

BAB IV

PERANCANGAN DAN PEMBUATAN SISTEM

Bab ini menjelaskan mengenai spesifikasi alat, diagram blok sistem, prinsip kerja alat, perancangan perangkat keras (*hardware*), sistem, maupun perangkat lunak (*software*). Perancangan perangkat keras meliputi: perancangan tombol remote kontrol, rangkaian penerima infra merah, mikrokontroler AT89S8252, *latch*, DAC, detektor nol, *driver* triac, triac, *display seven segmen*, lampu indikator. Perancangan sistem berisi rincian pengaturan sistem untuk akses masuk ruangan. Pada perancangan perangkat lunak (*software*) terlebih dahulu dilakukan pembuatan diagram alir program untuk memudahkan logika pembuatan program.

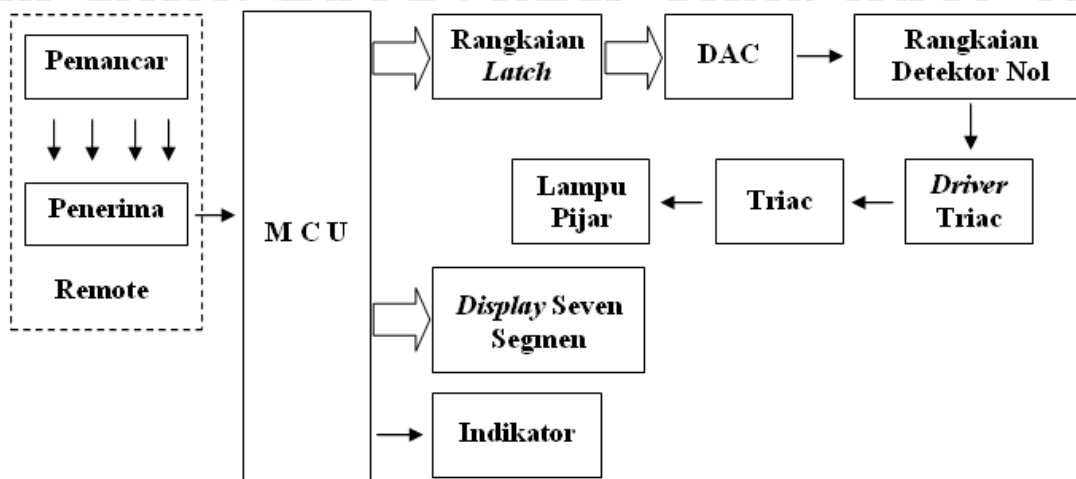
4.1. Penentuan Spesifikasi Alat

Spesifikasi rangkaian secara keseluruhan ditetapkan terlebih dahulu sebagai acuan dalam perancangan selanjutnya. Spesifikasi sistem yang direncanakan adalah sebagai berikut:

1. Menggunakan remote sebagai pengatur jarak jauh.
2. Mikrokontroler AT89S51 sebagai pusat kendali sistem.
3. Menggunakan DAC0800 sebagai perubah sinyal digital ke analog
4. Menggunakan NE555 sebagai pemotong tegangan AC.
5. Menggunakan MOC3021 sebagai *driver* triac dan Q4008 sebagai triac.
6. Tampilan level perubahan daya lampu menggunakan seven segmen.
7. Hasil perubahan ditampilkan dalam dua angka.
8. Indikator led sebagai penanda lampu yang sedang diatur.

4.2. Diagram Blok Alat

Diagram blok rangkaian alat pengatur intensitas cahaya lampu pijar menggunakan remote kontrol direncanakan ditunjukkan pada Gambar 4.1.



Gambar 4.1 Diagram Blok Sistem

Keterangan :

1. Remote kontrol (pemancar)
Digunakan untuk memberikan data sebagai perintah pengaturan lampu.
2. Remote kontrol (penerima)
Digunakan untuk menerjemahkan data pemancar agar dapat dikenali dan diolah oleh mikrokontroler.
2. Mikrokontroler AT89S8252
Digunakan sebagai unit pengontrol dan pengolah data.
3. Rangkaian *Latch*
Digunakan untuk menahan data dari mikrokontroler jika sedang mengolah data pada Impu yang lain.
4. DAC
Digunakan untuk mengubah sinyal digital menjadi analog.
5. Rangkaian Detektor nol
Digunakan untuk mendeteksi pada saat tegangan AC nol sesuai dengan frekuensi jala-jala listrik dan mengeluarkan gelombang pulsa sebagai pengontrol *diver* triac dengan frekuensi 100Hz.
6. *Driver Triac*
Digunakan untuk mengaktifkan dan mematikan triac.
7. Triac
Digunakan untuk mengatur besar tegangan pada beban.
8. *Display*

Digunakan untuk menampilkan level perubahan daya.

9. Indikator

Digunakan untuk menunjukkan lampu yang sedang diatur.

4.3 Prinsip Kerja Sistem

Intensitas cahaya lampu diubah dengan menggunakan remote kontrol. Dengan menekan tombol-tombol yang ada pada remote. Untuk memilih lampu gunakan tombol “1”, “2”, “3”, “4”, merubah intensitas cahaya gunakan tombol “PR+” dan “PR-“, nyala lampu maksimal “Vol+” dan nyala lampu minimal “Vol-“.

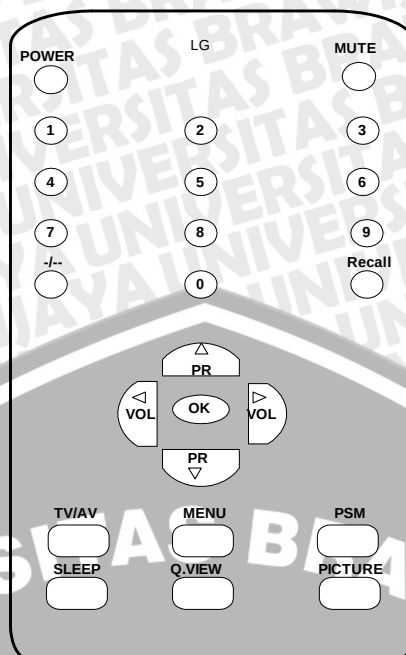
Perubahan cahaya perstep dapat dilihat pada display seven segmen 2 digit. Setiap perubahan dua persen menimbulkan perubahan pula pada nyala lampu. Lampu yang sedang diatur dapat dilihat pada display seven segmen 1 digit dan lampu indikator yang terletak dibawah lampu.

4.4 Perancangan Perangkat Keras

Perancangan perangkat keras (*hardware*) yang dimaksud adalah perancangan tombol remote kontrol, rangkaian penerima infra merah, rangkaian minimum sistem mikrokontroler AT89S8252, rangkaian *lacth*, rangkaian DAC0800, rangkaian detektor nol, rangkaian *driver* triac, triac, rangkaian *display* seven segment, rangkaian indikator.

4.4.1 Perancangan Tombol Remote Kontrol

Perancangan tombol remote kontrol ini dimaksudkan untuk mengetahui fungsi dan tombol mana saja yang digunakan dalam pengoperasian alat. Remote yang digunakan merupakan produk LG dengan tipe LG 6710V00019C. Tombol-tombol yang digunakan adalah 1, 2, 3, 4, PR+, PR-, Vol+, Vol-. Selain tombol tersebut tidak akan direspon oleh hardware.



