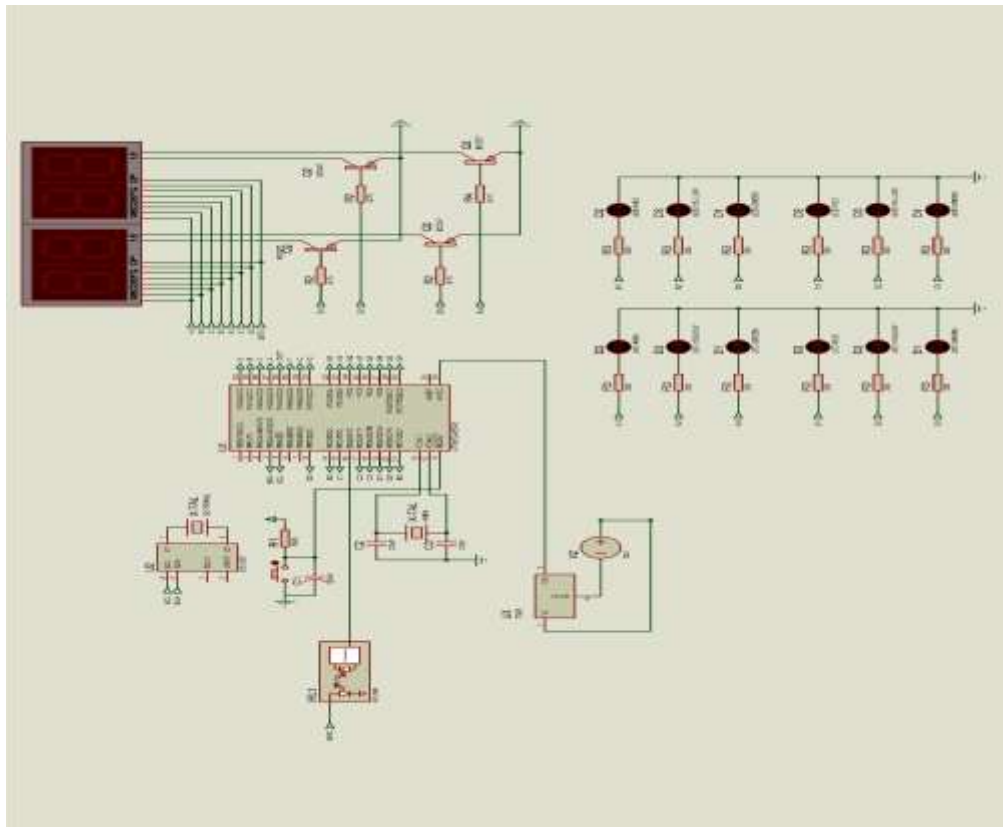
Gambaran umum mengenai proses pengendalian lampu lalu lintas melalui *remote control* dapat dilihat pada gambar 3.2.



Gambar 3.2. Diagram Blok Alat Pengendali

Dari diagram blok diatas maka dapat diketahui bahwa komponen-komponen yang digunakan diantaranya adalah *remote control*, *infra red receiver*, catu daya, mikropengendali ATmega 8535, *Real Time Clock* (RTC) DS 1307, *Seven Segment*, *Resistor* 220 Ω , *Light Emiting Diode* (LED). *Remote control* yang digunakan adalah *Sony Infrared Remote Control* dan berfungsi sebagai alat kendali utama. *Infra red*

Receiver berfungsi sebagai penerima sinyal infra merah dari *remote* yang akan diteruskan ke mikropengendali. Catu daya sebagai tegangan masukan pada perancangan lampu lalu lintas. Pada mikropengendali instruksi-instruksi akan diproses dan akan memeriksa pada RTC untuk referensi pewaktuan pada lampu. Waktu terkini akan ditampilkan pada *seven segment*. Untuk kendali lampu dari mikropengendali dihubungkan pada resistor dan terhubung ke masing-masing LED. Untuk skematik rangkaian alat pengendali keseluruhan ditunjukkan pada gambar 3.3.



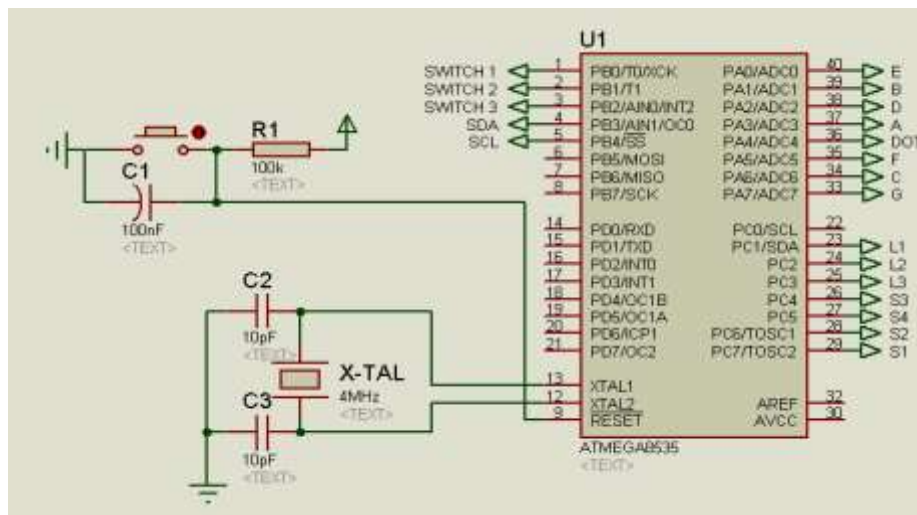
Gambar 3.3. Skematik Rangkaian Alat Pengendali Keseluruhan

