



**ALAT PENCAMPUR WARNA CAT AIR SECARA OTOMATIS
BERBASIS MIKROKONTROLER AT89S51**

TUGAS AKHIR

Diajukan Dalam Rangka Penyelesaian Studi Diploma III
Untuk Mencapai Gelar Ahli Madya

Disusun oleh :

Nama : Dwi Didid Setyawan
NIM : 5352303504
Program Studi : D3 Teknik Elektro
Jurusan : Teknik Elektro

**FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG**

2007

HALAMAN PENGESAHAN

Laporan Tugas Akhir ini telah dipertahankan di hadapan sidang penguji Tugas Akhir Fakultas Teknik Universitas Negeri Semarang

Pada hari :

Tanggal :

Pembimbing :

Tatyantoro A, S.T,M.T

NIP. 132 232 153

Penguji II :

Penguji I :

Drs. Agus Purwanto

NIP. 131 621 386

Tatyantoro A, S.T,M.T

NIP. 132 232 153

Ketua Jurusan Teknik Elektro,

Kaprodi DIII Teknik Elektro,

Drs. Djoko Adi Widodo, MT

NIP. 131 570 064

Drs. Agus Murnomo, MT

NIP. 131 616 610

Dekan Fakultas Teknik,

Prof. Dr. Soesanto

NIP. 130875753

KATA PENGANTAR

Alhamdulillah, puji syukur kehadiran Allah SWT atas segala Rahmat dan Hidayahnya-Nya yang telah diberikan sehingga penulis dapat menyelesaikan pembuatan laporan Tugas Akhir yang berjudul “ALAT PENCAMPUR WARNA CAT AIR SECARA OTOMATIS BERBASIS MIKROKONTROLER AT89S51”. Dalam kesempatan ini penulis juga ingin mengucapkan terima kasih kepada :

1. Bpk. Prof. Dr. Soesanto, Dekan Fakultas Teknik.
2. Bpk. Drs. Djoko Adi Widodo, MT, Ketua Jurusan Teknik Elektro.
3. Bpk. Drs. Agus Murnomo, MT, Kaprodi D III Teknik Elektro.
4. Bpk. Tatyantoro A, S.T,M.T, Dosen Pembimbing Tugas Akhir.
5. Joko Pitoyo, A. Md, atas kerjasamanya.
6. Teman-teman kos AFSADA atas solidaritas dan kebersamaannya.
7. Untuk semua teman-teman TIK'03.

Atas bantuan dan bimbingan yang telah diberikan kepada penulis selama penulis menyelesaikan pembuatan tugas akhir.

Laporan Tugas Akhir ini tentunya jauh dari sempurna, oleh karena itu penulis juga mohon saran dan kritik yang membangun dari pembaca untuk kesempurnaan laporan Tugas Akhir ini. Semoga laporan ini bermanfaat bagi penulis dan pembaca semuanya serta dapat dipergunakan sebaik-baiknya.

Semarang, 1 Januari 2007

Penulis

MOTTO DAN PERSEMBAHAN

Motto

".....Sesungguhnya Allah tidak akan mengubah keadaan suatu kaum sebelum mereka mengubah keadaan mereka sendiri....."

(QS. Ar Ra'd :11)

Perjuangan dan kerja keras akan sia-sia tanpa do'a...

Persembahan

Kupersembahkan karya ini untuk:

Bapak dan Ibu tercinta yang selalu memberi doa restu dan kasih sayang.

Kakak dan Adikku tersayang.

Teman-teman AFSADA KOST yang selalu kompak.

Teman-teman seperjuangan TIK 2003.

ABSTRAK

Dwi Didid Setyawan. 2007. **Alat Pencampur Warna Cat Air Secara Otomatis Berbasis Mikrokontroler AT89S51.** Tugas Akhir. D3 Teknik Elektro. Teknik Elektro. Fakultas Teknik. Universitas Negeri Semarang.

Dewasa ini banyak cara digunakan untuk mencampur warna cat agar diperoleh hasil warna campuran yang diinginkan. Pada umumnya orang mencampur cat dengan cara manual yaitu dengan diaduk, untuk perbandingan takaran cat yang akan dicampur juga masih menggunakan perkiraan yang belum cukup akurat.

Untuk bisa menghasilkan hasil campuran warna cat yang efektif dan akurat diperlukan prosentase takaran cat yang tepat, meskipun telah dilakukan pencampuran secara berulang-ulang. Tapi pada kenyataannya orang sering tidak memperhatikan perbandingan yang telah dibuatnya. Atas dasar itulah maka penulis berusaha merancang dan membuat sebuah sistem pencampur warna cat air secara otomatis berbasis Mikrokontroler AT89S51.

Hasil yang dicapai dalam tugas akhir ini adalah sebuah miniatur Pencampur Warna Cat Air Secara Otomatis Berbasis Mikrokontroler AT89S51. Sesuai dengan hasil pengamatan, membuktikan bahwa sistem pencampur warna cat air ini mampu bekerja dengan baik, walaupun masih ada kekurangan. Salah satunya adalah sistem hanya bisa untuk mencampur cat dengan bahan dasar air, karena dikhawatirkan bila digunakan untuk mencampur cat dengan bahan dasar minyak akan menyumbat selang pada katup solenoid bila tidak sering dibersihkan.

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
HALAMAN PENGESAHAN	ii
KATA PENGANTAR	iii
MOTO DAN PERSEMBAHAN	iv
ABSTRAK	v
DAFTAR ISI	vi
DAFTAR GAMBAR	viii
DAFTAR TABEL	ix
DAFTAR LAMPIRAN	x
BAB I. PENDAHULUAN	1
A. Latar Belakang	1
B. Permasalahan	2
C. Pembatasan Masalah	2
D. Tujuan	3
E. Manfaat	4
F. Sistematika Laporan Tugas Akhir	4
BAB II. DASAR TEORI	6
A. Pengertian Dasar Pencampuran Cat	6
B. Terminologi Warna	6
C. Mikrokontroler AT89S51	7
D. Tutorial Timer MCS-51	10
E. <i>Timer</i> Mikrokontroler	16
F. Bahasa Assembler	21
1. Definisi	21
2. Listing Program	22
3. Blok Diagram Pemrograman	23
4. Sub Program	24
G. Rangkaian Pendukung	20
1. Rangkaian Pengendali Mikrokontroler AT89S51	20

H. Komponen Pendukung.....	25
1. Transistor.....	26
2. Relay.....	27
3. Motor DC	27
4. Dioda	28
5. Resistor.....	28
6. Solenoid valve	29
BAB III.PROSES PERANCANGAN	31
A. Pembuatan Hardware	31
B. Blok Diagram Alat dan Prinsip Kerja.....	33
1. Blok Diagram	33
2. Prinsip kerja.....	33
C. Pembuatan Software	38
1. Pembuatan Flowchart Sistem	38
2. Pembuatan Flowchart Program	40
3. Pembuatan Program	41
BAB IV. PENGUJIAN DAN CARA PENGOPERASIAN	54
A. Pembuatan dan Pengujian Campuran Warna	54
B. Hasil dan Cara Pengoperasian	55
1. Hasil Warna.....	55
2. Cara Pengoperasian	57
BAB V. PENUTUP	59
A. Kesimpulan	59
B. Saran.....	59
DAFTAR PUSTAKA	61
LAMPIRAN	

DAFTAR GAMBAR

	Halaman
Gambar 1. Tata Letak Pin-Pin IC Mikrokontroler Intel AT89S51	8
Gambar 2. Bagan Timer Mode 0.....	13
Gambar 3. Bagan Timer Mode 1.....	13
Gambar 4. Bagan Timer Mode 2.....	13
Gambar 5. Bagan Timer Mode 3.....	14
Gambar 6. Cara Kerja Timer.....	14
Gambar 7. Pengaturan Timer dengan Software.....	15
Gambar 8. Pengaturan Timer Dengan Trigger.....	15
Gambar 9. Rangkaian Pengendali Mikrokontroler AT89S51	25
Gambar 10. <i>Simbol Transistor NPN (a) dan Transistor PNP (b)</i>	26
Gambar 11. <i>Macam-macam bentuk fisik transistor</i>	26
Gambar 12. <i>Simbol dan Bentuk Fisik Relay Single Kontak</i>	22
Gambar 13. Konstruksi Relay	22
Gambar 14. Operasi Motor DC Magnet Permanen.....	28
Gambar 15. <i>Simbol Dioda & Bentuk Fisik Dioda</i>	28
Gambar 16. <i>Simbol Resistor</i>	29
Gambar 17. <i>Solenoid Valve</i>	29
Gambar 18. Konstruksi miniatur alat pencampur.....	31
Gambar 19. Blok Diagram Sistem Pengendali.....	33
Gambar 20. Rangkaian Konfigurasi Push Button	35
Gambar 21. Rangkaian Driver.....	36
Gambar 22. Catu Daya 12&24V	37
Gambar 23. Flowchart Sistem	38
Gambar 24. Flowchart Program	40

DAFTAR TABEL

	Halaman
Tabel 1. Fungsi-fungsi Khusus Register	17
Tabel 2. Daftar Bit SFR TMOD	18
Tabel 3. Mode Operasi <i>Timer</i>	19
Tabel 4. Bit-bit SFR TCON	19
Tabel 5. Daftar alat dan bahan.....	32
Tabel 6. Simulasi Pencampuran	54
Tabel 7. Waktu <i>Delay</i> Pencampuran	55
Tabel 8. Hasil Warna Campuran	57

DAFTAR LAMPIRAN

Lampiran 1. Rangkaian Pencampur Cat Air Otomatis Berbasis Mikrokontroler AT89S51

Lampiran 2. a. Gambar Rangkaian push button
b. Gambar push button warna
c. Gambar Layout PCB rangkaian Pengendali Mikrokontroler AT89S51

Lampiran 3. Listing Program Pencampuran Warna Cat Secara Otomatis

Lampiran 4. Data *sheet* IC Mikrokontroler Intel AT89S51

BAB I

PENDAHULUAN

A. LATAR BELAKANG

Dewasa ini perkembangan ilmu dan teknologi semakin pesat, terutama dalam bidang pengendali. Banyak sekali peralatan industri yang memanfaatkan kemajuan di bidang pengendali sebagai upaya pengaplikasian kendali otomatis.

Ide atau gagasan penulis untuk membuat alat pencampur / pengoplos warna cat otomatis yang menggunakan aplikasi Mikrokontroler AT89S51 ini diperoleh dari survei bahwa masih banyak bengkel pengecatan *bodi* kendaraan baik mobil maupun motor yang masih mengoplos cat dengan cara manual. Di samping itu juga penulis mempunyai ide untuk mengembangkan hasil Tugas Akhir sehingga lebih mudah untuk diaplikasikan. Tugas Akhir yang akan penulis kembangkan berjudul “Alat Pencampur Cat Berbasis PC (*Personal Computer*) dengan Menggunakan Program Delphi ”.

Pada Tugas Akhir tersebut masih terdapat kelemahan dan kekurangan antara lain kurangnya efisien dan efektif dalam sistem pengendalinya. Karena pada TA tersebut masih menggunakan PC (*Personal Computer*) yang tentunya memakan tempat yang cukup luas dalam pemakaiannya serta masih dibutuhkannya operator / orang yang bisa menjalankan Program Delphi supaya bisa menjalankan proses pengoplosan cat tersebut. PC juga mempunyai kelemahan yaitu apabila sewaktu-waktu jaringan listrik PLN terjadi gangguan atau putus, maka program di dalam komputer tersebut juga

akan terganggu, yang mengakibatkan terhentinya proses pengoplosan cat. Sedangkan untuk menghidupkan komputer supaya program proses pengoplosan bisa berjalan kembali, perlu dibutuhkan waktu yang cukup lama yang dikarenakan *booting* awal komputer.

Tugas Akhir yang sedang penulis kerjakan ini adalah berupa miniatur alat mencampur cat air menggunakan pengendali IC Mikrokontroler dari keluarga MCS-51, sedangkan sebagai pengatur keluaran aliran cat pada masing-masing warna dasar bisa digunakan katup yang dioperasikan secara elektrik maupun mekanik. Penulis berharap bisa meningkatkan akurasi dan efektivitas dalam mencampur / mengoplos warna cat tanpa mengganggu proses pengoplosan yang sedang berlangsung.

Alat pengoplos cat miniatur ini dapat diterapkan pada karoseri dan bengkel-bengkel pengecatan lainnya sehingga akan membantu dalam proses pencampuran cat.

B. PERMASALAHAN

Masalah yang diangkat dalam hal ini adalah bagaimana membuat alat pencampur warna cat otomatis yang efisien, ringkas dan bisa didapatkan hasil pencampuran yang akurat meskipun proses pencampuran tersebut telah berulang kali dilakukan.

C. PEMBATAHAN MASALAH

Untuk menghindari salah penafsiran dalam permasalahan maka diberikan pembatasan masalah sebagai berikut:

1. Peralatan miniatur ini terbatas pada pembuatan rangkaian pengendali alat pencampuran warna cat secara otomatis dengan menggunakan IC Mikrokontroler AT89S51 sebagai komponen pengendali utamanya.
2. Sebagai pengatur keluaran aliran cat pada masing-masing warna cat dasar (merah, biru, kuning) menggunakan *solenoid valve* dengan ukuran $\frac{1}{4}$ inci, sedangkan sebagai pengaduk hasil campuran cat digunakan motor DC 12 Volt.
3. Pada pembuatan alat miniatur pencampur warna cat otomatis ini dikhususkan untuk cat dengan bahan dasar air, untuk mempermudah pengaturan aliran catnya.
4. Alat miniatur ini hanya menghasilkan hasil campuran warna cat sebanyak 12 macam.