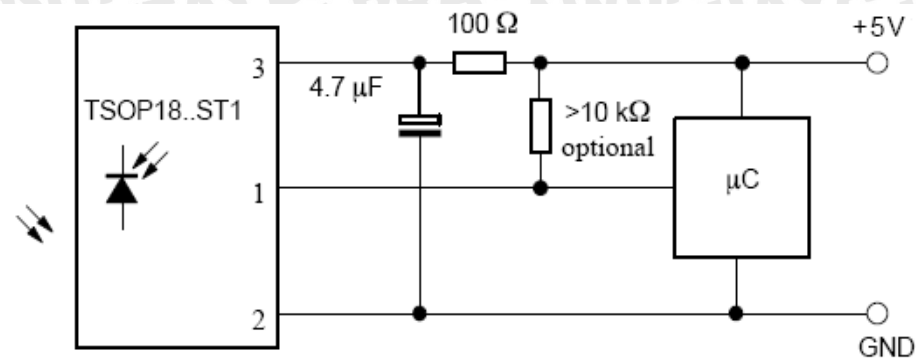
Gambar 4.2 Kontruksi Fisik Remote Kontrol
Sumber : LG 6710V00019C

Fungsi dari tombol-tombol remote kontrol yang digunakan yaitu :

- Tombol 1 : digunakan untuk memilih lampu nomor 1
- Tombol 2 : digunakan untuk memilih lampu nomor 2
- Tombol 3 : digunakan untuk memilih lampu nomor 3
- Tombol 4 : digunakan untuk memilih lampu nomor 4
- Tombol PR+ : digunakan untuk meningkatkan intensitas cahaya.
- Tombol PR- : digunakan untuk menurunkan intensitas cahaya.
- Tombol Vol+ : digunakan untuk menyet kondisi nyala lampu maksimal.
- Tombol Vol+ : digunakan untuk menyet kondisi nyala lampu minimal.

4.4.2 Penerima Inframerah

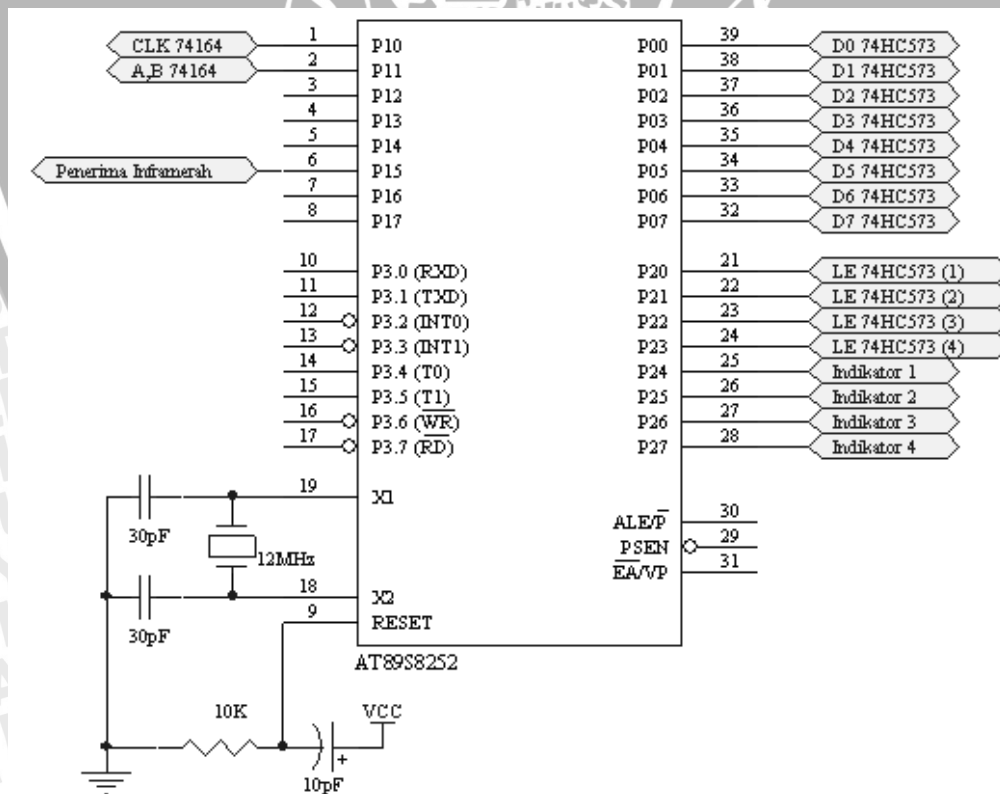
Detektor inframerah berfungsi untuk menangkap/menerima sinyal dari remote kontrol televisi yang digunakan. Pada sistem ini menggunakan infrared detektor module TSOP18 yang mempunyai keluaran dalam tingkat TTL. Dalam perencanaan penerima inframerah digunakan rangkaian yang sesuai *Typical Aplication* dalam datasheet TSOP18 seperti yang ditunjukkan pada Gambar 4.3.



Gambar 4.3 Rangkaian penerima inframerah

4.4.3 Rangkaian Mikrokontroler AT89S8252

Mikrokontroler yang digunakan pada rangkaian ini adalah mikrokontroler tipe AT89S8252 yang merupakan keluarga dari MCS-51. Komponen ini mempunyai 8 Kbyte *downloadable flash programable*, 2 Kbyte EEPROM, 32 pin masukan dan keluaran. Pemilihan mikrokontroler jenis ini karena mudah diperoleh di pasaran dan mempunyai memory internal. Sebagai otak dari pengolahan data dan pengontrolan alat, pin-pin AT89S8252 dihubungkan pada rangkaian pendukung membentuk suatu sistem seperti terlihat pada Gambar 4.4.



Gambar 4.4 Rangkaian Mikrokontroler

Pin-pin mikrokontroler yang digunakan yaitu:

1) Port 0

P0.0-P0.7 dihubungkan dengan D0-D7 74HC573 yang digunakan sebagai masukan data ke 74HC573.

2) Port 1

P1.0 dihubungkan ke CLK 74164 yang digunakan sebagai selektor seven segmen.

P1.1 dihubungkan ke A dan B 74164.

P1.5 dihubungkan ke penerima inframerah.

3) Port 2

P2.0-P2.3 dihubungkan ke pin *output control* (pin 1) 74HC573 yang digunakan sebagai pengontrol rangkaian *latch*.

P2.4-P2.7 dihubungkan ke rangkaian indikator.

4) XTAL1 dan XTAL2

Digunakan sebagai masukan dari rangkaian osiloskop kristal. Rangkaian osiloskop kristal terdiri dari kristal 11.0592 MHz, kapasitor C1 dan C2 yang digunakan masing-masing bernilai 30 pF, akan membangkitkan Pulsa *clock* yang menjadi penggerak bagi seluruh operasi internal mikrokontroler.

5) VCC

VCC dihubungkan dengan tegangan sebesar +5 V sesuai dengan tegangan operasi *chip* tunggal yang diijinkan dalam *datasheet*.

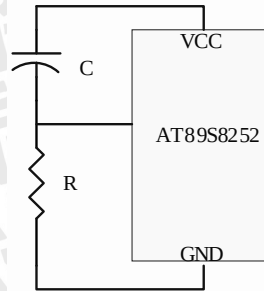
6) GND

GND dihubungkan ke *ground* catu daya.

7) Reset

Rangkaian reset diperlukan untuk mereset mikrokontroler secara otomatis setiap kali catu daya dinyalakan. Hal ini akan mereset program sehingga perintah program dieksekusi dimulai pada alamat awal. Untuk mereset mikrokontroler AT89S8252, pin RST harus diberi logika tinggi selama sekurangnya dua siklus mesin (24 periode osilator) setiap kali catu daya dinyalakan [*datasheet* AT89S8252, 1997:4-108].

Pada saat catu daya diaktifkan, maka kapasitor C_1 sesuai dengan sifat kapasitor akan terhubung singkat. Arus mengalir dari V_{CC} langsung ke pin RST sehingga kaki tersebut berlogika 1. Kapasitor akan terisi sampai tegangan pada kapasitor mencapai V_{CC} , otomatis tegangan pada RST akan turun menjadi logika 0 dan proses *reset* selesai. Rangkaian reset dibentuk dengan rangkaian RC yang ditunjukkan pada Gambar 4.5.