Dari gambar 3.3. menjelaskan rangkaian keseluruhan beserta *port-port* yang digunakan pada mikropengendali yang terhubung ke RTC, *Seven Segment* dan *Infrared Receiver*. Untuk elemen kendali rangkaian *driver* transistor tersusun dari 3 buah transistor yang dihubungkan pada 3 buah resistor 4K7 Ohm dan 12 buah resistor 330 Ohm. Sementara resistor 4K7 berfungsi untuk memperkecil arus yang akan masuk ke basis transistor. Hal ini dilakukan untuk mencegah agar transistor tidak rusak akibat besarnya arus yang masuk ke basis. Untuk resistor 330 Ohm berfungsi untuk mengurangi besarnya arus yang masuk ke LED. Jika LED padam berarti perangkat juga dalam kondisi padam, demikian juga sebaliknya jika lampu menyala maka dapat

3.2.2 Rangkaian Mikropengendali ATmega8535

Mikropengendali ATmega8535 merupakan komponen utama sebagai otak pengendalian sistem dalam tugas akhir ini. Data yang dikirimkan melalui *remote* akan diterima oleh *infrared receiver* yang langsung terhubung ke mikropengendali. ATmega8535 akan memproses dan mengolah data yang masuk kemudian memeriksa waktu terkini pada rangkaian *seven segment* dan referensi pewaktuan pada rangkaian *real time clock* DS1307. Setelah membandingkan waktu sesuai data yang ada pada *seven segment* dan RTC maka mikropengendali akan mengeksekusi instruksi pada rangkaian *driver* transistor dan *relay* untuk mengendalikan lampu baik itu menyala ataupun padam sesuai inputan data yang ada. EEPROM pada mikropengendali akan menyimpan data-data yang sudah diatur sehingga data tidak akan hilang jika catu daya utama terputus. Mikropengendali akan mengeksekusi data sesuai dengan instruksi yang diberikan. Mikropengendali ATmega8535 mempunyai pin sebanyak 40 dengan port I/O 32 bit yang dikelompokkan dalam empat *port* yaitu *Port A*, *Port B*, *Port C*, dan *Port D*. *Port-port* tersebut dapat digunakan untuk jalur input maupun output. Mikropengendali tersebut berfungsi sebagai komponen inti pengendali keseluruhan rangkaian yang tentunya sudah diprogram dulu sebelumnya. Dalam tugas akhir ini rangkaian mikropengendali terhubung dengan beberapa rangkaian yang lain diantaranya yaitu rangkaian *seven segment*, rangkaian *real time clock*, rangkaian *driver transistor* serta terhubung langsung ke sensor inframerah. Operasi *port input/output* pada mikropengendali AVR ATmega8535 dapat difungsikan sebagai *input* ataupun *output* dengan keluaran *high* atau *low*. Dengan mengatur fungsi port I/O untuk difungsikan sebagai *input* atau *output* diperlukan pengaturan pada register DDR dan *port*. *Port A* merupakan *port I/O* 8 bit dua arah (begitu juga untuk *port B*, *C* dan *D*). Setiap pinnya dapat menyediakan *internal pull-up resistor* atau dapat diatur per bit (begitu juga untuk *port B*, *C*, dan *D*). *Output buffer port A* dapat memberi arus sebesar 20 mA dan dapat mengendalikan *display LED* secara langsung (begitu juga untuk *port B*, *C*, dan *D*). *Data Direction Register Port* (DDR) harus diatur terlebih dahulu sebelum *port* digunakan. Untuk nama DDR selalu diikuti dengan nama *port* yaitu DDRA, DDRB,

DDRC, dan DDRD. Misalkan saja untuk bit-bit DDRA harus diisi 0 jika ingin memfungsikan pin-pin *port A* sebagai *input* dan diisi 1 jika sebagai *output* (begitu juga untuk *port B, C, dan D*). *Port-port* tersebut dapat digunakan untuk jalur input maupun output. Untuk membuat sistem minimum rangkaian mikropengendali ATmega8535 maka rangkaian tersusun atas mikropengendali tipe ATmega8535, satu kapasitor 100 nF untuk Vcc dan GND, satu resistor 4k7 dan dioda untuk *reset*, satu xtal 4 Mhz sebagai eksternal *oscillator*, dua buah kapasitor pada untuk rangkaian xtal, dan *header 5x2* untuk masing-masing *port* pada mikropengendali. Perancangan sistem minimum ATmega8535 dapat ditunjukkan pada gambar 3.5.