D. TUJUAN

Tujuan yang ingin dicapai dalam penulisan dan pembuatan Tugas Akhir ini adalah untuk membuat alat pencampur warna cat yang bisa membuat takaran dan sekaligus mengaduknya secara otomatis berbasis Mikrokontroler AT89S51, sehingga dapat mengurangi kesalahan dan kelalaian yang disebabkan oleh manusia.

E. MANFAAT

1. Memberikan pemahaman bagi para pembaca mengenai suatu alat yang diprogram dan dikendalikan dengan menggunakan Mikrokontroler AT89S51.
2. Hasil dari alat dan penelitian diharapkan dapat memberikan sumbangsih dalam bentuk pengetahuan dan teknik di lingkungan Fakultas Teknik Elektro Universitas Negeri Semarang.
3. Hasil dari penelitian ini diharapkan dapat memberikan alternatif lain yang lebih efektif dan efisien kepada bengkel-bengkel pengecatan dan karoseri dalam proses pencampuran warna cat.

F. SISTEMATIKA TUGAS AKHIR

1. Bagian awal

Bagian ini terdiri dari halaman judul, abstrak, pengesahan, motto, persembahan, kata pengantar, daftar isi, daftar gambar, daftar tabel dan lampiran.

2. Bagian isi

BAB I : PENDAHULUAN

Pendahuluan meliputi latar belakang, permasalahan, pembatasan masalah, penegasan istilah, tujuan penelitian, manfaat dan sistematika tugas akhir.

BAB II : DASAR TEORI

Bab ini meliputi dasar teori alat dan komponen yang akan digunakan.

BAB III : PROSES PERANCANGAN

Bab ini meliputi proses pembuatan alat, blok diagram alat dan cara kerja, serta pembuatan software.

BAB IV : PENGUJIAN DAN CARA PENGOPERASIAN

Bab ini meliputi pengambilan data dari alat yang sudah di uji coba dan cara pengoperasian alat..

BAB V : PENUTUP

Penutup meliputi kesimpulan dan saran.

3. Bagian akhir

Bagian ini terdiri dari daftar pustaka dan lampiran.

BAB II

DASAR TEORI

A. Pengertian Dasar Pencampuran Cat

Berasal dari kata campur yang berarti berbaur; menjadi satu. Mencampur berarti menyatukan atau mengumpulkan supaya menjadi satu atau tidak terpisah (terhadap dua atau lebih barang atau hal) (anwar Dessy.2001. Kamus Lengkap Bahasa Indonesia. Surabaya: KARYA ABDITAMA).

Dalam Tugas Akhir ini pencampuran dilakukan pada Ke-3 warna cat primer sesuai dengan prosentase perbandingan tertentu, sehingga menghasilkan warna cat yang berbeda.

B. Terminologi Warna

Warna yang kita cerap, sangat ditentukan oleh adanya pancaran cahaya. Warna benda-benda yang kita lihat sesungguhnya adalah pantulan dari cahaya yang menyimpannya, karena warna merupakan unsur cahaya. Warna yang bersumber dari cahaya disebut warna *aditif*. Contohnya adalah warna-warna yang dipancarkan oleh televisi dan *sign lamp*. Sedangkan warna-warna pada benda, dedaunan, lukisan atau cat termasuk warna *pigmen*, yakni butir-butir halus bahan warna. (Affendi, Yusuf. 1978. Disain Warna, Susunan dan Fungsinya. Bandung: Proyek Pengemb. Ilmu dan Teknologi ITB Bandung)

Pada Tugas Akhir ini penulis menggunakan warna pigmen. Yang terdiri dari 3 warna primer (merah, biru dan kuning) sebagai warna dasar campuran.

C. Mikrokontroler AT89S51

AT89S51 adalah Mikrokontroler keluaran Atmel dengan 4K byte Flash PEROM (*Programmable and Erasable Read Only Memory*), AT89S51 merupakan memori dengan teknologi *nonvolatile* memory, isi memory tersebut dapat diisi ulang ataupun dihapus berkali-kali (Panduan Praktis Teknik Antarmuka dan Pemrograman Mikrokontroler AT89S51, 2003:1).

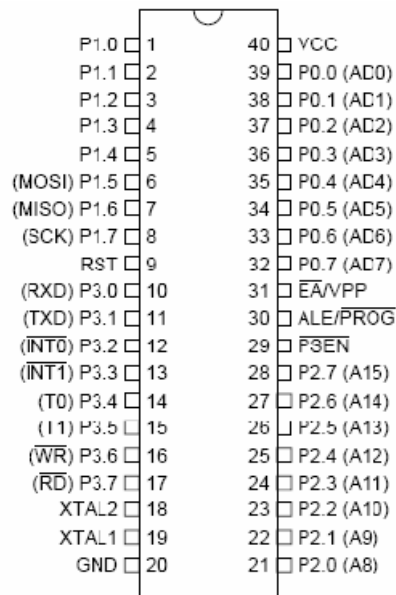
IC ini merupakan sebuah pusat penyimpanan sistem / program yang masih berbentuk bahasa *assembly*. Program tersebut nantinya akan digunakan untuk mengendalikan *solenoid valve* dan motor DC yang sebelumnya telah diterjemahkan oleh piranti *interface internal* mikro, yang berfungsi sebagai port keluar masuknya sinyal / data secara digital. Mikrokontroler umumnya terdiri dari CPU, Memory, I/O tertentu dan unit pendukung, misalnya *Analog to Digital Converter* (ADC) yang sudah terintegrasi di dalam Mikrokontroler tersebut. Kelebihan utama dari Mikrokontroler ialah telah tersedianya RAM dan peralatan I/O Pendukung sehingga ukuran board Mikrokontroler menjadi sangat ringkas dibanding dengan PC. Dalam hal ini jenis IC mikro yang digunakan adalah ATMEL dan berasal dari keluarga MCS-51.

Mikrokontroler AT89S51 memiliki sejumlah kelebihan yaitu sebagai berikut :

1. Sebuah CPU 8 bit yang termasuk keluarga MCS-51
2. Osilator internal dan rangkaian pewaktu.
3. RAM internal 128 byte (on chip)
4. Empat buah programmable port I/O, masing-masing terdiri atas 8 buah jalur I/O.

5. Dua buah *timer/counter* 16 bit.
6. Enam buah jalur interupsi.
7. Sebuah port serial dengan kontrol *serial full duplex* UART.
Kemampuan melaksanakan operasi perkalian, pembagian dan operasi boolean (bit).
8. Kecepatan pelaksanaan instruksi persiklus 1 mikro detik pada frekuensi clock 24 MHz.

Dengan keistimewaan di atas pengendali Mikrokontroler AT89S51 menjadi lebih sederhana dan tidak memerlukan IC pendukung yang banyak. Boleh dikatakan Mikrokontroler intel AT89S51 ini mempunyai keistimewaan dari segi perangkat keras.



Gambar 1. Tata Letak Pin-Pin IC Mikrokontroler Intel AT89S51

(*Microcontroller with 4K Bytes In-System Programmable Flash,2001:2*)

Susunan pin-pin IC Mikrokontroler AT89S51 diperlihatkan pada gambar diatas. Penjelasan dari masing-masing pin adalah sebagai berikut :

1. Pena 1 sampai 8 (Port 1) merupakan port paralel 8 bit dua arah yang dapat digunakan untuk berbagai keperluan (*general purpose*).
2. Pena 9 (Reset) adalah masukan reset (aktif tinggi) yang digunakan untuk mereset *program counter* sehingga program dilaksanakan mulai dari *address 000H*.
3. Pena 10 sampai 17 (Port 3) adalah paralel port 8 bit dua arah yang memiliki fungsi pengganti. Fungsi pengganti meliputi TXD (*Transmit Data*), RXD (*Receive Data*), Int0 (*Interrupt 0*), Int1 (*Interrupt 1*), T0 (*Timer 0*), T1 (*Timer 1*), WR (*Write*), dan RD (*Read*). Bila fungsi pengganti tidak dipakai pada pin-pin ini dapat digunakan sebagai port paralel 8 bit serbaguna.
4. Pin 18 (XTAL 2) adalah pin keluaran ke rangkaian osilator *internal*. Pin ini dipakai bila menggunakan osilator kristal.
5. Pin 19 (XTAL 1) adalah pin masukan ke rangkaian osilator *internal* sebuah osilator kristal atau sumber osilator luar dapat digunakan.
6. Pin 20 (*Ground*) dihubungkan ke Vss atau *ground*.
7. Pin 21 sampai 28 adalah port paralel 2 (port 2) selebar 8 bit dua arah. Port 2 ini mengirim byte alamat bila dilakukan pengaksesan memori *eksternal*.
8. Pin 29 adalah pin PSEN (*program store enable*) yang merupakan sinyal pengontrol yang membolehkan program memori eksternal

masuk ke dalam bus selama proses pemberian/pengambilan instruksi (*fetching*).

9. Pin 30 adalah ALE (*address latch enable*) yang digunakan untuk menahan alamat memori *eksternal* selama pelaksanaan instruksi.
10. Pin 31 (EA). Bila pin ini diberi logika tinggi, Mikrokontroler akan melaksanakan instruksi dari ROM/EPROM ketika isi program counter kurang dari 4096. Bila diberi logika rendah, Mikrokontroler akan melaksanakan seluruh instruksi dari memori program luar.
11. Pin 32 sampai 39 (port 0) merupakan port paralel 8 bit *open drain* dua arah. Bila digunakan untuk mengakses memori luar, port ini akan mengkombinasikan alamat memori dengan data.
12. Pin 40 (Vcc) dihubungkan ke Vcc (+5 Volt).

D. TUTORIAL TIMER MCS-51

Timer 89C51

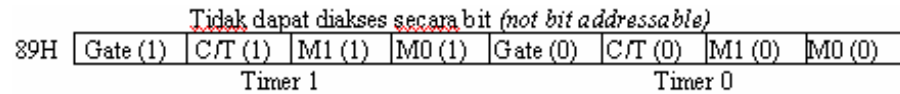
- 89C51 mempunyai dua buah timer yaitu Timer 0 dan Timer 1
- Masing-masing Timer terdiri dari 16 bit counter yang bersifat programmable

Register-register Timer

- Timer Mode Register (TMOD) di alamat 89H
- THx dan TLx
- Timer Control Register (TCON) di alamat 88H

Timer Mode Register

TMOD



Gate : Timer akan berjalan bila bit ini set dan INT0 (untuk Timer 0) atau INT1 (untuk Timer 1) berkondisi high

C/T : 1 = Counter

0 = Timer

M1 & M0: Untuk memilih mode timer

THx dan TLx (x adalah nomor Timer)

Merupakan Register yang menunjukkan nilai dari timer di mana masing-masing Timer mempunyai dua buah register yaitu:

- THx untuk high byte
- TLx untuk low byte

TH0 : Timer 0 High Byte terletak pada alamat 8AH

TL0 : Timer 0 Low Byte terletak pada alamat 8BH

TH1 : Timer 1 High Byte terletak pada alamat 8CH

TL1 : Timer 1 Low Byte terletak pada alamat 8DH

1.1. Timer Control Register (TCON)

Pada register ini, hanya 4 bit saja yaitu TCON.4, TCON.5, TCON.6 dan TCON.7 saja yang mempunyai fungsi berhubungan dengan timer.

TCON

	TCON.7	TCON.6	TCON.5	TCON.4	TCON.3	TCON.2	TCON.1	TCON.0
88	TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Register Timer

TCON.7 atau TF1: Timer 1 Overflow Flag yang akan set bila timer overflow. Bit ini dapat di-clear oleh software atau oleh hardware pada saat program menuju ke alamat yang ditunjuk oleh interrupt vector.