Gambar 4.5 Rangkaian Reset
Sumber: *Atmel*, 1997: 2-63

Karena kristal yang digunakan mempunyai frekuensi sebesar 11.0592 MHz, maka satu periode membutuhkan waktu sebesar :

$$T = \frac{1}{f_{XTAL}} = \frac{1}{11.0592 \text{ MHz}} \text{ s} = 0,090422 \cdot 10^{-6} \text{ s} \text{ Sehingga waktu minimal logika}$$

tinggi yang dibutuhkan untuk mereset mikrokontroler adalah :

$$\begin{aligned} t_{reset(\min)} &= T \times \text{periode yang dibutuhkan} \\ &= 9,0422 \cdot 10^{-8} \times 24 \\ &= 2,170 \mu\text{s} \end{aligned} \quad (4.1)$$

Jadi mikrokontroler membutuhkan waktu minimal 2,170 μs untuk mereset program. Untuk menghindari penurunan waktu minimal di bawah 2,170 μs karena faktor toleransi kesalahan nilai resistor dan kapasitor, maka waktu minimal ditentukan sebesar 20 μs . Nilai R dan C dapat dicari dengan persamaan berikut:

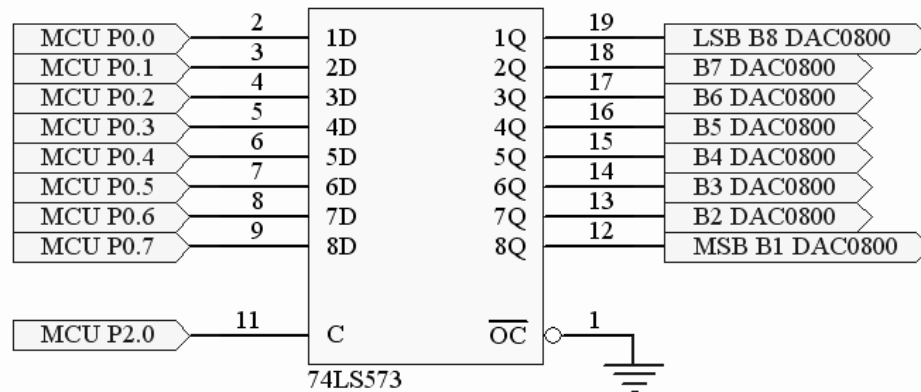
$$\begin{aligned} t &= 0,357 \cdot R \cdot C \\ 20 \cdot 10^{-6} &= 0,357 \cdot 10^3 \cdot C \\ C &= 5,6 \cdot 10^{-9} \text{ F} \end{aligned} \quad (4.2)$$

Jadi dengan nilai komponen $R = 10 \text{ K}\Omega$, nilai kapasitor yang dapat memenuhi syarat untuk mereset mikrokontroler harus diatas 5 nF. Untuk kemudahan perancangan dipilih $C = 10 \text{ nF}$.

4.4.4 Rangkaian Latch

Pada perancangan ini menggunakan IC 74HC573 yang dihubungkan ke mikrokontroller sebagai masukan datanya. Rangkaian ini digunakan untuk menahan data sementara dari mikrokontroler dan mengeluarkan data tersebut setelah mendapatkan perintah dari mikrokontroler. Data *input* dimasukkan ke pin 1D sampai 8D, lalu

keluarannya ialah pada pin 1Q sampai 8Q. Rangkaian penahan atau *latch* ditunjukkan pada Gambar 4.6.



Gambar 4.6 Rangkaian Latch

Pin *output control* (pin 1) diset *active low*, karena agar semua data yang masuk di *input* D akan dapat dikeluarkan ke *output* Q. Pin *latch enable* (pin 11) dihubungkan ke mikrokontroller yang igunkn untuk mengendalikan kapan data yng terdapat di D dikeluarkan atau ditahan dulu. Jika pin *latch enable* diberi logika 0, maka data yang ada pada pin-pin Q akan ditahan. Sedangkan jika pin *latch enable* diberi logika 1, maka data tidak akan ditahan pada pin Q. data yang masuk di pin D akan selalu sama dengan data pada pin-pin Q.

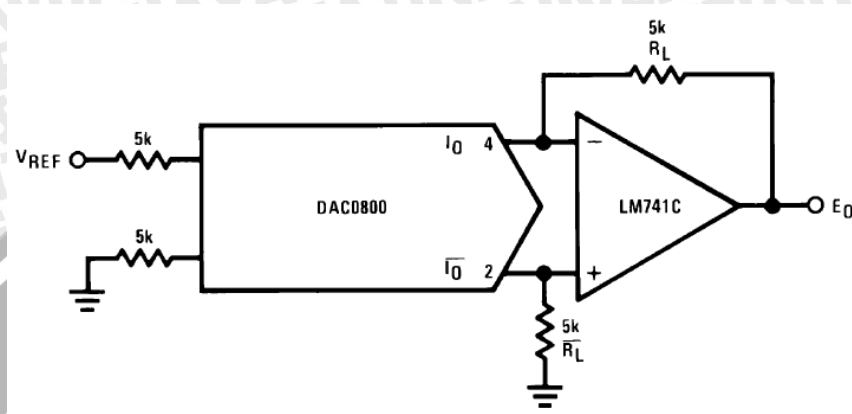
4.4.5 Rangkaian DAC0800

Digital to Analog Converter digunakan untuk mengubah besaran sinyal digital 8 bit yang merupakan keluaran dari mikrokontroler, yang nantinya digunakan sebagai masukan ke rangkaian detektor nol. Digital to Analog Converter yang digunakan adalah DAC 0800 yang mempunyai kemampuan mengkonversi 8 bit data digital menjadi besaran tegangan dalam range tertentu. Keluarannya berupa arus, sehingga diperlukan komponen tambahan untuk merubahnya menjadi tegangan. Besarnya tegangan keluaran tergantung pada tegangan referensi yang dimasukkan ke DAC. Dalam datasheet disebutkan bahwa $I_{REF\ MAX}$ adalah 5mA. Dalam perancangan ini digunakan tegangan referensi (V_{REF}) sebesar 5V dengan resistansi referensi (R_{REF}) sebesar 5k Ω (sesuai dengan *Typical Aplication* dalam datasheet)

$$\begin{aligned}
 I_{REF} &= \frac{V_{ref}}{R_{ref}} \\
 &= \frac{5V}{5k}
 \end{aligned}
 \quad (4.3)$$

$$= 1 \text{ mA}$$

Jadi R_{REF} sebesar $5k\Omega$ masih dapat digunakan, karena I_{REF} masih dibawah nilai maksimal (5mA). Keluaran DAC0800 adalah berupa arus, untuk mengubahnya menjadi tegangan diperlukan rangkaian tambahan yang menggunakan LM741. Seperti yang terdapat pada Gambar 4.7.