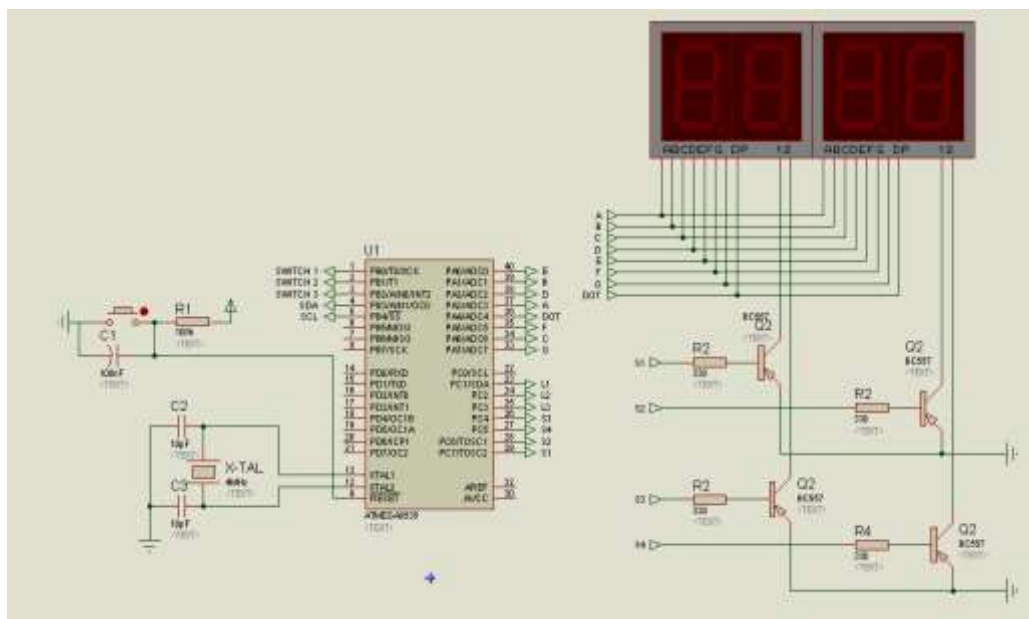


Gambar 3.5 Perancangan Sistem Minimum ATmega8535

3.2.3 Rangkaian *Seven segment*

Rangkaian *Seven segment* digunakan untuk mengatur waktu terkini dalam penyalan kondisi lampu lalu lintas, pengaturan kendali lampu lalu lintas dan pengaturan untuk *keylock*. Waktu yang ditampilkan oleh rangkaian *seven segment* pada Tugas Akhir ini dalam format jam dan menit dalam format 24 jam. Untuk dapat menampilkan sebuah karakter pada LED *seven segment*, maka harus diatur agar hanya *segment* tertentu saja yang menyala sesuai dengan bentuk tampilan. Rangkaian ini beroperasi baik dengan bantuan rangkaian *real time clock*. *Real time clock* akan menyimpan data waktu yang ditampilkan oleh *seven segment*. Rangkaian *seven segment* tersebut langsung terhubung dengan mikropengendali dan mendapat catuan daya lewat mikropengendali tersebut. Pada dasarnya penggunaan *seven segment*

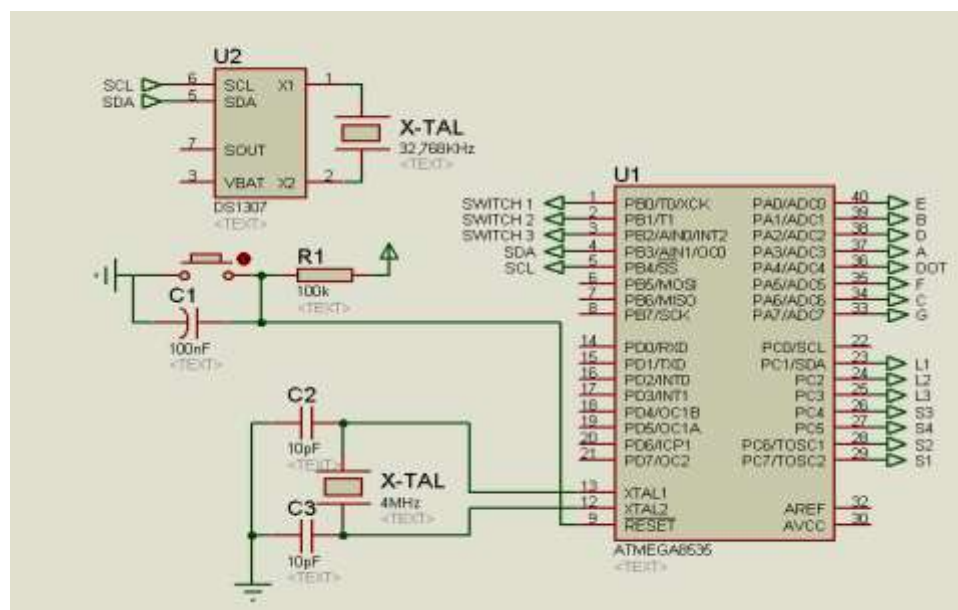
menggunakan konsep *scanning display* agar lebih menghemat dalam penggunaan *port*. Teknik *scanning* ini memanfaatkan kelemahan atau keterbatasan mata manusia yang tidak dapat berakomodasi secara baik ketika dua atau lebih karakter ditampilkan secara bergantian tetapi dengan tundaan waktu (*delay*) yang sangat cepat sehingga seakan-akan karakter-karakter tersebut ditampilkan secara simultan atau bersamaan. Tipe *seven segment* yang digunakan dalam tugas akhir ini adalah *seven segment common anode*. Tampilan *common anode* adalah aktif *low (low enable)* sebab LED mendapatkan tegangan *low* untuk pencahayaan setiap *segment*. Dengan konfigurasi tersebut maka LED *seven segment* harus diberi tegangan catu bersama sebesar 5 volt, sehingga segmen akan menyala jika diberi nilai nol (*low*) dan padam jika diberi nilai satu (*high*). Untuk dapat menampilkan sebuah karakter pada LED *seven segment*, maka harus diatur agar hanya *segment* tertentu saja yang menyala sesuai dengan bentuk tampilan. Untuk rangkaian *seven segment* tersusun atas beberapa komponen yaitu dua buah *seven segment double common anode*, empat buah transistor BC557, dan empat buah resistor senilai 4K7. Rangkaian *seven segment* akan terhubung ke *port* mikropengendali dan bekerja dengan memanfaatkan rangkaian *real time clock* sehingga waktu dapat terus berjalan secara terus menerus. Untuk perancangan rangkaian *seven segment* ditunjukkan pada gambar 3.6.