TCON.6 atau TR1: 1 = Timer 1 aktif

0 = Timer 1 non aktif

TCON.5 atau TF0: Sama dengan TF1

TCON.4 atau TR0: Sama dengan TR1

TCON.3 hingga TCON.0 akan dibahas pada bagian interrupt

Mode Timer

Mode Timer terdiri dari:

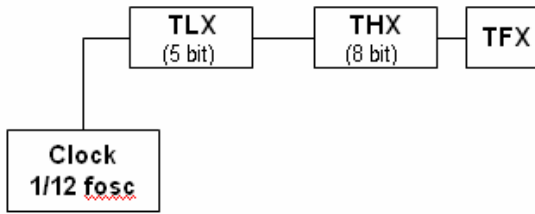
- Mode 0 Timer 13 bit
- Mode 1 Timer 16 bit
- Mode 2 Timer 8 bit auto reload
- Mode 3 Split Timer

Mode 0 (13 Bit Timer)

Tidak dapat diakses secara bit (not bit addressable)

89H	Gate (1)	C/T (1)	M1 (1)	M0 (1)	Gate (0)	C/T (0)	M1 (0)	M0 (0)
	X	X	0	0	X	X	X	X

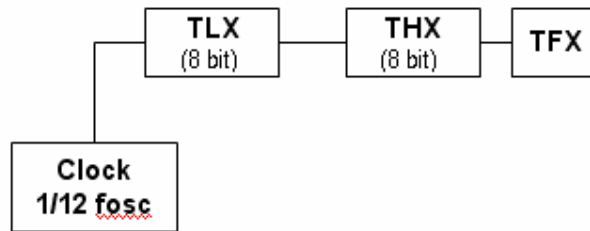
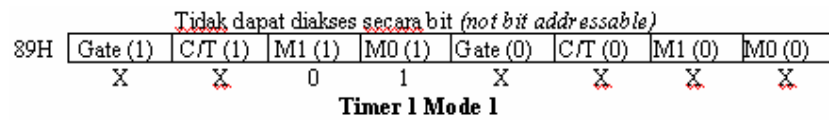
Timer 1 Mode 0



Gambar 2. Bagan Timer Mode 0

Mode 1 (16 Bit Timer)

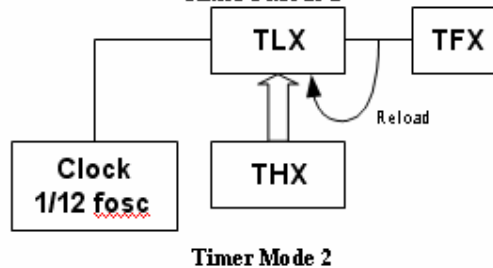
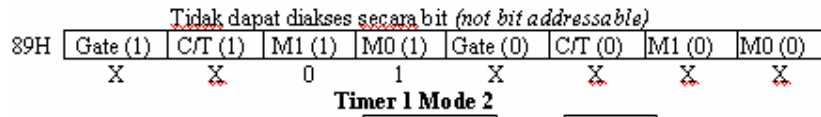
TMOD



Gambar 3. Bagan Timer Mode 1

Mode 2

TMOD

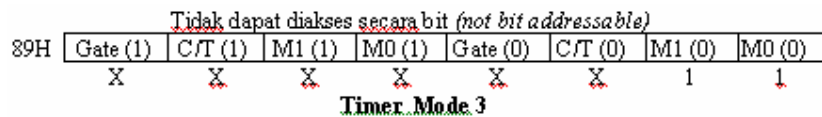


Timer Mode 2

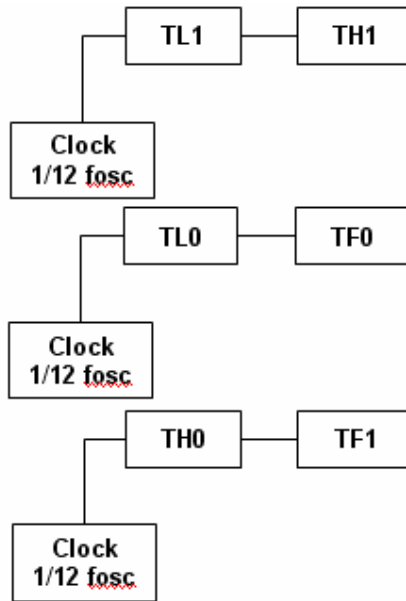
Gambar 4. Bagan Timer Mode 2

Mode 3

TMOD

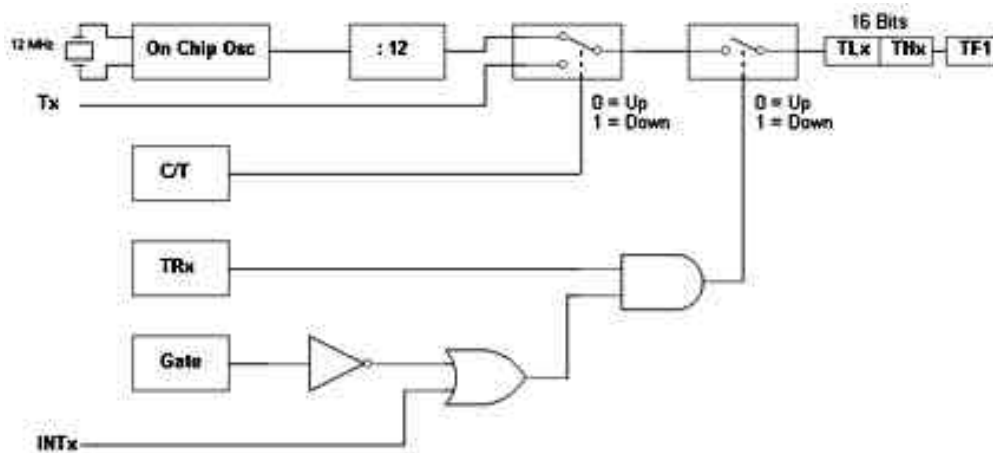


Pada Mode ini, Timer 0 terpisah menjadi 2 buah 8 bit timer dan TF1 dari Timer 1 tidak akan berpengaruh walaupun Timer 1 overflow. Timer 1 masih dapat digunakan untuk mode-mode yang lain seperti baud rate serial



Gambar 5. Bagan Timer Mode 3

Cara Kerja Timer



Gambar 6. Cara Kerja Timer

OPERASI TIMER

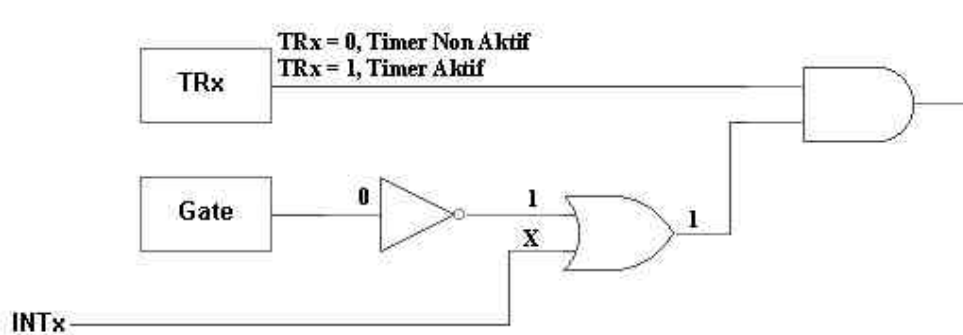
Timer dapat bekerja dengan sumber clock dari:

- Internal berdasarkan $1/12$ frekwensi oscillator, $C/T = 0$
- External berdasarkan trigger dari Tx (T0 untuk Timer 0 dan T1 untuk Timer 1) $C/T = 1$

Timer dapat bekerja berdasar trigger dari:

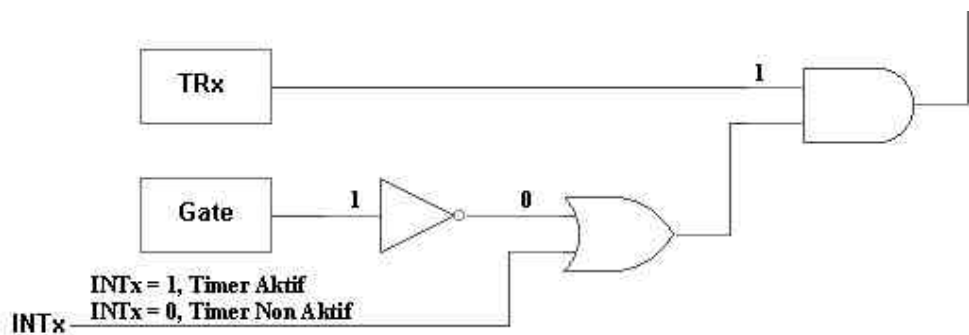
- Software (Bit TRx)
- Hardware (kaki INTx)

Timer ditrigger dari software



Gambar 7. Pengaturan Timer dengan Software

Timer Di trigger dari Hardware



Gambar 8. Pengaturan Timer Dengan Trigger

E. *Timer* Mikrokontroler

Mikrokontroler AT89S51/C51 hadir dengan dua *timer*, keduanya bisa dikontrol, diset, dibaca, dan dikonfigurasi sendiri-sendiri. *Timer* memiliki tiga fungsi umum, yaitu:

1. Menghitung waktu antara dua kejadian (*event*)
2. Menghitung jumlah kejadian itu sendiri
3. Membangkitkan *baud rate* untuk port serial

Sebuah *timer* bekerja dengan mencacah. Tidak tergantung pada fungsi sebagai *timer*, *counter*, atau generator *baud rate*, sebuah *timer* akan selalu ditambah satu oleh Mikrokontroler.

Menggunakan *Timer* Untuk Mengukur Waktu

Saat sebuah *timer* digunakan untuk mengukur waktu, dia akan bertambah satu setiap satu siklus mesin. Setiap siklus mesin membutuhkan 12 pulsa kristal. Maka apabila sebuah 89C51 dengan kristal 11,059 MHz, maka *timer* setiap detiknya akan berharga:

$$\frac{11,059 \text{ MHz}}{12 \text{ pulsa}} = 921.583 \text{ pencacahan}$$

Dengan kata lain, terdapat 921.583 kali pencacahan dalam setiap detiknya. Tidak seperti instruksi-instruksi yang bisa memakan satu hingga empat siklus mesin, sebuah *timer* selalu konsisten bertambah satu setiap satu kali siklus mesin.

Melalui cara serupa, apabila diinginkan mendapatkan pewaktu 0,005 detik, maka dibutuhkan *timer* yang mencacah hingga:

$$0,005 \times 921.583 = 46.07,915$$

Ini berarti kita perlu memonitor cacahan dari *timer* hingga mencapai harga 4608. Walaupun tidak benar-benar presisi karena menghilangkan hitungan 0,915, namun cukup mendekati dan dapat ditolerir.

Timer SFR

Sebagaimana telah di singgung di atas, 89C51 memiliki dua buah *timer* yang setiap fungsinya identik. *Timer* pertama disebut dengan *TIMER0* dan *timer* kedua disebut dengan *TIMER1*. Kedua *timer* saling berbagi dua macam SFR, yaitu TMOD dan TCON, yang mengontrol *timer*, dan masing-masing *timer* memiliki dua macam SFR yang spesifik yaitu TH0/TL0 untuk *TIMER0* dan TH1/TL1 untuk *TIMER1*.

Untuk lebih jelasnya lihat tabel di bawah ini.

Nama SFR	Keterangan	Alamat
TH0	<i>Timer 0 High Byte</i>	8Ch
TL0	<i>Timer 0 Low Byte</i>	8Ah
TH1	<i>Timer 1 High Byte</i>	8Dh
TL1	<i>Timer 1 Low Byte</i>	8Bh
TCON	<i>Timer Control</i>	88h
TMOD	<i>Timer Mode</i>	89h

Tabel 1. Fungsi-fungsi Khusus Register

TIMER0 memiliki dua macam SFR yang eksklusif bagi dirinya sendiri, yaitu TH0 dan TL0 yang membentuk harga aktual dari *timer*. Misalnya *TIMER0* berharga 1000, maka TH0 akan berisi 3 sedangkan TL0 akan berisi 232. Untuk melihat harga sebenarnya, kalikan harga TH0 dengan 256 dan kemudian tambahkan dengan TL0.

TIMER1 identik dengan *TIMER0* kecuali bahwa SFR eksklusif yang dimilikinya adalah TH1 dan TL1. Dan karena kedua *timer* ini memiliki

kapasitas dua *byte*, maka harga maksimum yang bisa ditampung adalah 65.535. Dengan demikian, apabila *timer* telah melampaui harga 65.535, maka dia akan *reset* atau *overflow* dan kemudian kembali ke harga awal 0.

SFR TMOD

SFR TMOD digunakan untuk mengontrol mode operasi dari kedua *timer*. Setiap bit dari SFR ini menyediakan informasi bagi Mikrokontroler bagaimana menjalankan *timer*. Empat bit orde tinggi (bit 4 hingga bit 7) berhubungan dengan *TIMER1*, sedangkan empat bit orde bawah (bit 0 hingga bit 3) mempunyai fungsi sama yang diperuntukkan bagi *TIMER0*.

Bit	Nama	Fungsi	Timer
7	GATE 1	Jika bit ini diset, <i>timer</i> hanya akan bekerja jika INT1 (P3.3) berlogika 1. Jika bit ini dinolkan, <i>timer</i> akan bekerja tanpa dipengaruhi kondisi INT1	1
6	C/T 1	Jika bit ini diset, <i>timer</i> akan menghitung kondisi pada T1 (P3.5). Jika bit ini dinolkan, <i>timer</i> akan bertambah satu setiap siklus mesin	1
5	T1M 1	Bit mode <i>timer</i>	1
4	T1M 0	Bit mode <i>timer</i>	1
3	GATE 0	Jika bit ini diset, <i>timer</i> hanya akan bekerja jika INTO (P3.2) berlogika 1. Jika bit ini dinolkan, <i>timer</i> akan bekerja tanpa dipengaruhi kondisi INTO	0
2	C/T 0	Jika bit ini diset, <i>timer</i> akan menghitung kondisi pada T0 (P3.4). Jika bit ini dinolkan, <i>timer</i> akan bertambah satu setiap siklus mesin	0
1	T0M 1	Bit mode <i>timer</i>	0
0	T0M 0	Bit mode <i>timer</i>	0

Tabel 2. Daftar Bit SFR TMOD

Seperti terlihat pada tabel di atas, ada 4 bit yang menyatakan mode untuk kedua *timer*. Masing-masing dua bit untuk satu *timer*. Adapun mode operasi yang dimaksud di sini adalah sebagaimana tercantum dalam tabel di bawah ini.

TxM1	TxM0	Mode Timer	Keterangan
0	0	0	<i>Timer 13 bit</i>
0	1	1	<i>Timer 16 bit</i>
1	0	2	<i>Timer 8 bit (auto reload)</i>
1	1	3	<i>Mode Split Timer</i>

Tabel 3. Mode Operasi *Timer*

Mode *Timer* 16 Bit (Mode 1)

Timer mode 1 adalah *timer* 16 bit. Mode ini adalah mode yang paling umum digunakan. Fungsinya sama dengan *timer* 13 bit, namun yang di gunakan adalah 16 bit. TLx akan mencacah dari 0 hingga 255. Jika TLx melebihi 255, dia akan reset menjadi 0 dan menambah THx dengan 1. Karena kemampuan 16 bit, maka mode ini memiliki batas maksimum harga 65.535. Sehingga jika *timer* diset dalam mode ini, dia akan menjadi nol setelah 65.535 siklus mesin.