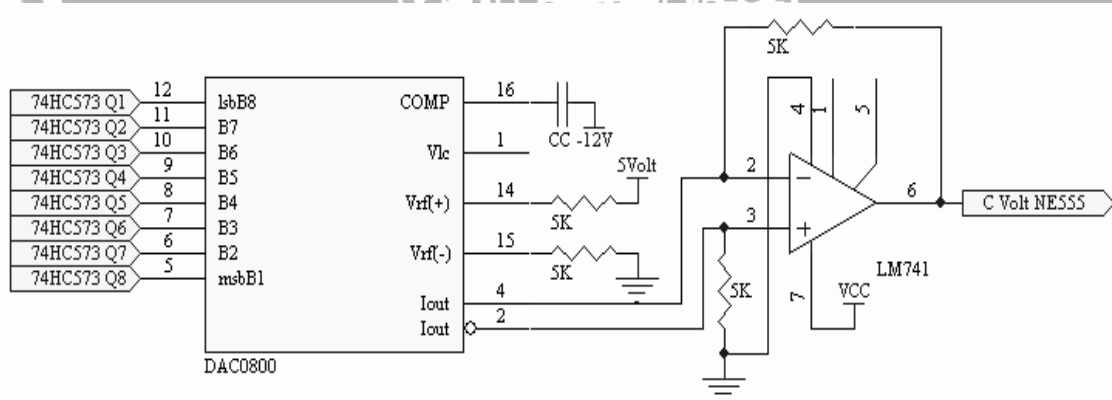


Gambar 4.7 Penggunaan LM741 Dalam Rangkaian DAC0800
Sumber: Fairchild Semiconductor, *DAC0800 Datasheet*

Dengan menggunakan rangkaian diatas nilai E_O dapat diketahui. Seperti yang terdapat dalam Fairchild Semiconductor, *DAC0800* datasheet:

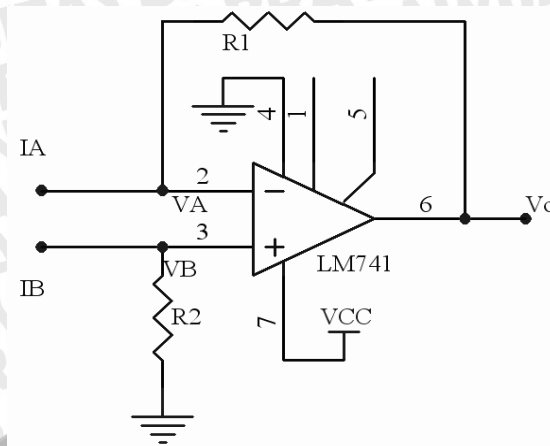
$$E_O = V_{ref} \left(\frac{-255}{256} + \frac{2x}{256} \right), \text{ dimana } x \text{ adalah bit masukan DAC0800.} \quad (4.4)$$

Untuk memperoleh nilai diatas R_L harus sama dengan R_{REF} , maka $R_L = R_{REF} = 5k\Omega$, seperti yang terdapat pada Gambar 4.8.



Gambar 4.8 Rangkaian DAC0800

Pengubahan keluaran DAC yang berupa arus menjadi tegangan menggunakan LM741. Komponen ini difungsikan sebagai penguat diferensial. Jadi, perubahan tegangannya tergantung dari kedua arus masukan penguat. Rangkaian penguat diferensial terdapat pada Gambar 4.9.



Gambar 4.9 Rangkaian Penguat Diferensial

$$V_B = V_A$$

$$V_B = I_B \cdot R_2$$

$$0 = I R_1 + I_A$$

$$0 = \left(\frac{V_0 - V_A}{R_1} \right) + I_A$$

$$(V_0 - V_A) = -I_A \cdot R_1$$

$$(V_0 - (I_B \cdot R_2)) = -I_A \cdot R_1$$

$$V_0 = (I_B \cdot R_2) - (I_A \cdot R_1)$$

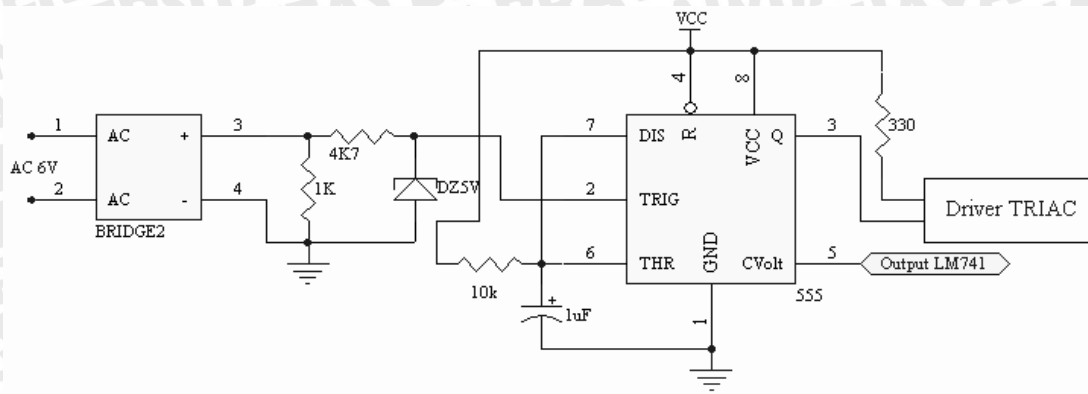
$$R_1 = R_2 = R$$

$$V_0 = (I_B - I_A) \cdot R$$

Dari persamaan diatas dapat diketahui fungsi LM741 serta perubahan arus keluaran DAC menjadi tegangan.

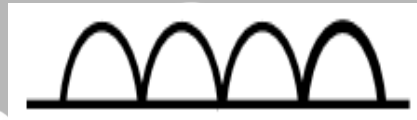
4.4.6 Rangkaian Detektor Nol

Rangkaian detektor nol menggunakan rangkaian multivibrator yang dirangkai dengan komponen lain sehingga membentuk detektor nol yang dapat mengetahui perlintasan titik nol. Rangkaian detektor nol menggunakan IC555 dengan mode *monostable* ditunjukkan pada Gambar 4.10.



Gambar 4.10 Rangkaian Detektor Nol

Detektor ini menggunakan susunan dioda jembatan (bridge diodes) berfungsi untuk menyearahkan tegangan masukan. Keluaran dari dioda jembatan ini terlihat pada Gambar 4.11.



Gambar 4.11 Gelombang Keluaran Dioda Jembatan

Dari gambar diatas dapat dilihat terjadinya titik nol dua kali setiap periodenya karena frekuensi penyearah setengah gelombang sama dengan frekuensi jaringan listrik, maka untuk penyearah gelombang penuh menghasilkan frekuensi dua kali frekuensi tegangan jaringan. Dengan asumsi tegangan jaringan 50 Hz, maka dengan dioda jembatan didapat frekuensi 100 Hz. Sedangkan dioda zener adalah pengatur tegangan supaya tegangan yang masuk ke LM555 sebesar 5 volt. Dalam datasheet disebutkan $\Delta T = 1.1 RC$, maka:

$$f = \frac{1}{T} \quad (4.5)$$

$$f = \frac{1}{1.1RC}$$

$$100 = \frac{1}{1.1(10 \cdot 10^3 \cdot C)}$$

$$C = \frac{1}{11 \cdot 10^5}$$

$$C = 0.9 \cdot 10^{-6}$$

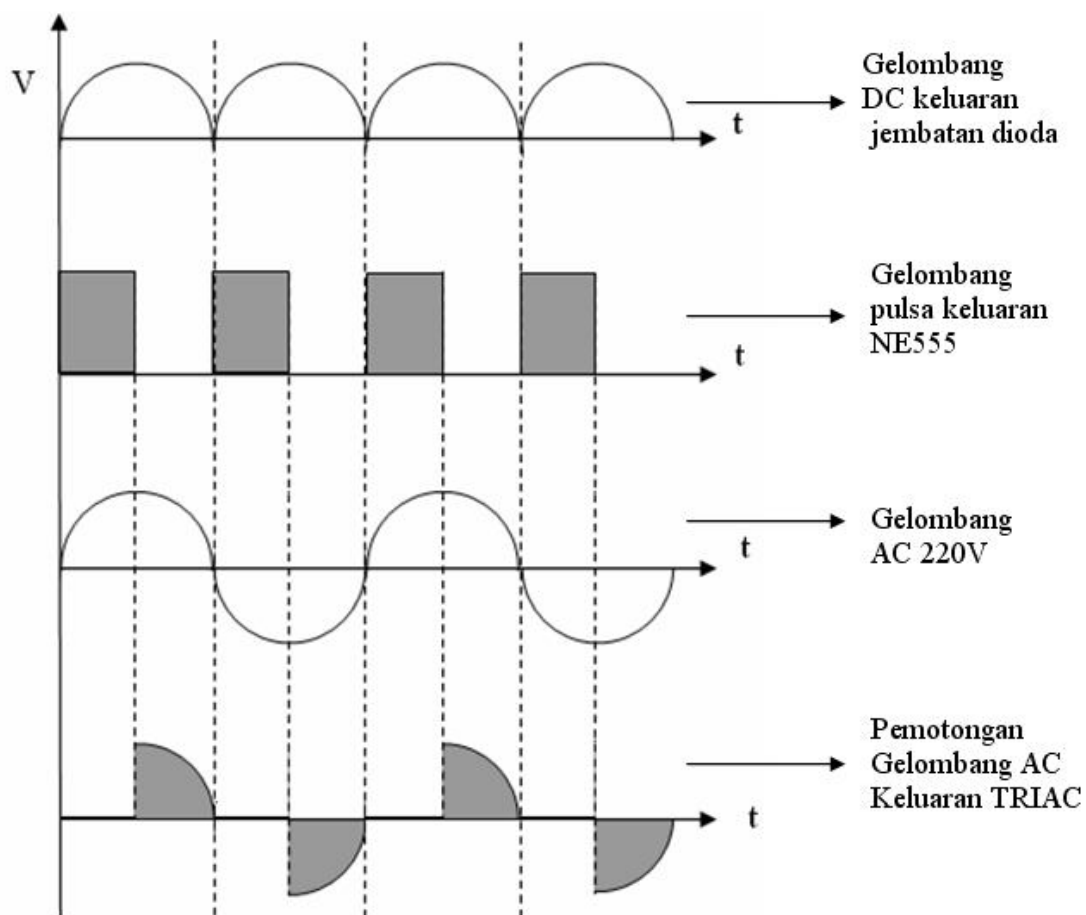
Jika digunakan $R=10k\Omega$, maka besar C adalah $0.9\mu F$. Untuk mempermudah pencarian, maka digunakan C sebesar $1\mu F$ karena terdapat dipasaran. Dalam rangkaian ini

pengendalian fasa dilakukan dengan mendeteksi kapan terjadinya perlintasan titik nol, maka setiap terjadi perlintasan titik nol, detektor memberikan sinyal kepada pengendali untuk diproses. Untuk itu, keluaran dari dioda jembatan harus diolah menjadi pulsa agar dapat diterima pengendali dengan baik, yaitu menggunakan rangkian multivibrator monostabil. Rangkaian ini digunakan agar menghasilkan gelombang kotak (pulsa detak). Tujuan untuk menghasilkan gelombang kotak (pulsa) karena pada masukan *driver* triac membutuhkan sinyal digital yang berupa gelombang kotak (pulsa) terlihat pada Gambar 4.12.



Gambar 4.12 Gelombang Pulsa Keluaran Rangkaian Detektor Nol

Pada Gambar 4.13 ditunjukkan bentuk-bentuk gelombang keluaran dari masing blok sesuai yang telah direncanakan. Gelombang ke-1 adalah gelombang *trigger* untuk NE555 yang berasal dari gelombang AC yang disearahkan dengan tegangan sebesar 5V. Gelombang ke-2 adalah keluaran NE555 yang berbentuk gelombang kotak (pulsa). Frekuensi gelombang ini adalah 100HZ karena diharapkan dapat memotong kedua siklus gelombang AC. *Duty-cyclenya* dapat diatur dengan tegangan masukan pada pin 5. Keluaran NE555 dihubungkan ke MOC3021 sebagai *driver* triac. MOC3021 akan “ON” pada saat keluaran NE555 berlogika “0”, dan sebaliknya. Sedangkan jika MOC3021 “ON” triac akan “ON”, jadi dapatdisimpulkan bahwa NE555 berlogika “terbalik” dengan MOC3021 dan triac. Hasil pemotongan gelombang pada triac dapat dilihat pada gelombang ke-4.



Gambar 4.13 Perencanaan Gelombang Keluaran dari Masing-masing Blok Rangkaian.

4.4.7 Rangkaian *Driver Triac* dan *Triac*

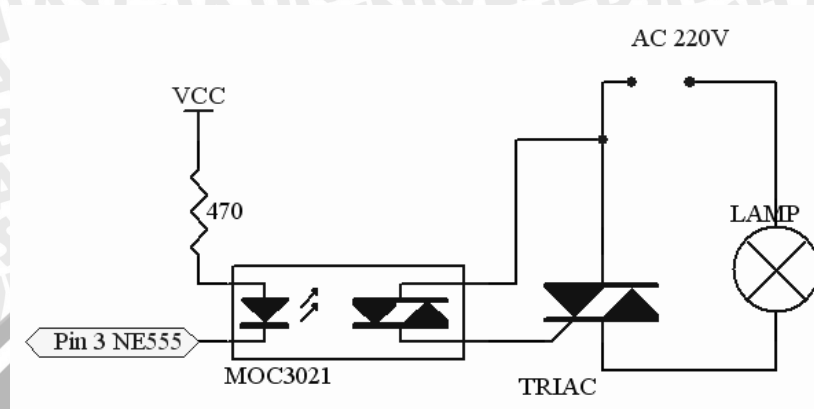
Pada perancangan rangkaian *driver* digunakannya *transistor switching* difungsikan untuk mengkondisikan tegangan dari mikrokontroler secara switching, disamping itu digunakannya optoisolator MOC3021, yang berguna sebagai coupler dan isolator antara rangkaian detektor nol dengan triac atau sebagai rangkaian pengaman didalam sistem ini. Didalam komponen ini seolah-olah terdapat led yang menyala jika diberi tegangan pada kutub anoda dan ground pada katoda. Keluaran NE555 sebagai rangkaian detektor nol adalah gelombang kotak. Maka MOC3021 akan aktif jika keluaran NE555 berlogika “0”. Didalam datasheet disebutkan bahwa $I_{FT} = 15 \text{ mA}$, I_{FT} adalah nilai arus yang digunakan untuk mentrigger led yang terdapat dalam MOC3021.

$$I_{FT} = \frac{V_{CC}}{R} \quad (4.6)$$

$$R = \frac{5V}{15mA}$$

$$R = 333.33 \Omega$$

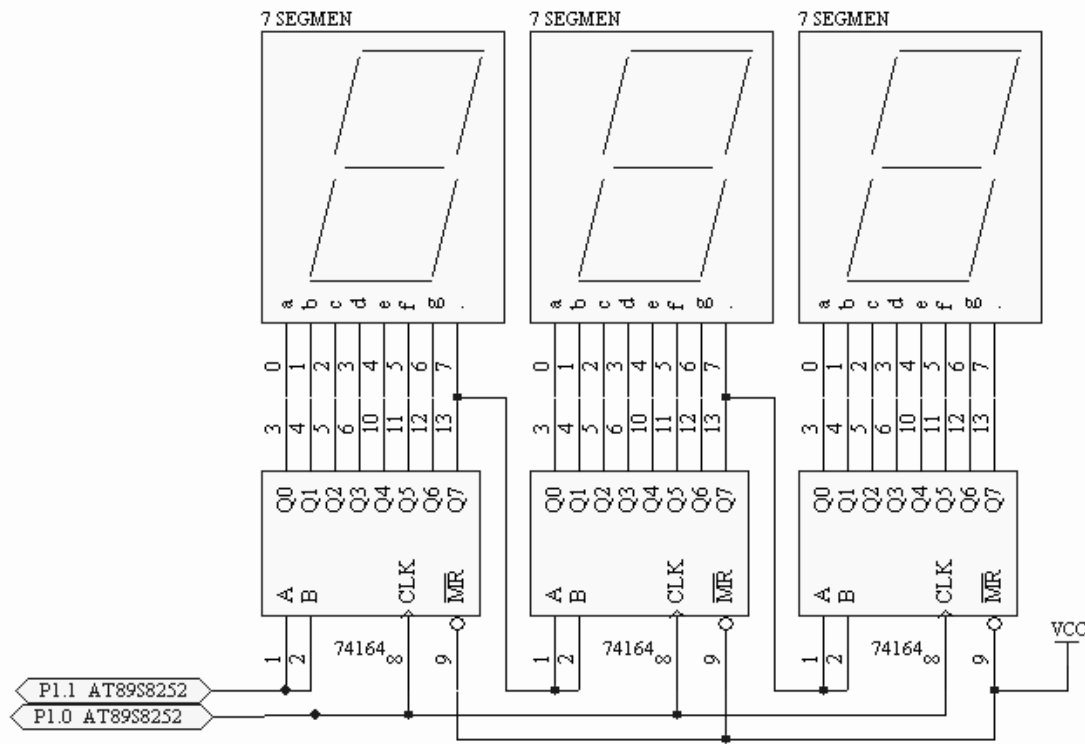
Nilai resistansi tersebut diatas tidak terdapat dipasaran, jadi diganti dengan nilai resistansi sebesar 470Ω . Keluaran MOC3021 digunakan untuk mentrigger *gate* triac. Rangkaian *driver* triac dan triac seperti pada Gambar 4.14.