Gambar 3.6 Perancangan Rangkaian *Seven segment*

3.2.4 Rangkaian *Real Time Clock*

Rangkaian *Real Time Clock* DS1307 digunakan sebagai referensi pewaktuan pada kondisi lampu lalu lintas. Rangkaian tersebut dapat menyimpan data waktu sehingga dapat digunakan untuk menjalankan waktu secara terus-menerus yang dapat ditampilkan pada *seven segment*. Mikropengendali akan membandingkan data yang ada pada RTC dengan waktu terkini sehingga jika cocok maka RTC akan memberikan instruksi pada mikropengendali untuk mengeksekusinya. Pada rangkaian *real time clock* terdapat sebuah baterai *backup* yang digunakan sebagai baterai cadangan sehingga memungkinkan waktu akan terus berjalan meskipun pada saat tidak terhubung dengan catu daya utama. Rangkaian real time clock berfungsi untuk menyimpan dan memproses data yang berhubungan dengan waktu. RTC mampu menyimpan data dalam menit, jam, hari, bulan bahkan tahun.



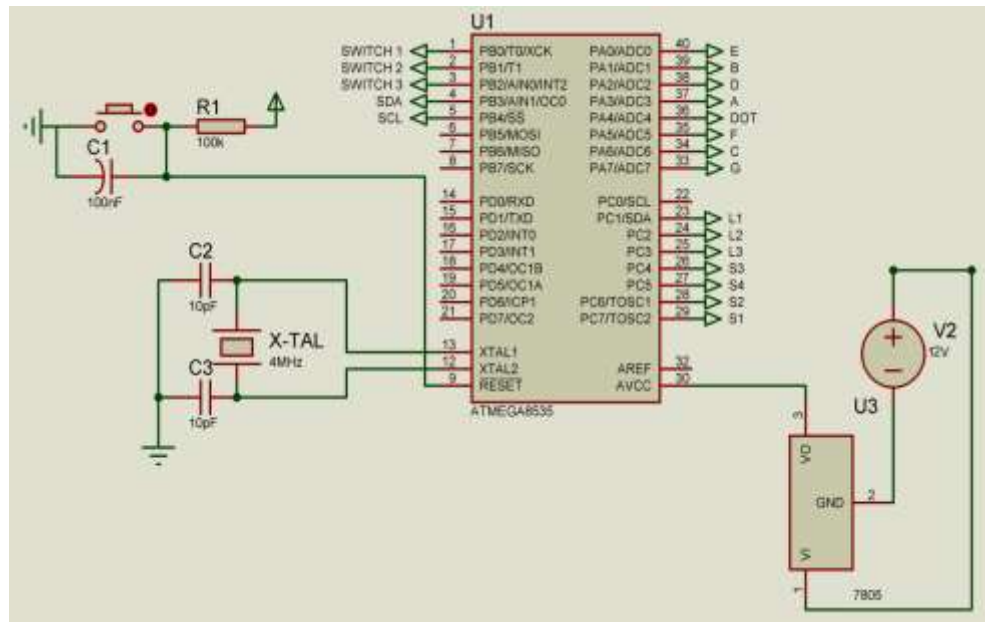
Gambar 3.7 Perancangan Rangkaian *Real Time Clock*