SFR TCON

SFR ini mengontrol kedua *timer* dan menyediakan informasi yang sangat berguna berkaitan dengan *timer-timer* tersebut. Struktur SFR TCON dapat dilihat pada tabel di bawah ini.

Bit	Nama	Alamat	Fungsi	<i>Timer</i>
7	TF1	8Fh	<i>Timer 1 Overflow</i> . Bit ini diset oleh Mikrokontroler jika <i>Timer 1 overflow</i>	1
6	TF0	8Eh	<i>Timer 1 Run</i> . Jika bit ini diset maka <i>Timer 1</i> akan bekerja. Sebaliknya jika direset maka <i>Timer 1</i> akan mati.	1
5	TR1	8Dh	<i>Timer 0 Overflow</i> . Bit ini diset oleh Mikrokontroler jika <i>Timer 0 overflow</i>	0
4	TR0	8Ch	<i>Timer 0 Run</i> . Jika bit ini diset maka <i>Timer 0</i> akan bekerja. Sebaliknya jika direset maka <i>Timer 1</i> akan mati.	0

Tabel 4. Bit-bit SFR TCON

Dalam tabel hanya dicantumkan 4 bit dari 8 bit yang ada pada SFR TCON. Hal ini karena hanya 4 bit (bit 4 hingga bit 7) yang berkaitan dengan *timer*, sedangkan bit sisanya berkaitan dengan interupsi yang akan dibahas selanjutnya. Untuk mengeset atau mereset bit-bit SFR tidak perlu dengan memberikan nilai 8 bit. Bit-bit SFR bisa dialamati per bit. Dengan demikian perubahan satu atau beberapa bit tidak akan mengganggu status bit-bit yang lain.

Membaca Status *Timer*

Membaca status *timer* ada dua cara. Yang pertama dengan membaca harga aktual 16 bit dari *timer*, dan yang kedua adalah mendeteksi apakah *timer* menemui kondisi *overflow*. Jika *timer* yang digunakan adalah *timer* mode 8-bit, pembacaan harga aktual cukup mudah. Bacalah harga 1 *byte* tersebut dan selesailah sudah. Namun jika *timer* yang digunakan adalah mode 13 bit atau 16 bit, permasalahan menjadi lebih rumit. Bagaimana jika harga aktual *low byte* adalah 255 dan pembacaan *high byte* adalah 15. Seharusnya harga sebenarnya adalah *high byte* 14 dan *low byte* adalah 255, karena saat membaca *low byte* sebesar 255, beberapa sesaat kemudian *high byte* akan bertambah satu saat pembacaan, sehingga pembacaan menjadi meleset sebesar 256 hitungan karena terletak pada *high byte*.

Pemecahannya adalah dengan membaca *high byte* terlebih dahulu dan kemudian membaca *low byte*. Setelah itu *high byte* dibaca lagi dan kemudian dibandingkan dengan pembacaan semula, bila berbeda, maka yang dipakai adalah pembacaan *high byte* yang pertama. Kadang yang perlu diketahui hanyalah saat *timer* reset menjadi nol. Dengan kata lain, tidak penting berapa

harga aktual dari *timer*, namun kapan *timer overflow* dan kembali menjadi nol. Saat *overflow*, Mikrokontroler secara otomatis mengeset bit TFX dalam register TCON. Ini berarti pengecekan *overflow* cukup dengan mengecek apakah bit TFX set atau tidak. Dengan cara ini bisa dibuat program untuk menentukan selang yang pasti. Dari pembahasan sebelumnya, diketahui bahwa untuk mendapatkan selang 0,005 detik diperlukan pencacahan hingga 4608 kali. Penggunaan mode ini akan menginisialisasi *timer* dengan harga selisih antara 65.535 dan 4608, yaitu 60.927. Sehingga 4608 cacahan berikutnya setelah 60.927 akan menyebabkan *timer overflow*.

Segmen Program *Timer*:

1. MOV TH0,#0EDH ; (237X256=60.627)
2. MOV TL0,#0FFH ; (60.627+255=60.927)
3. MOV TMOD,#01 ; *Timer* 0 Mode 16-bit
4. SETB TR0 ; Start *Timer* 0
5. Jbc TF0,\$; Loop sampai *overflow*

F. Bahasa Assembler

1. Definisi

Program dalam MCU (*Microcontroller Unit*) yang disimpan dalam PEROM atau EPROM adalah bahasa mesin, yaitu suatu kode-kode instruksi yang memerintahkan MCU untuk melakukan suatu pekerjaan tertentu. Kode-kode tersebut ditulis dalam sistem bilangan hexa desimal seperti contohnya : 74 02 75 F0 03 dan seterusnya, tetapi yang tersimpan dalam bentuk fisik di memori tersebut adalah kode “1” dan “0”, antara lain

contohnya sebagai berikut: 001110100b (74h), 00000010 (02h), 01110101b (75h), 11110000b (F0h) dan 00000011b (03h). Alangkah sulitnya mengerti atau membuat program bahasa mesin, karena hanya berupa kode-kode hexa dan untuk menerjemahkannya diperlukan “kamus” yang dapat menerjemahkan apa itu kode 74h dan seterusnya.

Guna mempermudah pemrograman, diciptakan penerjemah atau dalam istilah *software* disebut kompiler, yang mengonversi bahasa yang lebih tinggi, yang lebih dimengerti manusia. Setingkat lebih tinggi dari bahasa mesin adalah bahasa assembler, selanjutnya di atasnya adalah bahasa tingkat tinggi seperti visual basic, pascal, C dan sebagainya.

Program assembler adalah file text yang ditulis dengan menggunakan text editor, semacam Norton Commander, Sidekick, Notepad, QuickEdit, Ultra Edit dan sebagainya, selanjutnya kompiler akan menerjemahkan menjadi bahasa mesin, lengkap dengan perhitungan-perhitungan lompat relatif, lompat absolut dan sebagainya. Dalam dunia MCU, kompilernya dinamakan cross compiler.

2. Listing Program

Secara umum perintah dalam bahasa assembly disusun sebagai berikut:

```
[label:] mnemonic [operand1],operand2[,operand3][;komentar]
```

- **Label:** sifatnya opsional label digunakan sebagai penanda lokasi baris perintah. Label yang digunakan adalah kata apa saja terserah

kita. Label selalu diikuti oleh titik dua (:) dan tidak boleh diawali dengan angka.

- **Mnemonic:** adalah sebuah singkatan yang mudah diingat yang menunjukkan arti perintah tertentu.
- **Operand:** adalah parameter tambahan yang mengikuti mnemonic. Jumlahnya tergantung mnemonic-nya.
- **Komentar:** sifatnya juga opsional. Bagian ini tidak akan diproses oleh assembler, tetapi berguna untuk keperluan dokumentasi agar program menjadi lebih mudah dimengerti.

Keterangan :

Setiap perintah akan dilaksanakan secara urut dari atas ke bawah terkecuali ada perintah untuk melompat atau interupsi.

3. Blok Diagram Pemrograman

Dalam membuat program, kita membutuhkan susunan bentuk program yang urut sesuai dengan proses kerja alat tersebut, sehingga nantinya mudah dimengerti dan dipahami. Dan dalam *troubleshooting*-nya pun kita lebih mudah untuk mengatasinya.

Secara garis besar program assembler terdiri atas tiga blok, yaitu sebagai berikut:

- a. Pengaturan awal, atau sering disebut dengan inisialisasi.

Contoh: `mov a,#0FEH`

- b. Program utama.

Contoh: `loop: mov P1,a`

```
rr a
acall delay
sjmp loop
```

c. Bagian subprogram.

Contoh: *delay*:

```
mov R7,#10
L1: mov R6,#250
L2: mov R5,#250
L3: djnz R5,L3
    djnz R6,L2
    djnz R7,L1
ret
end
```

4. Subprogram

Adalah serangkaian baris perintah yang akan dilaksanakan jika dipanggil. Secara umum format penulisan subprogram adalah sebagai berikut:

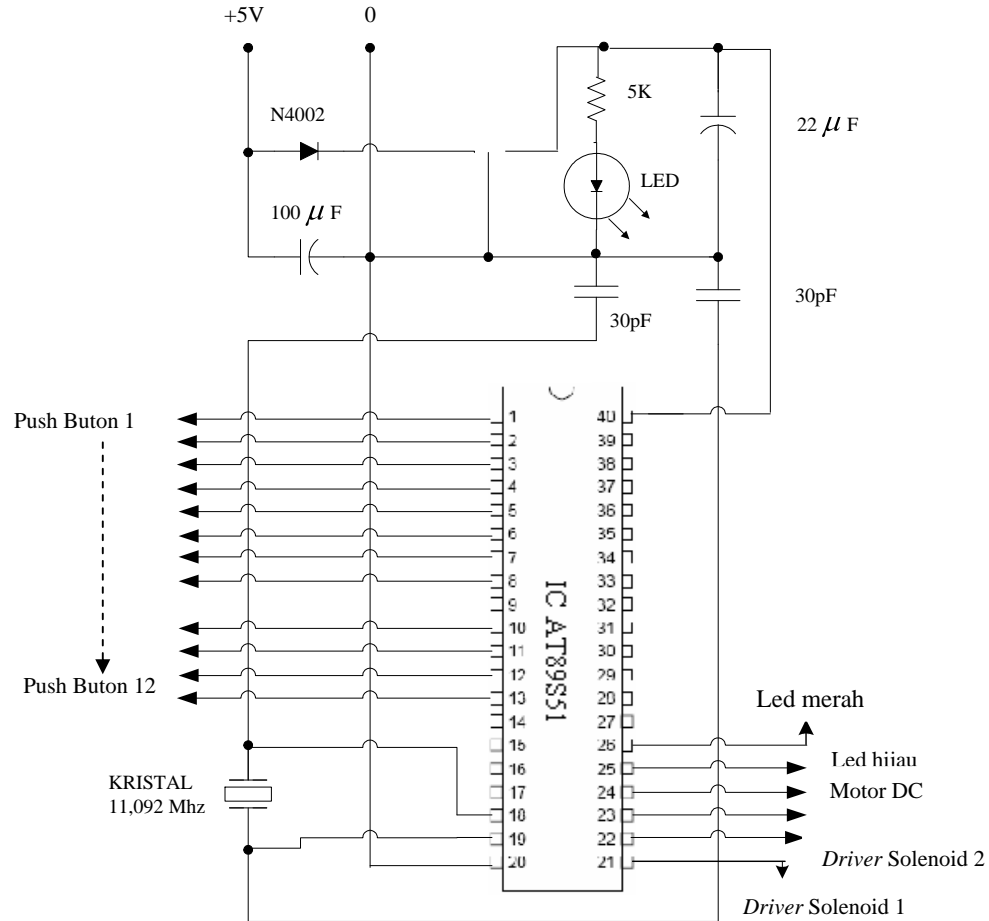
Nama:

Serangkaian baris perintah

Ret

G. Rangkaian Pendukung

1. Rangkaian Pengendali Mikrokontroler AT89S51



Gambar 9. Rangkaian pengendali Mikrokontroler AT89S51

Keterangan:

Minimum sistem AT89S51 adalah sebuah rangkaian pengendali dengan IC Mikrokontroler AT89S51 yang disamping berfungsi sebagai pengendali, juga berfungsi sebagai penyimpan program. Rangkaian ini membutuhkan tegangan 5Volt untuk bisa dioperasikan. Di samping itu di

dalam rangkaian ini juga terdapat kristal 11,092 Mhz, yang berfungsi untuk mengatur *clock* saat mengeksekusi program.

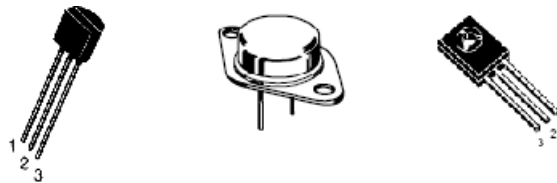
H. Komponen Pendukung

1. Transistor

Dalam elektronika, transistor adalah salah satu komponen aktif. Ada banyak jenis transistor yang ada di pasaran, salah satu jenis transistor tersebut adalah transistor bipolar. Transistor bipolar mempunyai 3 buah kaki yaitu kolektor (C), basis (B), dan emitor (E). Transistor jenis ini dibedakan menjadi 2 macam yaitu NPN dan PNP.



Gambar 10. *Simbol* Transistor NPN (a) dan Transistor PNP (b)



Gambar 11. Macam-macam bentuk fisik transistor

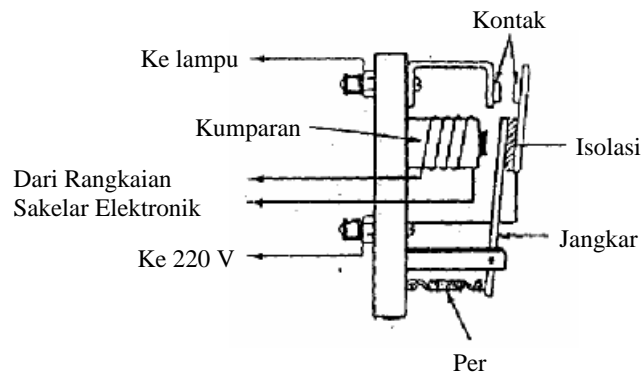
Salah satu fungsi transistor adalah sebagai sakelar elektronik. apabila Vcc diberi tegangan tetapi basis belum mendapat arus basis (I_b), maka arus dari kolektor (I_c) belum bisa mengalir ke emitor. Pada keadaan ini transistor diibaratkan seperti sakelar terbuka.