Gambar 4.14 Rangkaian *Driver* Triac dan Triac

4.4.8 Rangkaian *Display* Seven Segmen

Pada perancangan ini digunakan seven segmen sebagai *display* dan IC 74164 sebagai *driver* seven segmen. Penggunaan IC ini dikarenakan untuk menghemat port mikrokontroler, selain itu untuk mempermudah *hardware* nya. Q0-Q7 IC 74164 dihubungkan ke A-G seven segment. Masukan dari mikrokontroler dihubungkan seri ke AB dari ketiga IC74164, sedangkan CLK (clock) dipararel antara satu IC dengan yang lainnya. Dat yang masuk di AB akan dikeluarkan di Q0 IC1, kemudian jika CLK diberi logika “1” maka data akan bergeser ke Q1, sedangkan Q0 sudah terisi data yang baru. Proses ini berlangsung terus-menerus sampai ke Q7 IC3 dan menggunakan waktu yang sangat cepat sehingga tidak terlihat oleh mata perubahan data tersebut. Rangkaian *display* seven segmen dapat dilihat pada Gambar 4.15.



Gambar 4.15 Rangkaian *Display Seven Segmen*

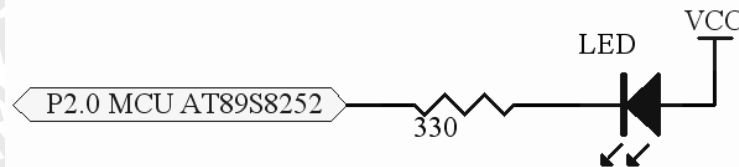
4.4.9 Rangkaian Indikator

Pada rangkaian ini digunakan led sebagai indikator lampu yang sedang diatur intensitas cahayanya. Jika lampu sedang diatur, maka led menyala dan sebaliknya. Kutub katoda led dihubungkan ke MCU, jadi jika P2.0 berlogika “0” led akan menyala sedangkan jika berlogika “1” led mati. Seperti diketahui arus led adalah 20mA, jadi dari situ dapat ditentukan nilai resistansi yang digunakan sebagai pengaman led agar tidak kelebihan arus.

$$V = I \cdot R \quad (4.7)$$

$$\begin{aligned} R &= \frac{V_{cc}}{I} \\ &= \frac{5V}{20mA} \\ &= 250 \Omega \end{aligned}$$

Nilai resistansi tersebut tidak terdapat dipasaran, maka digunakan resistansi sebesar 330 Ω . Keseluruhan rangkaian indikator dapat dilihat pada Gambar 4.16.

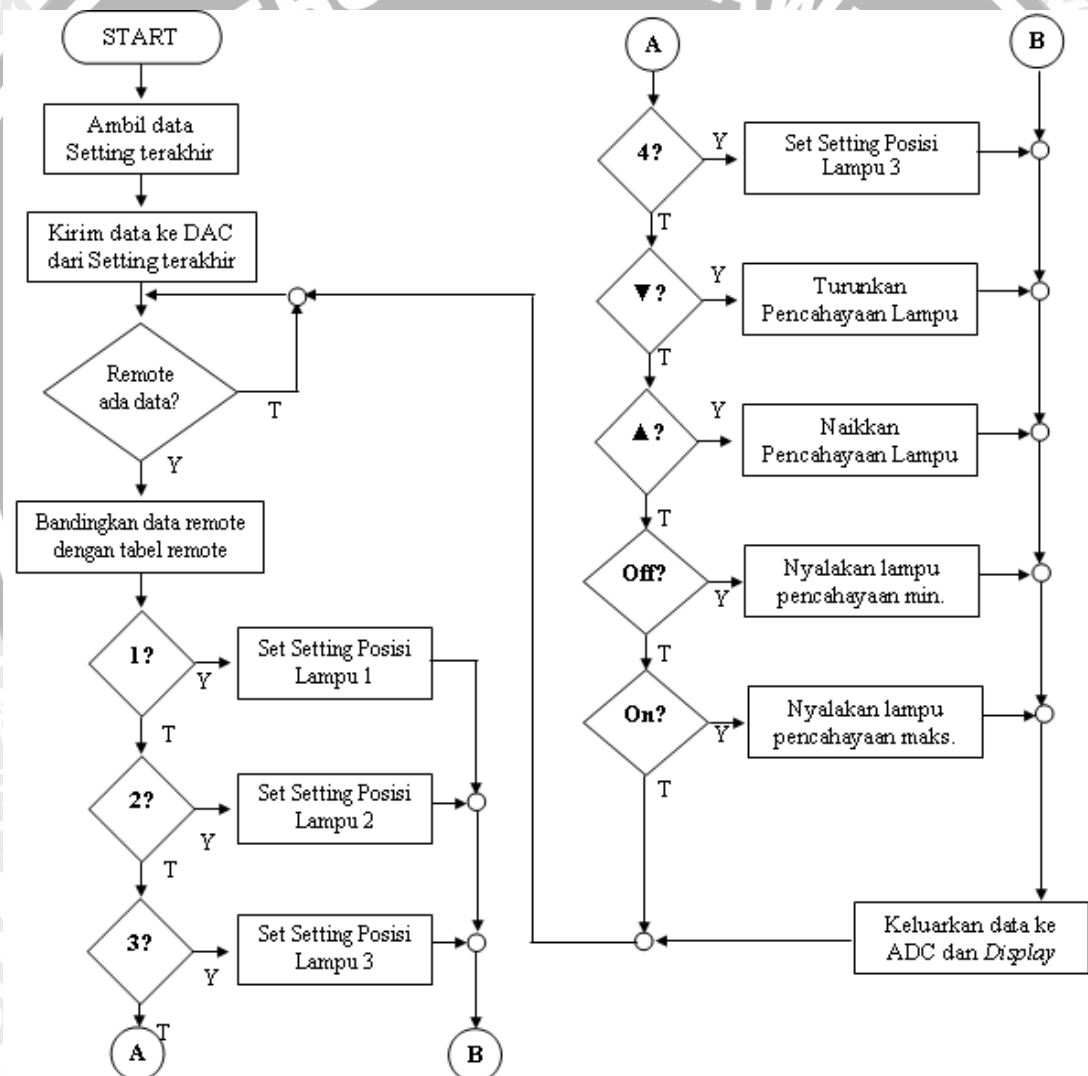


Gambar 4.16 Rangkaian Indikator

4.5 Perancangan Perangkat Lunak

4.5.1 Perancangan Perangkat Lunak Sistem

Perangkat lunak yang dirancang, dibuat dengan menggunakan bahasa *assembler* mikrokontroler AT89S8252. Diagram alir program utama dari alat pengatur intensitas cahaya lampu pijar menggunakan remote ditunjukkan pada Gambar 4.17.