Selain itu pada rangkaian juga mempunyai baterai *backup* atau baterai cadangan yang digunakan sebagai penyimpan daya apabila dalam posisi tidak terhubung ke catu daya utama. Baterai tersebut dinamakan baterai BIOS. Dengan demikian saat rangkaian RTC tidak mendapatkan suplai daya dari catu daya utama maka rangkaian dapat terus bekerja dengan beralih pada mode baterai agar data waktu yang sudah disimpan tidak hilang. RTC terhubung langsung ke mikropengendali untuk bisa disinkronisasikan dengan komponen

yang lainnya khususnya pada rangkaian *seven segment* sehingga waktu bisa berjalan sebagaimana mestinya. Rangkaian *real time clock* terdiri dari sebuah IC DS1307, sebuah kapasitor 100 nF, sebuah baterai BIOS yang digunakan sebagai baterai cadangan apabila rangkaian tidak terhubung dengan catu daya utama, xtal 32,768 Khz, sebuah dioda dan *port* untuk SCL /SDL. Untuk perancangan rangkaian *real time clock* ditunjukkan pada gambar 3.7.

3.2.5 Catu Daya

Catu daya (*power supply*) yang digunakan dalam rangkaian adalah satu buah *adaptor* dengan tegangan 12 VDC yang digunakan sebagai sumber tegangan pada keseluruhan rangkaian. Catu daya merupakan bagian terpenting pada sebuah rangkaian elektronika karena catu daya merupakan sumber tenaga utama yang akan mensuplai daya ke seluruh rangkaian. catu daya dalam rangkaian terdiri dari sebuah *adaptor*. *Adaptor* merupakan sebuah rangkaian elektronika yang dapat mengubah tegangan AC menjadi DC. Rangkaian ini adalah alternatif pengganti dari sumber tegangan DC misalnya batu baterai dan *accumulator*. Keuntungan menggunakan adaptor dibandingkan dengan batu baterai maupun *accumulator* adalah berhubungan dengan ketersediaan tegangan karena *adaptor* dapat mengambil sumber tegangan AC yang ada dirumah. Selain itu *adaptor* mempunyai jangka waktu yang tidak terbatas asalkan tersedianya tegangan AC. Suplai tegangan dari catu daya tersebut mencakup ke seluruh rangkaian yang meliputi rangkaian mikropengendali, rangkaian *seven segment*, rangkaian *real time clock*, rangkaian *driver transistor* serta sensor infra merah. Karena menggunakan sebuah catu daya tunggal dengan tegangan operasi sebesar 12 volt maka diperlukan sebuah komponen yaitu LM8905 yang berfungsi sebagai penstabil tegangan menjadi 5 volt sehingga keseluruhan rangkaian mendapatkan tegangan sesuai dengan porsinya. Perancangan catu daya ditunjukkan pada gambar 3.8.