2. Relay

Relay adalah sebuah sakelar elektromagnetik yang apabila kumparan/*coil*-nya dialiri arus akan menimbulkan medan magnet pada kumparan tersebut yang berfungsi untuk membuka atau menutup satu atau beberapa kontak sakelar.



Gambar 12. Simbol dan Bentuk Fisik Relay Single Kontak



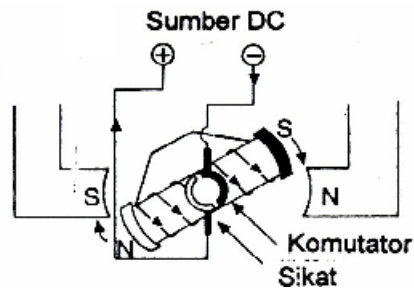
Gambar 13. Konstruksi Relay

3. Motor DC

Motor listrik adalah peralatan listrik yang mengubah dari besaran listrik ke mekanik. Koping antara sistem listrik dan sistem mekanik adalah melalui medium dari medan arus listrik atau medan muatan.

Prinsip dasar sebuah motor listrik adalah adanya dua buah medan magnet permanen dengan medan magnet buatan. Prinsip ini sesuai dengan kaidah tangan kiri *flemming* yang menyebutkan bahwa jika ada garis gaya

magnet yang menembus telapak tangan, dan arah arus searah dengan jari-jari tangan maka akan timbul gaya yang searah dengan ibu jari.



Gambar 14. Operasi Motor DC Magnet Permanen

4. Dioda

Dioda juga termasuk salah satu komponen aktif yang secara umum berfungsi sebagai penyearah. Dioda mempunyai 2 buah kutub yaitu kutub anoda (A) dan kutub katoda (K). Dioda akan menghantar arus listrik apabila mendapat forward bias yaitu anoda lebih positif daripada katoda. Tetapi dioda akan menahan arus listrik apabila mendapat reverse bias yaitu anoda lebih negatif dari katoda.



Gambar 15. Simbol Dioda & Bentuk Fisik Dioda

5. Resistor

Resistor/tahanan adalah termasuk salah satu komponen pasif. Resistor diberi simbol R dengan satuan O (ohm). Resistor berfungsi

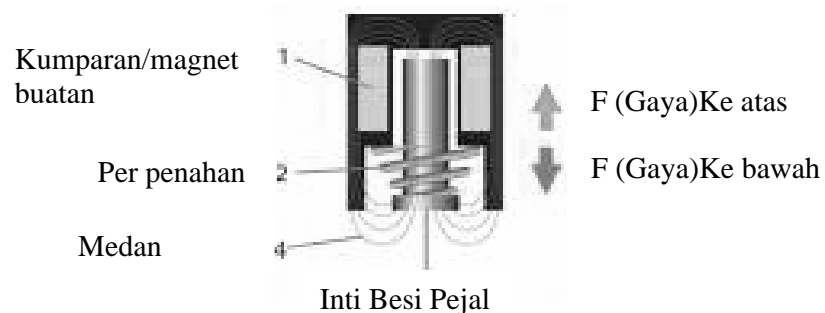
sebagai pengatur arus atau tegangan yang masuk/keluar suatu rangkaian. Untuk pengatur arus, resistor dihubungkan paralel sedangkan untuk pengatur tegangan, resistor dihubungkan seri.



Gambar 16. Simbol Resistor

6. Solenoid valve

Solenoid DC secara konstruktif selalu mempunyai inti yang pejal dan terbuat dari besi lunak. Dengan demikian mempunyai bentuk yang simpel dan kokoh. Selain itu maksudnya agar diperoleh konduktansi optimum pada medan magnet. Bila ada kelonggaran udara, tidak akan mengakibatkan kenaikan temperatur operasi, karena temperatur operasi hanya akan tergantung pada besarnya tahanan kumparan serta arus listrik yang mengalir.



Gambar 17. Solenoid valve

Ket:

F Ke atas : Gaya karena adanya pengaruh medan magnet

F Ke bawah : Gaya karena pengaruh medan magnet hilang

Bila solenoid DC diaktifkan (*switched on*) maka arus listrik yang mengalir meningkat secara perlahan. Ketika arus listrik dialirkan ke dalam kumparan akan terjadi elektromagnet. Selama terjadinya induksi akan menghasilkan gaya yang berlawanan dengan tegangan yang digunakan. Bila solenoid dipasifkan (*switched off*) maka medan magnet yang pernah terjadi akan hilang dan dapat mengakibatkan tegangan induksi yang besarnya bisa beberapa kali lipat dibandingkan dengan tegangan yang ada pada kumparan.

Tegangan induksi ini dapat mengakibatkan rusaknya isolasi pada gulungan koil, selanjutnya bila hal ini terjadi terus akan terjadi percikkan api. Untuk mengatasi hal ini maka harus dibuat rangkaian yang meredam percikkan api, misalnya dengan memasang tahanan yang dihubungkan secara paralel dengan induktansi. Sehingga bila terjadi pemutusan arus listrik, energi akan tersimpan dalam bentuk medan magnet dan dapat hilang lewat tahanan yang dipasang tadi.

Keuntungan Solenoid DC :

- a. Mudah pengoperasiannya
- b. Usianya lama
- c. Bunyi yang dihasilkan lemah
- d. Tenaga untuk mengoperasikan kecil

Kelemahan Solenoid DC :

- a. Perlu peredam percikkan api
- b. Terjadi tegangan tinggi saat pemutusan arus
- c. Perlu adaptor bila yang dipakai tegangan AC
- d. Bagian yang kontak cepat aus

BAB III

PROSES PERANCANGAN

A. Pembuatan Hardware

Pesawat simulasi digunakan sebagai aplikasi yang nyata dari sistem pengendalian yang berbentuk sebuah miniatur. Gambar perencanaan perlu dipersiapkan terlebih dahulu guna membantu proses pembuatan pesawat simulasi. Gambar konstruksi miniatur alat adalah sebagai berikut :



(a) Tampak Dalam

(b) Tampak luar

Gambar 18. Konstruksi miniatur alat pencampur

Dalam pembuatan pesawat simulasi sesuai di atas memerlukan beberapa alat dan bahan sebagai berikut :

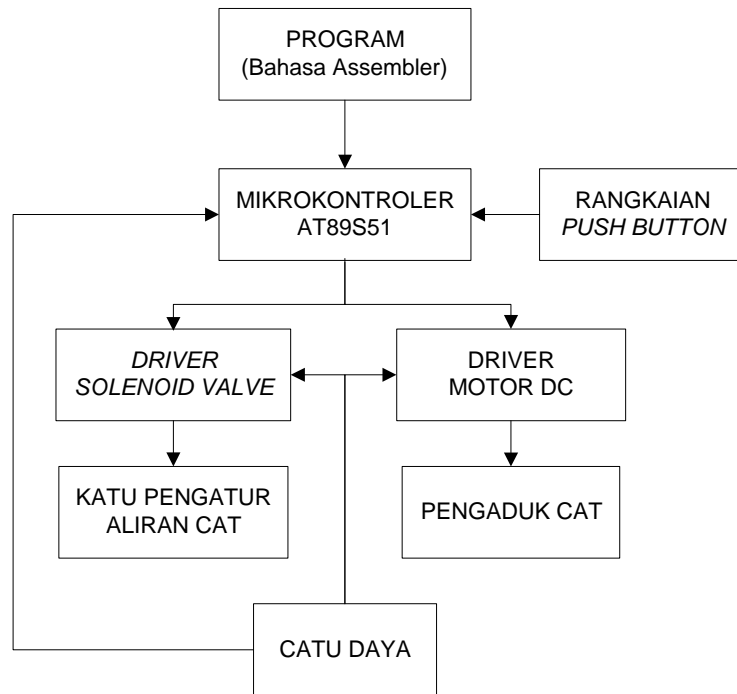
Alat			
No.	Nama	Jumlah	Merk
1	Palu	1 buah	(standar)
2	Gergaji Kayu	1 buah	(standar)
3	Gergaji Besi	1 buah	(standar)
4	Tang Kombinasi	1 buah	(standar)
5	Penggaris	1 buah	Butterfly
6	Bor Tangan	1 buah	Yama
7	Mata Bor ϕ 1mm	1 buah	(standar)
8	Mata Bor ϕ 3mm	1 buah	(standar)
9	Mata Bor ϕ 8 mm	1 buah	(standar)
10	Multimeter	1 buah	Sanwa
11	Solder 30 Watt	1 buah	Goat
12	Spidol permanen F	1 buah	Snowman
13	Kikir Bulat	1 buah	(standar)
14	Kikir Segitiga	1 buah	(standar)
15	Tenol	1 buah	pancing
16	Obeng + dan -	1 buah	(standar)
Bahan			
No.	Nama	Jumlah	Ukuran
1	IC Mikrokontroler AT89S51	1 buah	(standar)
2	<i>Solenoid valve</i> 24VDC	3 buah	$\frac{1}{4}$ Inchi
3	Motor DC 12 V	1 buah	(standar)
4	Kristal 11,092 MHz	1 buah	(standar)
5	Kapasitor keramik 30 μ F	2 buah	(standar)
6	Transistor	4 buah	C 945
7	Relay 12VDC/5A	4 buah	(standar)
8	Dioda 1N4002	6 buah	1 watt
9	Trafo King	1 buah	5 A
10	7824	1 buah	IC
11	7812	2 buah	IC
12	7805	1 buah	IC
13	Dioda Bridge	2 buah	2 A
14	Elco 2200uF	2 buah	16 V
15	Elco 220uF	3 buah	16 V
16	Capasitor 100nF	3 buah	100 V
17	PCB polos	1 buah	10 x 20 cm
18	Kabel pelangi	secukupnya	(standar)
19	Sakelar On-Off AC	1 buah	(standar)
20	Kabel Jack AC	1 buah	(standar)
21	<i>Push button</i>	12 buah	(standar)
22	LED merah	2 buah	(standar)
23	LED hijau	1 buah	(standar)
24	Tempat cat warna dasar	3 buah	disesuaikan
25	Tempat pengadukan cat	1 buah	disesuaikan
26	Selang <i>solenoid valve</i>	Secukupnya	(standar)

Tabel 5. Daftar alat dan bahan

B. Blok Diagram Alat dan Prinsip Kerja

I. Blok Diagram

Blok diagram ini merupakan gambaran dari sistem yang dibuat yang terdiri dari Program, Mikrokontroler AT89S51, rangkaian *Driver*, catu daya dan rangkaian penggerak lainnya.



Gambar 19. Blok Diagram Sistem Pengendali

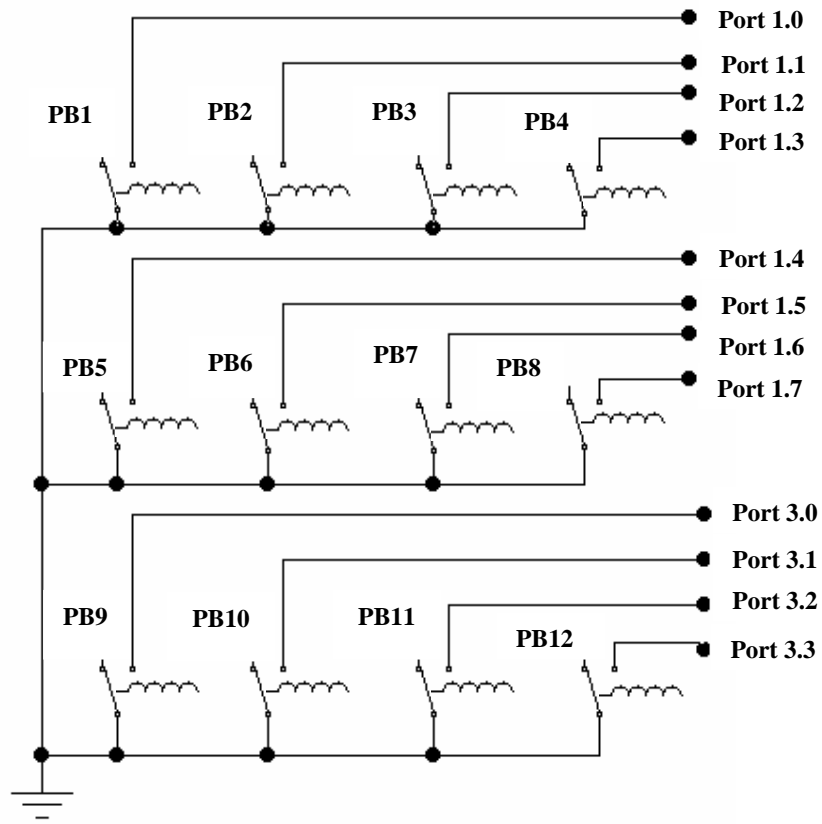
II. Prinsip kerja

Pembahasan cara kerja sistem pengoplosan cat secara otomatis berbasis Mikrokontroler AT89S51 ini mengacu pada blok diagram, yaitu sebagai berikut;

1. Bahasa pemrograman yang digunakan adalah Bahasa assembler. Bagian ini berfungsi untuk memberikan perintah-perintah dalam

bentuk bahasa program assembler ke dalam IC Mikrokontroler untuk mengendalikan sistem kerja proses pencampuran cat secara keseluruhan.