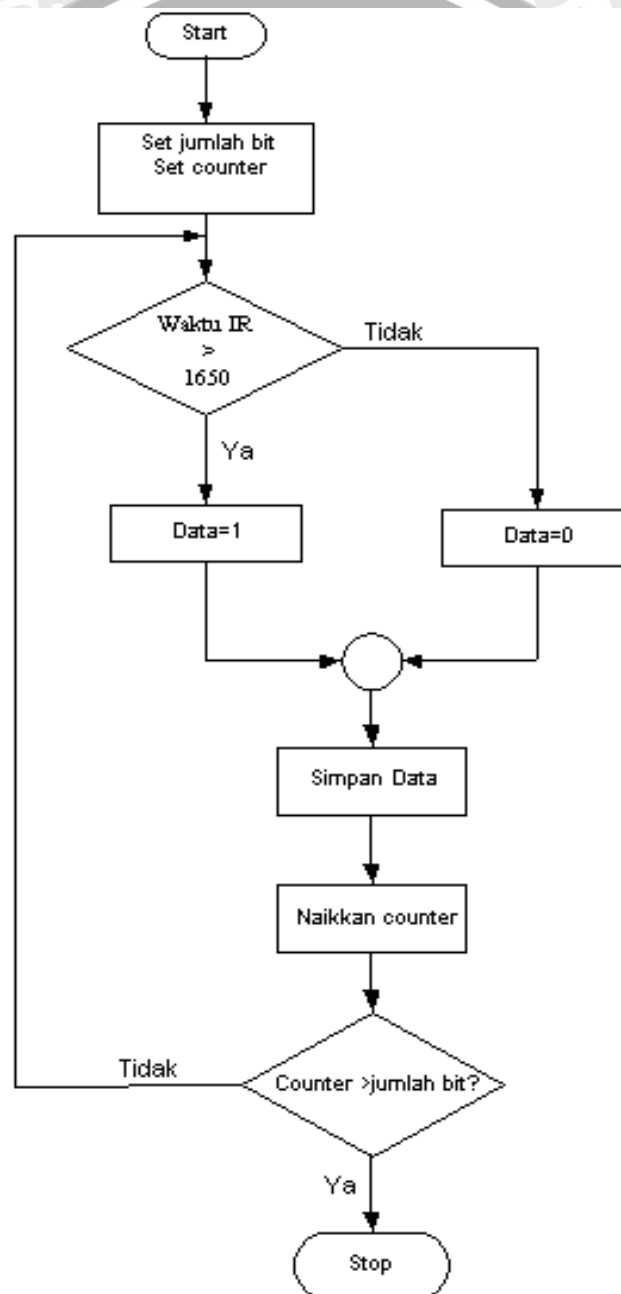
4.17 Diagram Alir Program Utama

4.5.2 Perancangan Perangkat lunak Remote control

Jenis remote control yang digunakan pada perancangan sistem ini adalah remote goldstar dengan metode pengiriman data secara space, panjang data 32 bit dan beberapa ketentuan lainnya seperti terdapat pada dasar teori dalam bab sebelumnya.

4.5.2.1 Mengambil kode remote control dengan tipe space

Gambar 4.18 menunjukkan diagram alir untuk mengambil kode remote control dengan tipe space.



Gambar 4.18 Flowchart Program Mengambil Kode Remote Kontrol Dengan Tipe Space

Pada Gambar 4.18 menunjukkan diagram alir dalam menganalisis data-data remote kontrol hasil sampling untuk menemukan kode remote control. Sebelum proses pencarian kode dilakukan, terlebih dahulu diberi nilai awal untuk jumlah bit, counter dan jumlah data 1. Jumlah bit adalah panjang data remote kontrol yang dikirimkan, 16, 24, 32 atau 48 bit. Pemberian nilai awal counter digunakan sebagai cacahan dalam proses pencarian kode. Jumlah counter ini sesuai atau sama dengan jumlah bit. Sedangkan jumlah 1 merupakan jumlah pencuplikan data yang berlogika 1. Jumlah logika 1 akan tetap besarnya karena tipe yang digunakan adalah tipe space. Jumlah logika 1 yang tetap besarnya akan digunakan sebagai pembanding dengan jumlah logika 0 yang besarnya berubah. Dengan mengetahui jumlah logika 1 dan 0, dapat dihitung pula lebar logika 1 dan lebar logika 0 dengan menggunakan persamaan berikut :

$$T1 = \Sigma \text{logika}_1 \times t_{\text{sampling}} \quad (4.8)$$

$$T0 = \Sigma \text{logika}_0 \times t_{\text{sampling}} \quad (4.9)$$

Dengan :

$$T1 = \text{Lebar pulsa logika 1 } (\mu\text{S})$$

$$T0 = \text{Lebar pulsa logika 0 } (\mu\text{S})$$

$$T_{\text{sampling}} = \text{waktu pencuplikan kode remote control}$$

Misalkan jumlah logika 1 adalah 8 kali, jumlah logika 0 adalah 24 kali dan t_{sampling} adalah 50 μS maka lebar pulsa 1 dan pulsa 0 adalah :

$$T1 = 8 \times 50$$

$$= 400 \mu\text{S}$$

$$T0 = 24 \times 50$$

$$= 1200 \mu\text{S}$$

Dari lebar pulsa tersebut dapat diketahui panjang data remote control. Jika remote kontrol menggunakan 32 bit, maka panjang datanya adalah :

$$T = (T1 + T0) \times 32 \quad (4.10)$$

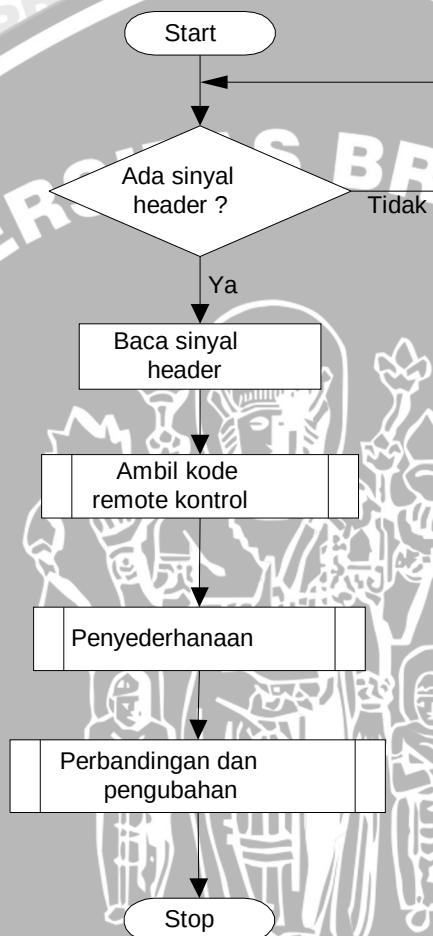
$$= (400 \mu\text{S} + 1200 \mu\text{S}) \times 32$$

$$= 51200 \mu\text{S}$$

$$= 51,2 \text{ms}$$

Nilai tersebut merupakan panjang data atau kode remote kontrol, untuk mengetahui panjang data seluruhnya adalah panjang data tersebut ditambah header. Setelah pemberian nilai awal, selanjutnya dilakukan proses pencarian kode remote kontrol dengan cara membandingkan jumlah logika 1 dengan jumlah logika 0. Jika jumlah logika 1 sama

dengan jumlah logika 0, maka data diisi dengan 0. Jika jumlah logika 0 lebih besar dari jumlah logika 1, maka data diisi dengan 1. Data tersebut akan disimpan dalam array kemudian dilakukan proses perbandingan lagi sampai jumlah counter sama dengan jumlah bit. Dengan demikian akan diperoleh jumlah kode remote kontrol seluruhnya berjumlah sama dengan jumlah bit. Pada Gambar 4.19 menjelaskan flowchart pengubah kode remote control.