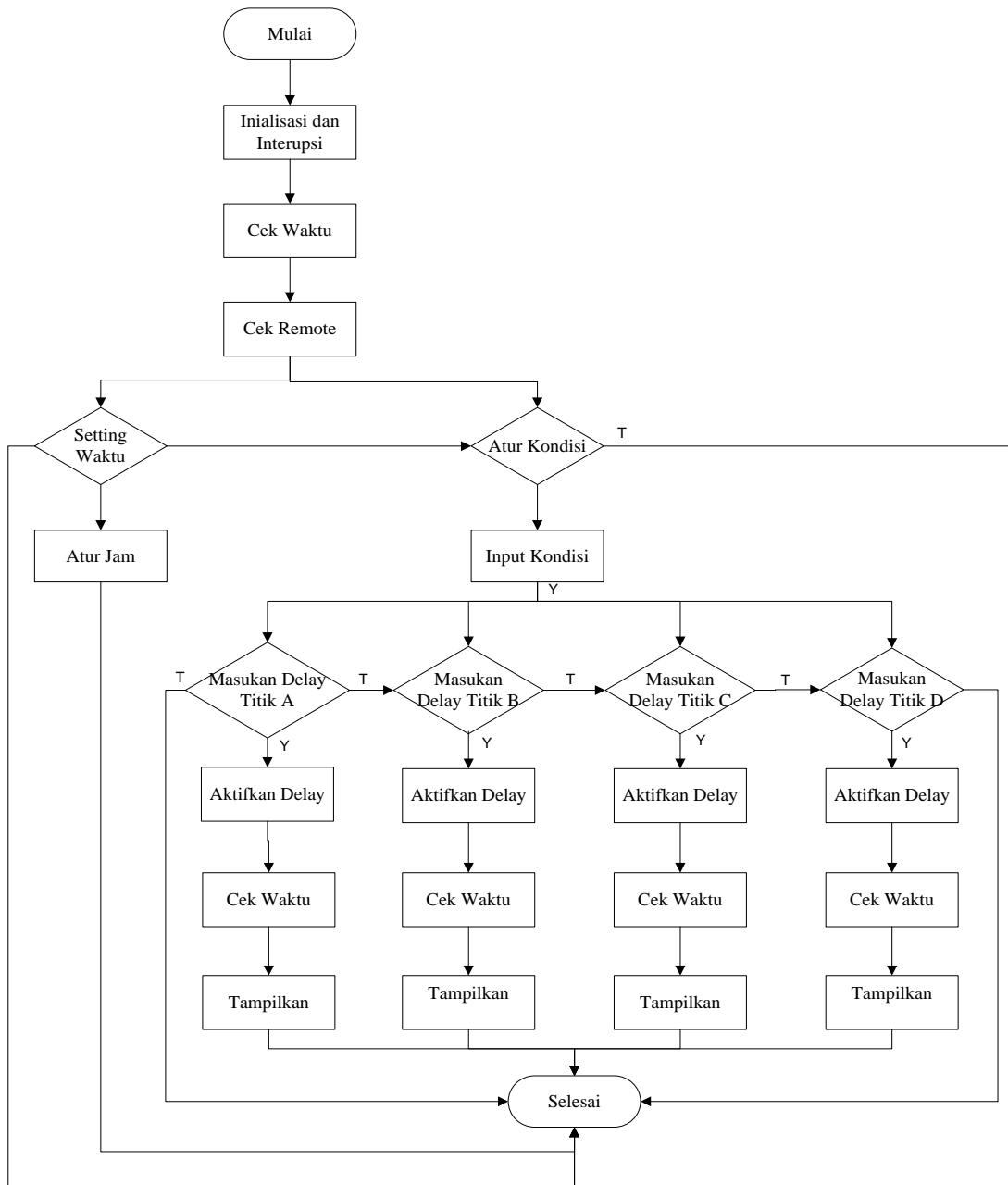


Gambar 3.8 Perancangan Catu Daya

3.3 Perancangan Perangkat Lunak

3.3.1 Flow Chart Sistem

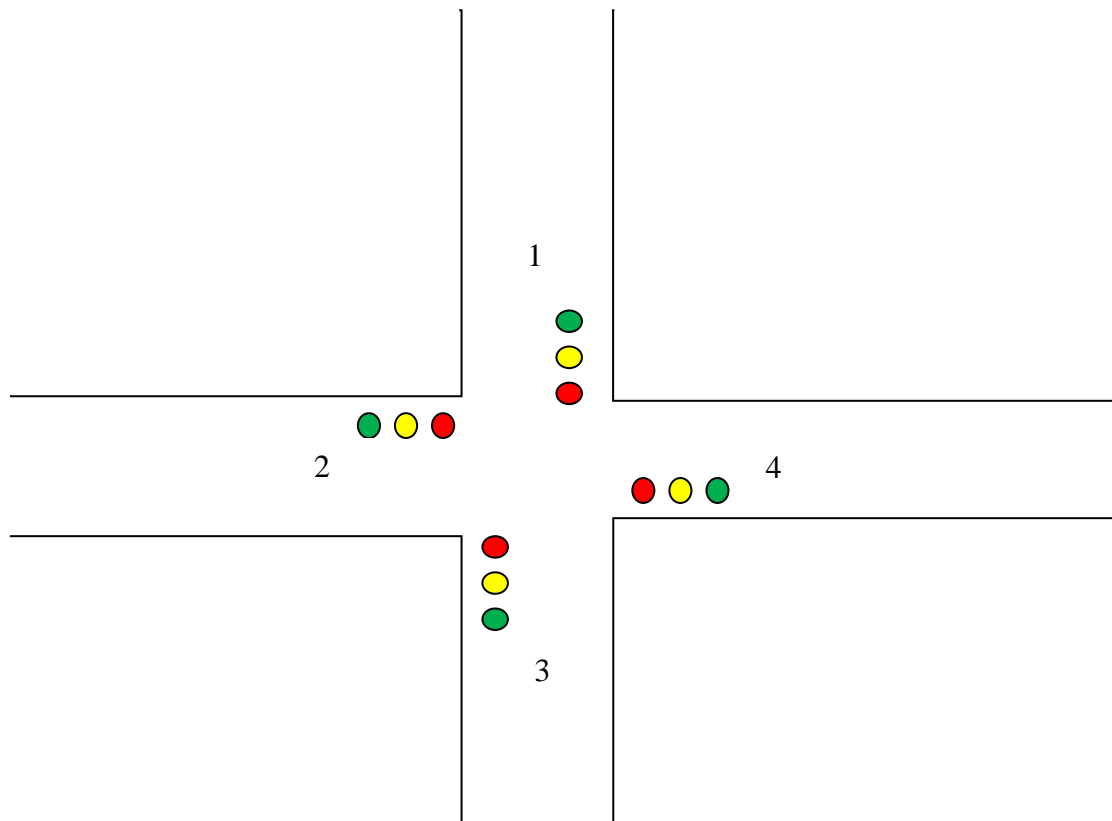
Untuk mengoperasikan alat yang telah dibuat dibutuhkan program dengan bahasa *assembler* yang telah dikompiler kemudian diisikan pada mikropengendali dari rangkaian pengontrol. Adapun tiap tahap penyusunan perangkat lunak diantaranya adalah menyusun diagram alir (*flow chart*) program rangkaian kendali dan membuat perangkat lunak berdasarkan diagram alir yang telah disusun dengan menggunakan bahasa *assembler*. AVR Studio digunakan untuk menuliskan *listing* program atau *coding*. Program yang sudah jadi kemudian disimpan dengan tipe hex yang kemudian akan diisikan ke mikropengendali ATMEGA8535. Selain sebagai *compiler*, AVR Studio digunakan untuk memeriksa koneksi *port-port* pada mikropengendali, untuk mengatur *fusesbit* dan *lockbit* serta digunakan untuk memeriksa *flash* dan EEPROM. Jika mikropengendali sudah terkoneksi dengan laptop melalui USB maka dapat dilakukan *write*, *read* dan *leaving* program. Tentu saja sebelumnya dengan memilih jenis dan tipe mikropengendali yang digunakan.