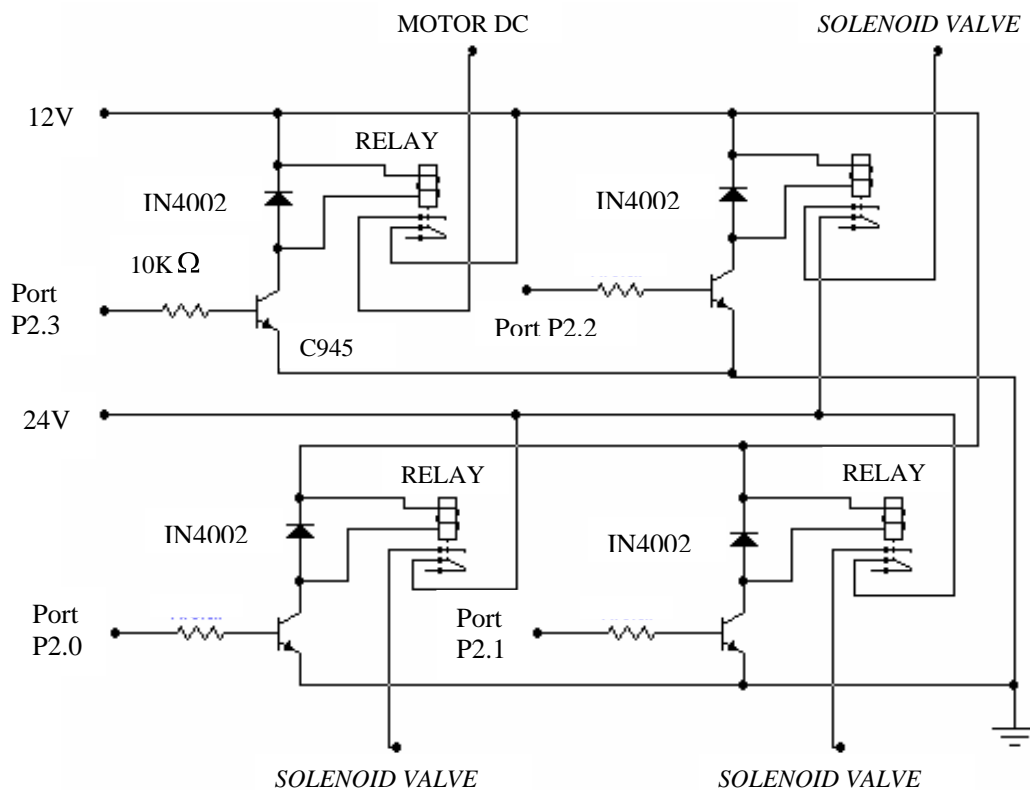
2. IC Mikrokontroler AT89S51 berfungsi sebagai pusat pengendali. Pada saat program pencampuran berlangsung, port 2.0 sampai dengan port 2.3 pada IC Mikrokontroler AT89S51 secara berurutan mengeluarkan tegangan sebesar ± 2 volt yang berfungsi untuk mengaktifkan rangkaian *driver solenoid valve* dan *driver motor DC* dengan cara membuka atau menutup *solenoid valve* serta menjalankan motor DC sebagai pengaduk hasil campuran cat yang sudah ditentukan sesuai dengan waktu tunda pada program.
3. Rangkaian *push button* terdiri dari konfigurasi 12 *push button* yang di paralel dengan *ground (0)*. *Push button* dihubungkan ke port 1 dan port 3 di rangkaian pengendali Mikrokontroler AT89S51 sebagai tombol untuk mengeksekusi program pencampuran warna cat. *Push button* di paralel dengan *ground* karena port 1 dan port 3 aktif tinggi (pada keadaan normal mengeluarkan sinyal 1 atau tegangan sebesar ± 5 volt) dengan kata lain Mikrokontroler akan mengeksekusi program apabila menerima sinyal 0, oleh karena itu pada program utama (lihat halaman 37) digunakan instruksi *jnb* (*jump if not bit*) yang artinya program akan melompat ke instruksi yang dituju jika bernilai 0.



Gambar 20. Rangkaian Konfigurasi *Push button*

4. Rangkaian *driver* berfungsi untuk menjalankan *solenoid valve* dan motor DC. Rangkaian *driver* ini terdiri dari sakelar Digital yang berfungsi untuk mengendalikan beban berupa solenoid dan motor DC. Rangkaian sakelar digital ini dibentuk oleh komponen resistor $10k\Omega$, transistor C945, dioda 1N4002, dan relay 12Vdc. Resistor pada kaki basis akan membatasi arus yang masuk ke basis transistor. Dioda 1N4002 berfungsi untuk menahan tegangan balik dari relay dari kondisi aktif ke kondisi tidak aktif. Transistor berfungsi sebagai sakelar untuk mengaktifkan/menonaktifkan relay. Saat basis transistor

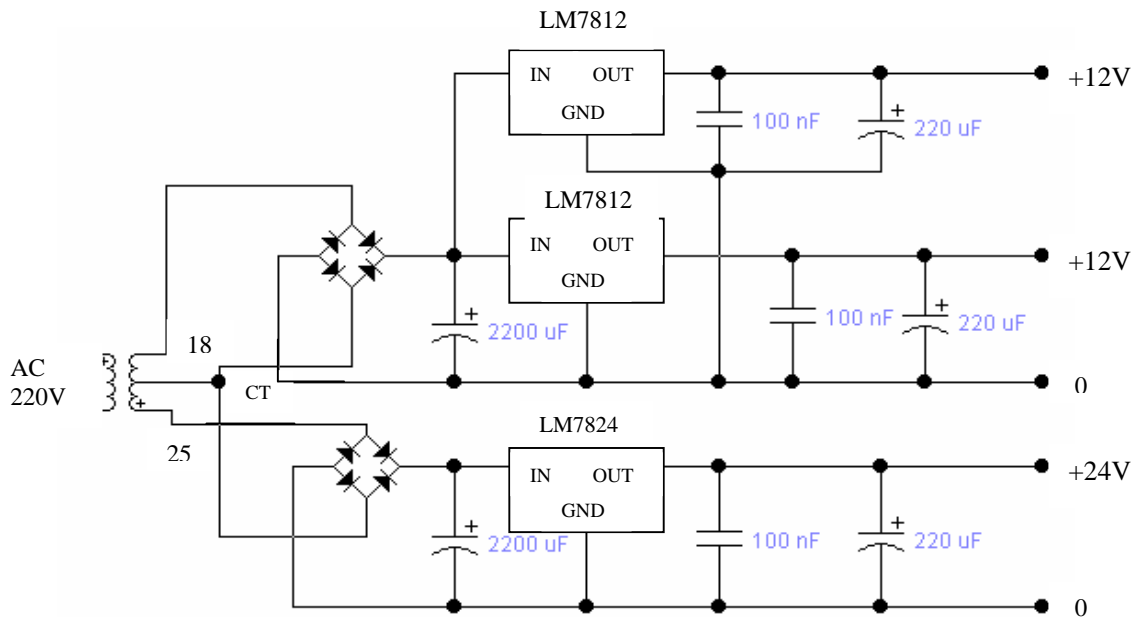
mendapat arus I_b maka arus I_c akan dapat mengalir ke emitor melalui koil relay dan relay dalam kondisi aktif sehingga beban pun akan hidup. Pada keadaan seperti itu, transistor diibaratkan seperti sakelar tertutup. Sedangkan saat basis transistor tidak mendapat arus I_b maka arus I_c tidak dapat mengalir ke emitor melalui koil relay dan relay dalam kondisi tidak aktif sehingga beban pun akan mati. Pada keadaan seperti itu, transistor diibaratkan seperti sakelar terbuka.



Gambar 21. Rangkaian *Driver*

5. Tegangan DC 18V dan 25V disearahkan oleh dioda bridge. Keluaran dari dioda bridge ini kemudian distabilkan oleh dua buah IC, yaitu LM7812 dan LM7824 yang menghasilkan tegangan DC sebesar +12V

dan +24V. Oleh karena tegangan yang diperlukan pada tiap rangkaian tidak sama, rangkaian catu daya ini mempunyai dua buah keluaran tegangan DC, yaitu 12V dan 24V yang berfungsi untuk memberi pasokan tegangan pada motor DC dan *solenoid valve*. Sedangkan untuk supply tegangan pada rangkaian pengendali Mikrokontroler digunakan 12V_{DC} yang kemudian distabilkan lagi oleh IC LM7805 menjadi 5V_{DC}. Kapasitor 100 nF berfungsi untuk membuang noise/ripple pada tegangan output DC.



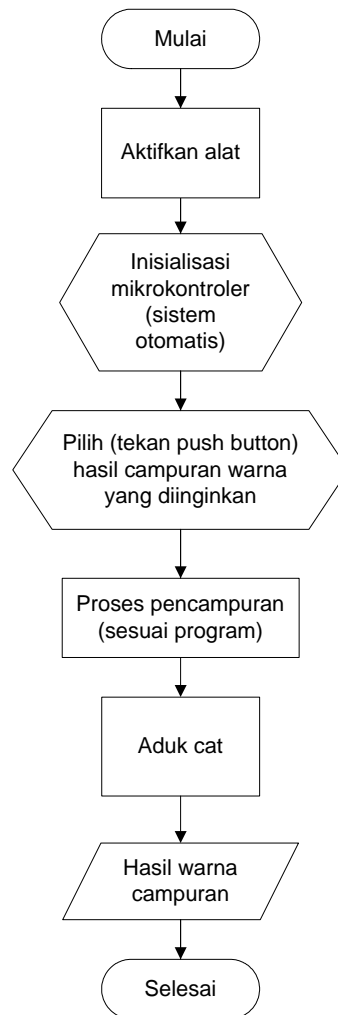
Gambar 22. Catu Daya 12&24V

6. Katup solenoid/*solenoid valve* adalah sebuah pengatur aliran cat berfungsi untuk mengatur besar aliran dari masing-masing cat (merah, biru, kuning). Katup pengatur ini bekerja sesuai program.

7. Pengaduk cat merupakan proses Finishing dari proses pengoplosan cat ini. Pengaduk ini akan mengaduk / mencampur ketiga macam warna cat dengan perbandingan tertentu yang sebelumnya telah diproses oleh katup pengatur aliran cat. Pengaduk cat ini digerakkan oleh motor DC, dengan tegangan sebesar 12 Volt.

C. Pembuatan Software

1. Pembuatan *Flowchart* Sistem

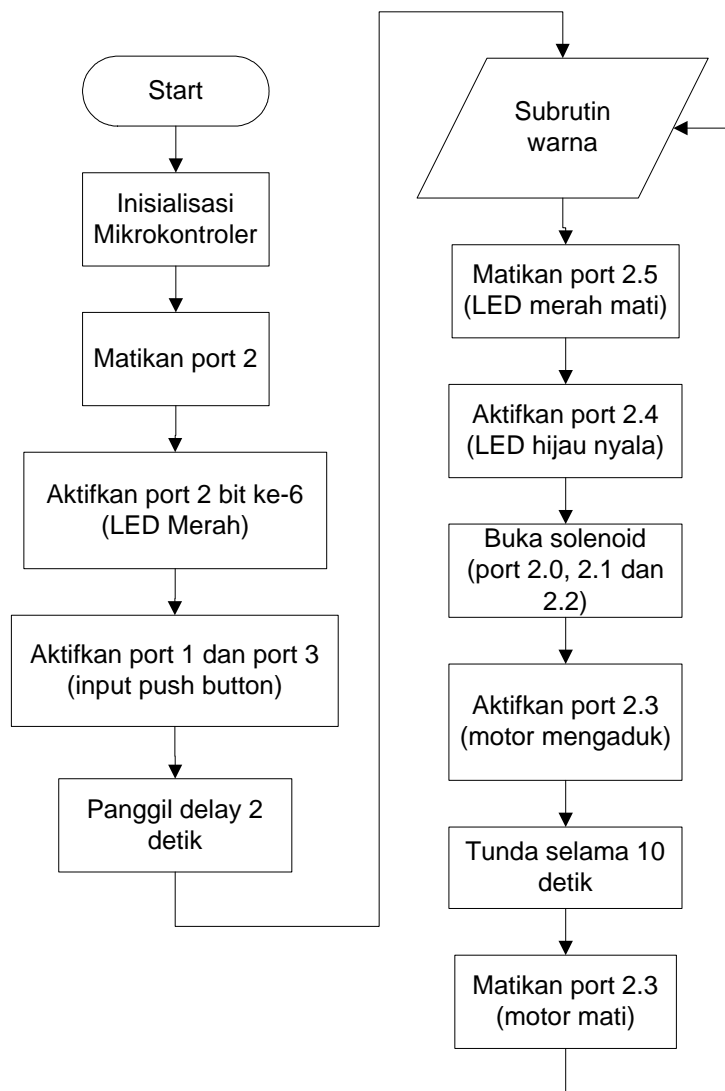


Gambar 23. *Flowchart* Sistem

Keterangan *Flowchart* :

Alat pencampur warna cat otomatis ini menggunakan sistem otomatis dari awal pencampuran sampai pengadukan. Sistem pencampuran ini dikendalikan oleh program Mikrokontroler AT89S51. Sebelum alat dijalankan, IC Mikrokontroler AT89S51 harus sudah diisi program hex pencampuran warna dan siap untuk dipasang pada rangkaian minimum sistem atau rangkaian pengendali Mikrokontroler AT89S51. Kemudian setelah alat aktif, maka secara otomatis Mikrokontroler akan melakukan proses inisialisasi program awal dan mengaktifkan *push button* 1 sampai dengan *push button* 12. *Push button* tersebut merupakan tombol utama untuk menjalankan program pencampuran warna cat, yang jika salah satu tombol tersebut ditekan, maka secara langsung akan mengeksekusi program dan mengaktifkan solenoid 1,2,3 dan motor pengaduk sesuai dengan waktu *delay* pengaktifannya masing-masing. Setelah warna tercampur maka secara langsung motor DC akan mengaduk hasil campuran tersebut dengan waktu *delay* mengaduk yang sama pada setiap program.