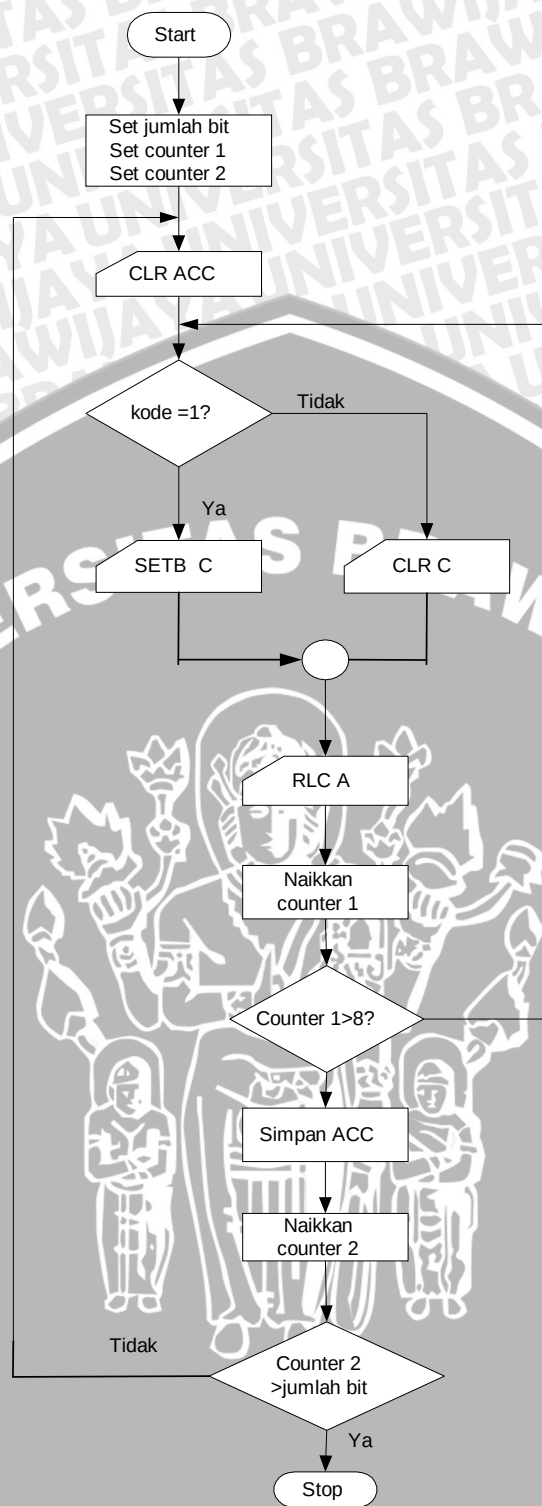
Gambar 4.19 Flowchart Pengubah Kode Remote Kontrol

4.5.2.2 Penyederhanaan

Proses penyederhanaan ini digunakan untuk menyederhanakan panjang data remote kontrol yang tadinya 32 byte menjadi 32 bit atau 4 byte. Diagram alir program penyederhanaan ditunjukkan pada Gambar 4.18. Dalam proses penyederhanaan terdapat dua counter yaitu counter 1 dan counter 2, dimana counter 1 digunakan untuk menghitung cacahan sebanyak 8 kali, sedangkan counter 2 digunakan untuk menghitung jumlah cacahan sebanyak jumlah bit.

Dalam proses penyederhanaan data, memanfaatkan accumulator dan flag carry. Sebelum dilakukan pengecekan terhadap remote kontrol, accumulator harus dibuat nol dengan memberikan perintah CLR ACC. Jika kode remote kontrol sama dengan 1 maka flag carry akan diset. Jika kode remote kontrol sama dengan 0, maka flag carry akan direset, setelah dilakukan pengesetan flag carry, langkah selanjutnya menggeser accumulator kekiri dengan perintah RLC A. Proses ini akan diulang sampai 8 kali untuk memperoleh data satu byte. Kemudian data tersebut disimpan dalam memori. Pada Gambar 4.20 ditunjukkan alur program penyederhanaan panjang data remote control.

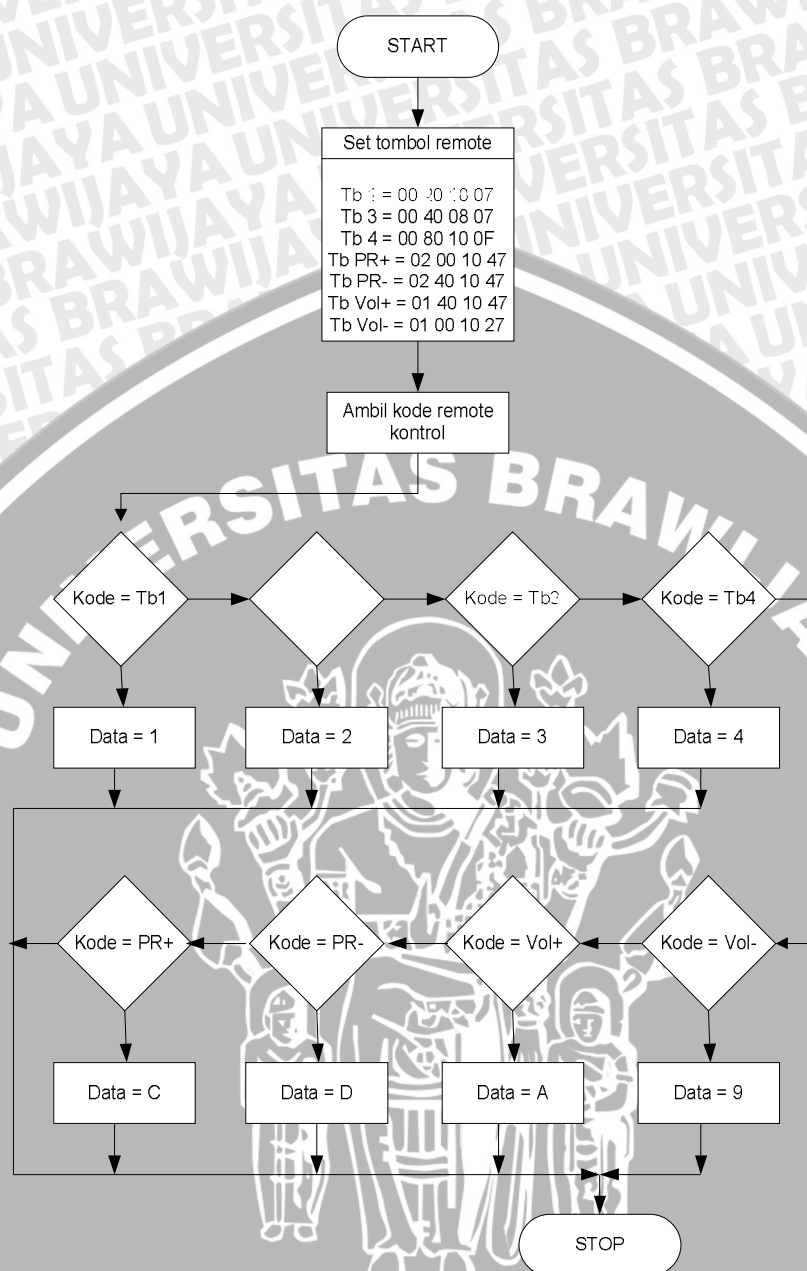




Instruksi assembler

Gambar 4.20 Flowchart Program Penyederhanaan Panjang Data Remote Kontrol

4.5.1.3 Perbandingan dan perubahan



Gambar 4.21 Flowchart Perbandingan dan Perubahan

Pada Gambar 4.21 ditunjukkan perbandingan dan perubahan data yang masuk dari sensor penerima inframerah dengan data yang telah ditetapkan dalam program.