Gambar 3.9. Data Pembacaan Inputan

Dari gambar 3.9 mengenai diagram alir rangkaian utama diatas dapat dijelaskan bahwa jika ada inialisasi dan interupsi dari *remote control* maka *infrared receiver* akan menerima data tersebut kemudian akan diteruskan ke mikropengendali untuk diproses. Pada mikropengendali akan memeriksa waktu pada *Real Time Clock*. Waktu tersebut akan ditampilkan pada *display seven segment* secara terus menerus sesuai dengan kondisi waktu yang sebenarnya. Dari mikropengendali maka dapat diatur untuk mengendalikan lampu-lampu baik untuk kondisi menyala ataupun padam. Jika data instruksi pada *header*

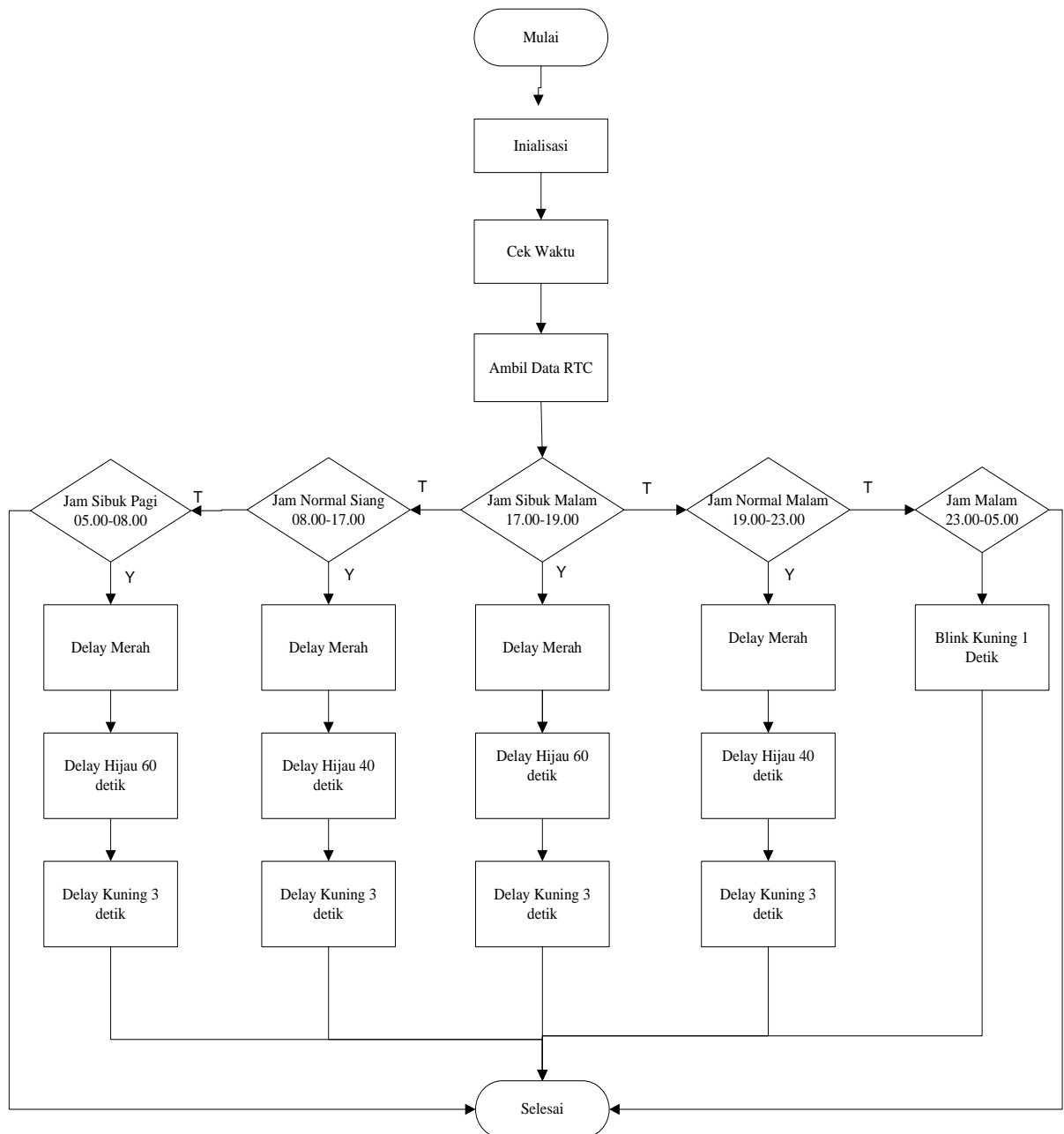
sesuai maka pada *receiver* akan menerima data tersebut kemudian akan diproses untuk mengendalikan lampu merah, kuning dan hijau sesuai kondisi yang sudah diatur sebelumnya.