2. Pembuatan *Flowchart* Program



Gambar 24. *Flowchart* Program

3. Pembuatan Program

Di bawah ini akan dijelaskan beberapa bagian program yang penting yang digunakan sebagai kendali pencampur warna cat otomatis:

```
$MOD51
ORG 0000h
mov P2,#00h
mov P2,#20h
mov P1,#0FFh
mov P3,#0FFh
lcall Delay_2s
```

Keterangan:

Instruksi program ORG 0000h di atas merupakan inisialisasi awal pengalamatan Mikrokontroler. Selanjutnya `mov P2,#00h` adalah suatu instruksi untuk memerintahkan Mikrokontroler mengeluarkan sinyal '0' pada semua port 2, sedangkan `mov P2,#20h` merupakan instruksi untuk menyalakan lampu indikator merah sebagai tanda bahwa tidak ada proses pencampuran.

```
;+++++;
;=====Program Utama=====;
;+++++;
Start_Loop:
Jnb P1.0,vivid_red
Jnb P1.1,deco_red
Jnb P1.2,orange_papaya
Jnb P1.3,purple
Jnb P1.4,orange
Jnb P1.5,olive
Jnb P1.6,army_green
Jnb P1.7,pastel_green
Jnb P3.0,forest_green
Jnb P3.1,coffee_brown
Jnb P3.2,leaf_brown
Jnb P3.3,silver_brown
sjmp Start_Loop
```

```
vivid_red:
    lcall
    vivid_red_1
    ret
deco_red:
    lcall
    deco_red_1
    ret
orange_papaya:
    lcall
    orange_papaya_1
    ret
purple:
    lcall
    purple_1
    ret
orange:
    lcall
    orange_1
    ret
olive:
    lcall
    olive_1
    ret
army_green:
    lcall
    army_green_1
    ret
pastel_green:
    lcall
    pastel_green_1
    ret
forest_green:
    lcall
    forest_green_1
    ret
coffee_brown:
    lcall
    coffee_brown_1
    ret
leaf_brown:
    lcall
    leaf_brown_1
    ret
silver_brown:
    lcall
    silver_brown_1
    ret
```

Keterangan:

Instruksi program di atas adalah instruksi program utama untuk mengaktifkan sakelar *push button* yang nantinya digunakan sebagai tombol utama untuk memanggil instruksi pencampuran ketiga warna cat primer menjadi warna campuran yang diinginkan.

```
;=====Subroutine untuk menghasilkan warna vivid_red=====;
vivid_red_1:
    clr    P2.5        ;lampu indikator warna merah mati
    setb   P2.4        ;lampu indikator warna hijau menyala tanda
                    sedang mengisi
    setb   P2.0        ;katup cat merah aktif
    lcall  Delay_140s  ;tunggu selama 140 detik
    clr    P2.0        ;matikan katup cat merah
    setb   P2.1        ;katup cat biru aktif
    lcall  Delay_10s   ;tunggu selama 10 detik
    clr    P2.1        ;matikan katup cat biru
    setb   P2.3        ;motor berputar
    lcall  Delay_10s   ;tunggu selama 10 detik
    clr    P2.3        ;motor mati
    clr    P2.4        ;lampu indikator warna hijau mati
    setb   P2.5        ;lampu indikator warna merah nyala tanda selesai
                    mencampur
    ret
```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna vivid red ditekan. Sehingga akan dihasilkan warna vivid red.

```
;=====Subroutine untuk menghasilkan warna deco_red=====;
deco_red_1:
    clr    P2.5        ;lampu indikator warna merah mati
    setb   P2.4        ;lampu indikator warna hijau menyala tanda
                    sedang mengisi
    setb   P2.0        ;katup cat merah aktif
    lcall  Delay_70s   ;tunggu selama 70 detik
    clr    P2.0        ;matikan katup cat merah
    setb   P2.1        ;katup cat biru aktif
```

```

lcall Delay_10s ;tunggu selama 10 detik
clr P2.1 ;matikan katup cat biru
setb P2.3 ;motor berputar
lcall Delay_10s ;tunggu selama 10 detik
clr P2.3 ;motor mati
clr P2.4 ;lampu indikator warna hijau mati
setb P2.5 ;lampu indikator warna merah nyala tanda selesai
mencampur
ret

```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna deco red ditekan. Sehingga akan dihasilkan warna deco red.

```

;=====Subroutine untuk menghasilkan orange_papaya=====;
orange_papaya_1:
clr P2.5 ;lampu indikator warna merah mati
setb P2.4 ;lampu indikator warna hijau menyala tanda
sedang mengisi
setb P2.0 ;katup cat merah aktif
lcall Delay_20s ;tunggu selama 20 detik
clr P2.0 ;matikan katup cat merah
setb P2.2 ;katup cat kuning aktif
lcall Delay_80s ;tunggu selama 80 detik
clr P2.2 ;matikan katup cat kuning
setb P2.3 ;motor berputar
lcall Delay_10s ;tunggu selama 10 detik
clr P2.3 ;motor mati
clr P2.4 ;lampu indikator warna hijau mati
setb P2.5 ;lampu indikator warna merah nyala tanda selesai
mencampur
ret

```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna *orange papaya* ditekan. Sehingga akan dihasilkan warna *orange papaya*.

```

;=====Subroutine untuk menghasilkan warna purple=====;
purple_1:
    clr    P2.5        ;lampu indikator warna merah mati
    setb   P2.4        ;lampu indikator warna hijau menyala tanda
                    sedang mengisi
    setb   P2.0        ;katup cat merah aktif
    lcall  Delay_20s   ;tunda selama 20 detik
    clr    P2.0        ;matikan katup cat merah
    setb   P2.1        ;katup cat biru aktif
    lcall  Delay_80s   ;tunda selama 80 detik
    clr    P2.1        ;matikan katup cat biru
    setb   P2.3        ;motor berputar
    lcall  Delay_10s   ;tunggu selama 10 detik
    clr    P2.3        ;motor mati
    clr    P2.4        ;lampu indikator warna hijau mati
    setb   P2.5        ;lampu indikator warna merah nyala tanda selesai
                    mencampur

    ret

```

Keterangan:

Sederetan instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna *purple* ditekan. Sehingga akan dihasilkan warna *purple*.

```

;=====Subroutine untuk menghasilkan warna orange=====;
orange_1:
    clr    P2.5        ;lampu indikator warna merah mati
    setb   P2.4        ;lampu indikator warna hijau menyala tanda
                    sedang mengisi
    setb   P2.0        ;katup cat merah aktif
    lcall  Delay_10s   ;tunda selama 10 detik
    clr    P2.0        ;matikan katup cat merah
    setb   P2.2        ;katup cat kuning aktif
    lcall  Delay_100s  ;tunda selama 100 detik
    clr    P2.2        ;matikan katup cat kuning
    setb   P2.3        ;motor berputar
    lcall  Delay_10s   ;tunggu selama 10 detik
    clr    P2.3        ;motor mati
    clr    P2.4        ;lampu indikator warna hijau mati
    setb   P2.5        ;lampu indikator warna merah nyala tanda selesai
                    mencampur

    ret

```

Keterangan:

Sederetan instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna *orange* ditekan. Sehingga akan dihasilkan warna *orange*.

```
;=====Subroutine untuk menghasilkan warna olive=====;
olive_1:
  clr   P2.5           ;lampu indikator warna merah mati
  setb  P2.4           ;lampu indikator warna hijau menyala tanda
                        sedang mengisi
  setb  P2.0           ;katup cat merah aktif
  lcall Delay_10s      ;tunda selama 10 detik
  clr   P2.0           ;matikan katup cat merah
  setb  P2.1           ;katup cat biru aktif
  lcall Delay_30s      ;tunda selama 30 detik
  clr   P2.1           ;matikan katup cat biru
  setb  P2.2           ;katup cat kuning aktif
  lcall Delay_100s     ;tunda selama 100 detik
  clr   P2.2           ;matikan katup cat kuning
  setb  P2.3           ;motor berputar
  lcall Delay_10s      ;tunggu selama 10 detik
  clr   P2.3           ;motor mati
  clr   P2.4           ;lampu indikator warna hijau mati
  setb  P2.5           ;lampu indikator warna merah nyala tanda selesai
                        mencampur
  ret
```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna *olive* ditekan. Sehingga akan dihasilkan warna *olive*.

```
;=====Subroutine untuk menghasilkan warna army_green=====;
army_green_1:
  clr   P2.5           ;lampu indikator warna merah mati
  setb  p2.4           ;lampu indikator warna hijau menyala tanda
                        sedang mengisi
```



```

setb P2.0      ;katup cat merah aktif
lcall Delay_10s ;tunda selama 10 detik
clr P2.0      ;matikan katup cat merah
setb P2.1      ;katup cat biru aktif
lcall Delay_30s ;tunda selama 30 detik
clr P2.1      ;matikan katup cat biru
setb P2.2      ;katup cat kuning aktif
lcall Delay_70s ;tunda selama 70 detik
clr P2.2      ;matikan katup cat kuning
setb P2.3      ;motor berputar
lcall Delay_10s ;tunggu selama 10 detik
clr P2.3      ;motor mati
clr P2.4      ;lampu indikator warna hijau mati
setb P2.5      ;lampu indikator warna merah nyala tanda selesai
                    mencampur

ret

```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna army green ditekan. Sehingga akan dihasilkan warna army green.

```

;=====Subroutine untuk menghasilkan warna pastel_green=====;
pastel_green_1:
clr P2.5      ;lampu indikator warna merah mati
setb P2.4      ;lampu indikator warna hijau menyala tanda
                    sedang mengisi

setb P2.1      ;katup cat biru aktif
lcall Delay_10s ;tunda selama 10 detik
clr P2.1      ;matikan katup cat biru
setb P2.2      ;katup cat kuning aktif
lcall Delay_80s ;tunda selama 80 detik
clr P2.2      ;matikan katup cat kuning
setb P2.3      ;motor berputar
lcall Delay_10s ;tunggu selama 10 detik
clr P2.3      ;motor mati
clr P2.4      ;lampu indikator warna hijau mati
setb P2.5      ;lampu indikator warna merah nyala tanda selesai
                    mencampur

ret

```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna pastel green ditekan. Sehingga akan dihasilkan warna pastel green.

```
;=====Subroutine untuk menghasilkan warna forest_green=====;
forest_green_1:
    clr    P2.5        ;lampu indikator warna merah mati
    setb   P2.4        ;lampu indikator warna hijau menyala tanda
                    sedang mengisi
    setb   P2.1        ;katup cat biru aktif
    lcall  Delay_40s   ;tunda selama 40 detik
    clr    P2.1        ;matikan katup cat biru
    setb   P2.2        ;katup cat kuning aktif
    lcall  Delay_60s   ;tunda selama 60 detik
    clr    P2.2        ;matikan katup cat kuning
    setb   P2.3        ;motor berputar
    lcall  Delay_10s   ;tunggu selama 10 detik
    clr    P2.3        ;motor mati
    clr    P2.4        ;lampu indikator warna hijau mati
    setb   P2.5        ;lampu indikator warna merah nyala tanda selesai
                    mencampur
    ret
```

Keterangan:

Instruksi program di atas merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna forest green ditekan. Sehingga akan dihasilkan warna forest green.

```
;=====Subroutine untuk menghasilkan warna coffee_brown=====;
coffee_brown_1:
    clr    P2.5        ;lampu indikator warna merah mati
    setb   P2.4        ;lampu indikator warna hijau menyala tanda
                    sedang mengisi
    setb   P2.0        ;katup cat merah aktif
    lcall  Delay_20s   ;tunda selama 20 detik
    clr    P2.0        ;matikan katup cat merah
    setb   P2.1        ;katup cat biru aktif
    lcall  Delay_40s   ;tunda selama 40 detik
```

```

clr    P2.1      ;matikan katup cat biru
setb   P2.2      ;katup cat kuning aktif
lcall  Delay_20s ;tunda selama 20 detik
clr    P2.2      ;matikan katup cat kuning
setb   P2.3      ;motor berputar
lcall  Delay_10s ;tunggu selama 10 detik
clr    P2.3      ;motor mati
clr    P2.4      ;lampu indikator warna hijau mati
setb   P2.5      ;lampu indikator warna merah nyala tanda selesai
                          mencampur

ret

```

Keterangan:

Instruksi program di atas adalah *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna coffee brown ditekan. Sehingga akan dihasilkan warna coffee brown.

```

;=====Subroutine untuk menghasilkan warna leaf_brown=====;
leaf_brown_1:
clr    P2.5      ;lampu indikator warna merah mati
setb   P2.4      ;lampu indikator warna hijau menyala tanda
                          sedang mengisi
setb   P2.0      ;katup cat merah aktif
lcall  Delay_20s ;tunda selama 20 detik
clr    P2.0      ;matikan katup cat merah
setb   P2.1      ;katup cat biru aktif
lcall  Delay_10s ;tunda selama 10 detik
clr    P2.1      ;matikan katup cat biru
setb   P2.2      ;katup cat kuning aktif
lcall  Delay_60s ;tunda selama 60 detik
clr    P2.2      ;matikan katup cat kuning
setb   P2.3      ;motor berputar
lcall  Delay_10s ;tunggu selama 10 detik
clr    P2.3      ;motor mati
clr    P2.4      ;lampu indikator warna hijau mati
setb   P2.5      ;lampu indikator warna merah nyala tanda selesai
                          mencampur

ret