Gambar 3.10 Miniatur Lampu Lalu Lintas Simpang Empat

Diagram alir pada gambar 3.12 mengenai sub sistem lampu lalu lintas menjelaskan bahwa setelah rangkaian menyala maka akan ada proses inialisasi untuk mengecek rangkaian. Kemudian mengecek waktu dari seven segment untuk pengkondisian waktu, dari mikro mengambil data dari RTC. Untuk selanjutnya diproses dan disetting melalui RTC. Untuk pengaturan lampu dengan lima kondisi yaitu kondisi jam sibuk pagi jam 05.00 WIB – jam 08.00 WIB, kondisi jam sibuk siang dari jam 08.00 WIB – jam 17.00 WIB, kondisi jam sibuk malam jam 17.00 WIB – jam 19.00 WIB, kondisi jam normal malam 19.00 WIB – jam 23.00 WIB, kondisi jam malam yaitu pada jam 23.00 WIB – 05.00 WIB. Asumsi untuk pengkondisian pertama delay nyala lampu yaitu pada jam sibuk pagi delay pada lampu hijau 60 detik, delay pada lampu kuning yaitu 3 detik untuk semua titik. Pada kondisi ke dua delay jam sibuk siang yaitu delay

lampu hijau 40 detik, dan delay lampu kuning 3 detik untuk semua titik. Pada kondisi ke tiga untuk delay jam sibuk malam delay lampu hijau 60 detik, dan delay lampu kuning 3 detik untuk semua titik. Pada kondisi ke empat untuk delay jam normal malam delay pada lampu hijau 40 detik, delay pada lampu kuning 3 detik untuk semua titik. Pada kondisi ke lima yaitu pengkondisian nyala lampu kuning *standby* dengan delay 1 detik.



Gambar 3.11 Sub Sistem Lampu Lalu Lintas

3.3.2 Aturan Kondisi Sesuai Pengaturan Dalam Program

Jasipag : cpi i2cdata, \$8

brge normal

rcall tunda1

rjmp ending

Normal : cpi i2cdata, \$17

brge jasimal

Normal2 : rcall tunda2

rjmp ending

Jasimal : cpi i2cdata, \$20

brge normal2

rcall tunda1

rjmp ending

Jamal : rcall blinkuning

rcall Night

rcall blinkmati

rcall Night

rjmp praending

Night : rcall rtc_read_detik

andi i2cdata,0b01111111

mov detik,i2cdata

ldi dele,\$1

```
add      detik,dele

cpi      detik,$5a

brge    jamal1

mov      dele,metik

andi    dele,0x0f

cpi      dele,$a

brge    jamal2

jamal4:rcall  rtc_read_detik

cp       detik, i2cdata

breq    jamal3

rjmp    jamal4

jamal3 : ret

jamal1 : subi detik,$5a

        rjmp  jamal4

jamal2   : subi  detik, $a

        ldi  dele,$10

        add  detik, dele

        rjmp jamal4

bajamal : rjmp    jamal

bajasipag: rjmp jasipag

praending: rcall  tes

          rcall  rtc_read_jam
```



```
        cpi          i2cdata,$23
        brge    bajamal
        cpi          i2cdata,$5
        brlo    bajamal
        cpi          i2cdata,$5
        brge    bajasipag
kunkun : rcall     tes
        rcall    rtc_read_detik
        andi    i2cdata,0b01111111
        mov     detik,i2cdata
        ldi     dele,$2
        add     detik,dele
        cpi     detik,$5a
        brge    kun1
        mov     dele,detik
        andi    dele,0x0f
        cpi     dele,$a
        brge    kun2
kun4: rcall     tes
        rcall    rtc_read_detik
        cp      detik, i2cdata
        breq    kun3
```

```
        rjmp    kun4

kun3 : ret

kun1 : subi   detik,$5a

        rjmp    kun4

kun2 : subi   detik, $a

        ldi    dele,$10

        add    detik, dele

        rjmp    kun4

ending: rcall tes

        rcall   rtc_read_detik

        andi   i2cdata,0b01111111

        mov    detik,i2cdata

        add    detik,dele

        cpi    detik,$5a

        brge   ending1

        mov    dele,detik

        andi   dele,0x0f

        cpi    dele,$a

        brge   ending2

ending4:rcall tes

        rcall   rtc_read_detik

        cp     detik, i2cdata
```

```
    breq  ending3

    rjmp  ending4

ending3 : ret

ending1 : subi    detik,$5a

    rjmp  ending4

ending2 : subi    detik, $a

    ldi   dele,$10

    add  detik, dele

    rjmp  ending4

ending5 : subi    detik,$60

    rjmp  ending4
```