```

Keterangan:

Instruksi program di atas adalah merupakan *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna leaf brown ditekan. Sehingga akan dihasilkan warna leaf brown.

```
;=====Subroutine untuk menghasilkan warna silver_brown=====;
silver_brown_1:
    clr    P2.5          ;lampu indikator warna merah mati
    setb   P2.4          ;lampu indikator warna hijau menyala tanda
                        sedang mengisi
    setb   P2.0          ;katup cat merah aktif
    lcall  Delay_40s     ;tunda selama 40 detik
    clr    P2.0          ;matikan katup cat merah
    setb   P2.1          ;katup cat biru aktif
    lcall  Delay_20s     ;tunda selama 20 detik
    clr    P2.1          ;matikan katup cat biru
    setb   P2.2          ;katup cat kuning aktif
    lcall  Delay_60s     ;tunda selama 60 detik
    clr    P2.2          ;matikan katup cat kuning
    setb   P2.3          ;motor berputar
    lcall  Delay_10s     ;tunggu selama 10 detik
    clr    P2.3          ;motor mati
    clr    P2.4          ;lampu indikator warna hijau mati
    setb   P2.5          ;lampu indikator warna merah nyala tanda selesai
                        mencampur
    ret
```

Keterangan:

Instruksi program di atas adalah *subroutine* program atau inti dari program utama yang akan tereksekusi apabila *push button* warna silver brown ditekan. Sehingga akan dihasilkan warna silver brown.

```

;=====subrutin tunda 5 milidetik=====;
Delay_5mS:  Push    TMOD
            Mov     TMOD,#01H
            Mov     TH0,#0EDH
            Mov     TL0,#0FFH
            Setb   TR0

tunggu_5mS:
            Jbc    TF0,sudah_5mS
            Ajmp   tunggu_5mS

sudah_5mS:  Clr     TR0
            Pop    TMOD
            Ret

;=====subrutin tunda 1 detik=====;
Delay_1s:   Mov     r0,#0200

Tunggu_1detik:
            Lcall  Delay_5mS
            Djnz  r0,Tunggu_1detik
            Ret

;=====subrutin tunda 2 detik=====;
Delay_2s:   lcall  Delay_1s
            lcall  Delay_1s
            ret

;=====subrutin tunda 5 detik=====;
Delay_5s:   lcall  Delay_1s
            lcall  Delay_1s
            lcall  Delay_1s
            lcall  Delay_1s
            lcall  Delay_1s
            ret

;=====subrutin tunda 10 detik=====;
Delay_10s:  lcall  Delay_5s
            lcall  Delay_5s
            ret

;=====subrutin tunda 20 detik=====;
Delay_20s:  lcall  Delay_10s
            lcall  Delay_10s
            ret

;=====subrutin tunda 30 detik=====;
Delay_30s:  lcall  Delay_10s
            lcall  Delay_10s
            lcall  Delay_10s
            ret

```

```

;=====subrutin tunda 40 detik=====;
Delay_40s:
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    ret
;=====subrutin tunda 60 detik=====;
Delay_60s:
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    ret
;=====subrutin tunda 70 detik=====;
Delay_70s:
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    ret
;=====subrutin tunda 80 detik=====;
Delay_80s:
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    ret
;=====subrutin tunda 100 detik=====;
Delay_100s:
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s
    lcall    Delay_10s

```

```

ret
;=====subrutin tunda 140 detik=====;
Delay_140s:
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
lcall    Delay_10s
ret

```

Keterangan:

Instruksi di atas merupakan program subrutin *delay* untuk mengatur berapa nilai *timer*/delay yang akan dipakai untuk membuka katup solenoid sesuai dengan jumlah takaran cat yang sudah ditentukan, dan juga untuk mengatur lamanya motor berputar sebagai pengaduk, yaitu dengan mengubah nilai pada TL0 dan TH0 (lihat halaman 46).

end

Keterangan:

Instruksi end di atas merupakan instruksi untuk mengakhiri program secara keseluruhan.

BAB IV

PENGUJIAN DAN PROSES PENGOPERASIAN

A. Pembuatan dan Pengujian Campuran Warna

Pembuatan dan pengujian campuran warna ini dilakukan untuk mendapatkan prosentase perbandingan warna yang tepat untuk bisa menghasilkan hasil campuran warna baru. Dalam pengujian ini digunakan cat air atau cat poster yang banyak ditemui ditoko-toko alat tulis. Dalam Pembuatan dan pengujian ini perancang menggunakan cat dengan merk “ASTURO” dan untuk mencari acuan hasil campuran warna perancang menggunakan daftar warna cat “EMCO” yang merupakan merk dagang dari perusahaan cat MATARAM PAINT Co.Ltd, Surabaya, Indonesia.

1. Tabel simulasi perbandingan warna primer untuk bisa menghasilkan warna campuran. Misalkan satuan yang digunakan adalah mililiter (*ml*).

No.	Warna Hasil Campuran	Warna primer (<i>ml</i>)			Ket
		Merah	Biru	Kuning	
1	Vivid red	14 <i>ml</i>	1 <i>ml</i>	-	Berhasil
2	Deco red	7 <i>ml</i>	1 <i>ml</i>	-	Berhasil
3	Orange papaya	1 <i>ml</i>	-	4 <i>ml</i>	Berhasil
4	Purple	1 <i>ml</i>	4 <i>ml</i>	-	Berhasil
5	Orange	1 <i>ml</i>	-	10 <i>ml</i>	Berhasil
6	Olive	1 <i>ml</i>	3 <i>ml</i>	10 <i>ml</i>	Berhasil
7	Army green	1 <i>ml</i>	3 <i>ml</i>	7 <i>ml</i>	Berhasil
8	Pastel green	-	1 <i>ml</i>	8 <i>ml</i>	Berhasil
9	Forest green	-	2 <i>ml</i>	3 <i>ml</i>	Berhasil
10	Coffee brown	1 <i>ml</i>	2 <i>ml</i>	1 <i>ml</i>	Berhasil
11	Leaf brown	2 <i>ml</i>	1 <i>ml</i>	6 <i>ml</i>	Berhasil
12	Silver brown	2 <i>ml</i>	1 <i>ml</i>	3 <i>ml</i>	Berhasil

Tabel 6. Simulasi Pencampuran Warna

2. Tabel waktu *delay* untuk masing-masing warna primer dalam proses pencampuran. Berdasarkan percobaan telah ditentukan untuk takaran adalah kelipatan $50\text{ ml} = 20$ detik.

No.	Warna Hasil Campuran	Warna primer (<i>ml</i>)						Ket
		Merah		Biru		Kuning		
		<i>ml</i>	Detik	<i>ml</i>	detik	<i>ml</i>	detik	
1	Vivid red	350	140	25	10	-	-	Berhasil
2	Deco red	175	70	25	10	-	-	Berhasil
3	Orange papaya	50	20	-	-	200	80	Berhasil
4	Purple	50	20	200	80	-	-	Berhasil
5	Orange	25	10	-	-	250	100	Berhasil
6	Olive	25	10	75	30	250	100	Berhasil
7	Army green	25	10	75	30	175	70	Berhasil
8	Pastel green	-	-	25	10	200	80	Berhasil
9	Forest green	-	-	100	40	150	60	Berhasil
10	Coffee brown	150	20	300	40	150	20	Berhasil
11	Leaf brown	50	20	25	10	150	60	Berhasil
12	Silver brown	100	40	50	20	150	60	Berhasil

Tabel 7. Waktu *Delay* Pencampuran

B. Hasil dan Cara Pengoperasian

I. Hasil Warna

Dari hasil pembuatan dan pengujian diatas, telah diperoleh hasil campuran warna baru sebanyak 12 macam, dengan rincian sebagai berikut:

NO	Nama warna	Hasil dari campuran warna primer	Hasil campuran
1	Vivid red	Merah dan biru	
2	Deco Red	Merah dan Biru	

3	Orange Papaya	Merah dan Kuning	
4	Purple	Merah dan Biru	
5	Orange	Merah dan Biru	
6	Olive	Merah, Biru dan kuning	
7	Army Green	Merah, Biru dan kuning	
8	Pastel Green	Biru dan kuning	
9	Forest Green	Biru dan kuning	

10	Coffee Brown	Merah, Biru dan kuning	
11	Leaf Brown	Merah, Biru dan kuning	
12	Silver Brown	Merah, Biru dan kuning	

Tabel 8. Hasil Warna Campuran

II. Cara Pengoperasian

Cara pengoperasian alat miniatur pencampur warna cat secara otomatis berbasis Mikrokontroler AT89S51 ini adalah :

- a. Persiapkan kelengkapan alat, yaitu seperangkat miniatur alat pencampur dan *hardware* yang terdiri dari rangkaian rangkaian catu daya, rangkaian *driver* solenoid dan motor, cat dengan warna primer yang sudah dikemas menjadi sebuah pesawat simulasi.
- b. Cek kabel catu daya yang menyuplai tegangan pada pesawat simulasi.
- c. Aktifkan alat miniatur tersebut dengan menghubungkan kabel catu daya ke tegangan $220V_{AC}$, kemudian tekan sakelar *power* ke posisi ON.
- d. Untuk menjalankan proses pencampuran, pilihlah dan tekan di antara ke-12 *push button* sesuai dengan macam warna yang tertera di pesawat

simulasi, pada keadaan ini lampu indikator warna merah menyala, menandakan tidak ada proses pencampuran yang sedang berlangsung.

- e. Apabila program berhasil dijalankan, maka secara otomatis ke-3 *solenoid valve* akan aktif dan mencampurkan ketiga macam warna cat primer atau sesuai dengan campuran hasil warna yang dihasilkan, kemudian motor DC akan berputar mengaduk hasil campuran tersebut. Pada keadaan ini lampu indikator warna hijau menyala menandakan ada proses pencampuran yang sedang berlangsung.
- f. Setelah proses pencampuran selesai, lepas tampungan tempat pengadukan untuk mengambil hasil campuran akhir.

BAB V

PENUTUP

A. Simpulan

Berdasarkan dari uraian di atas, dapat diambil beberapa kesimpulan yang penting yaitu sebagai berikut:

1. Pembukaan masing-masing katup *solenoid valve* pada proses pencampuran dioperasikan secara bertahap/berurutan.
2. Alat miniatur ini berhasil mencampurkan ketiga warna dasar cat secara otomatis sesuai dengan takaran yang sudah ditentukan tapi masih diperlukan pengembangan lebih jauh mengenai penyesuaian kekentalan cat, tekanan udara, dan faktor-faktor lain yang bisa mempengaruhi hasil campuran akhir.
3. Rangkaian pengendali dengan IC Mikrokontroler AT89S51 terbukti bisa melakukan pencampuran warna dengan menggunakan *solenoid valve* $\frac{1}{4}$ Inchi, sehingga dihasilkan tabel 6, tabel 7 dan tabel 8 sebagai hasil pencampuran untuk bukaan masing-masing valve.

B. Saran

Beberapa saran yang dapat digunakan untuk pengembangan sistem pengendali ini adalah sebagai berikut:

1. Apabila diinginkan untuk mencampur cat dengan bahan dasar minyak, lebih baik menggunakan *solenoid valve* yang berukuran $\frac{3}{4}$ Inchi karena penampang katup yang lebih besar akan mempermudah aliran catnya.

2. Perlu diteliti lebih jauh bagaimana pengaruh besar volume dan tekanan penampung warna cat dasar serta kekentalan masing-masing cat dasar terhadap campuran warna cat yang dihasilkan, sehingga bisa didapatkan hasil campuran yang sempurna.
3. Agar tampilan lebih menarik dan menghemat penempatan *push button* warna, disarankan untuk menggunakan keypad matriks berukuran 3×4 , dengan menggunakan tampilan LCD.

DAFTAR PUSTAKA

- Affendi, Yusuf. 1978. *Disain Warna, Susunan dan Fungsinya*. Bandung: Proyek Pengemb. Ilmu dan Teknologi ITB Bandung
- Christanto Danny. 2004. *Panduan Praktikum Dasar Mikrokontroller Keluarga MCS-51 Menggunakan DT-51 Minimum System Ver 3.0 dan DT-51 Trainer Board*. Surabaya: Innovative Electronics.
- Putra Eko Agfianto. 2004. *Belajar Mikrokontroler AT89S51/51/52/55 (Teori dan Aplikasi)*. Yogyakarta : Gava Media.
- Setiawan Sulhan. 2006. *Mudah dan Menyenangkan Belajar Mikrokontroler*. Yogyakarta : Andi.
- Sudjadi. 2005. *Teori dan Aplikasi Mikrokontroler AT89C51*. Yogyakarta : Graha Ilmu.
- Suhata, ST. 2005. *Aplikasi Mikrokontroler sebagai Pengendali Peralatan Elektronik*. Jakarta: Elex Media Komputindo.
- Sunaryo Joko. 2006. *Alat pengoplos Cat Berbasis Personal Komputer*. Semarang: Universitas Negeri Semarang
- Tokheim, Roger. 1996. *Prinsip-Prinsip Digital Elektronika*. Jakarta : Erlangga.