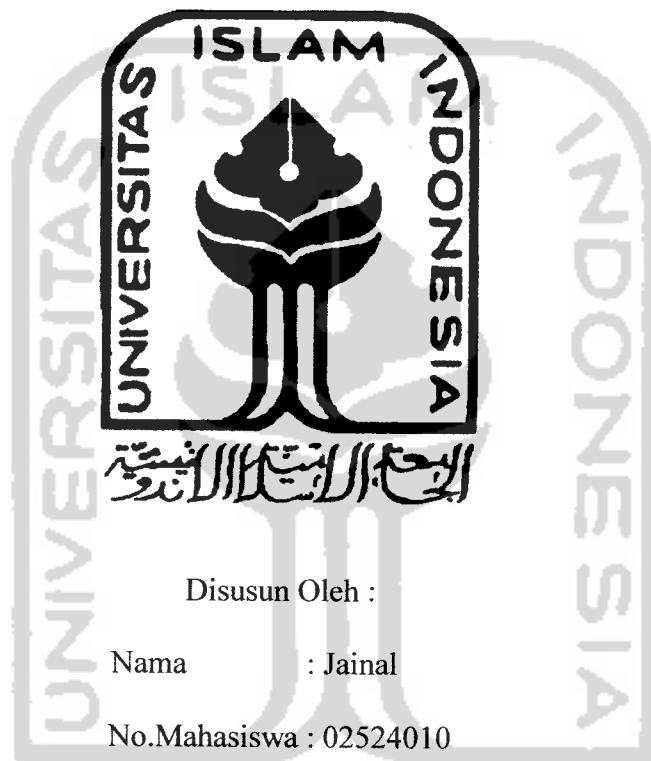


**MOBIL ROBOT BERLENGAN DIKONTROL
LEWAT PORT PARALEL**

TUGAS AKHIR

Diajukan Sebagai Salah Satu Syarat Untuk Memperoleh Gelar Sarjana Strata 1

Jurusan Teknik Elektro



Disusun Oleh :

Nama : Jainal

No.Mahasiswa : 02524010

**JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA
JOGJAKARTA**

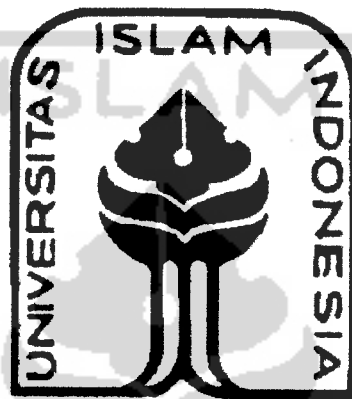
2007

LEMBAR PENGESAHAN PEMBIMBING

MOBIL ROBOT BERLENGAN DIKONTROL LEWAT

PORT PRINTER

TUGAS AKHIR



Disusun Oleh:

Nama : Jainal

No.Mahasiswa : 02524010

Jogjakarta , 10 Mei 2007

Pembimbing I,

Hj. Budi Astuti, Ir, MT

Pembimbing II,

Dwi Ana Ratnawati, ST

LEMBAR PENGESAHAN PENGUJI

MOBIL ROBOT BERLENGAN DIKONTROL LEWAT PORT

**PARALEL
TUGAS AKHIR**

Oleh :

Nama : jainal

No.Mahasiswa : 02524010

**Telah Dipertahankan di Depan Sidang Penguji sebagai Salah Satu Syarat untuk
Memperoleh Gelar Sarjana Teknik Elektro Fakultas Teknologi Industri**

Universitas Islam Indonesia

Yogyakarta, 30 Mei 2007

Tim Penguji,

Hj.Budi Astuti, Ir, MT

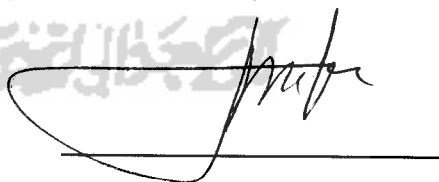
Ketua

Dwi Ana Ratnawati, ST

Anggota I

Hendra Setiawan, ST, MT

Anggota II



Mengetahui,
Ketua Jurusan Teknik Elektro
Universitas Islam Indonesia

Yuwono, ST, Msc

HALAMAN PERSEMBAHAN

Kupersembahkan tugas akhir ini kepada

Bapak dan ibu tersayang

Kakakku yang tersayang

Teman–temanku penghuni Asrama Bontang terima kasih atas bantuan kalian

Pengurus Himpunan Mahasiswa Bontang Jogjakarta

Dan semua yang tidak bisa kusebutkan namanya

Terima kasih atas doa dan bantuannya



MOTTO

Apapun yang dapat dilihat dengan mata,

Sesulit apapun kita pasti dapat mengerjakannya,

Dan berani memulai adalah suatu kemenangan

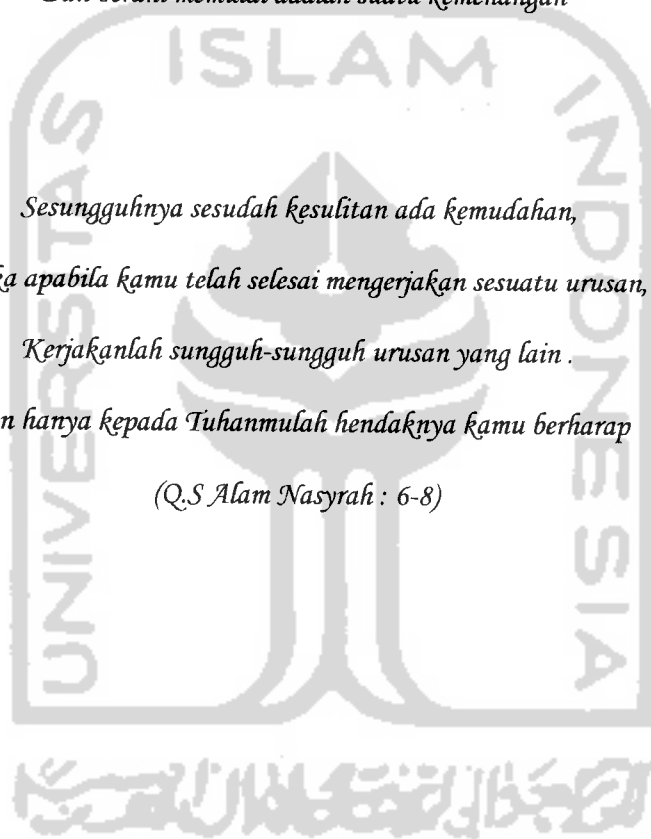
Sesungguhnya sesudah kesulitan ada kemudahan,

Maka apabila kamu telah selesai mengerjakan sesuatu urusan,

Kerjakanlah sungguh-sungguh urusan yang lain .

Dan hanya kepada Tuhanmulah hendaknya kamu berharap

(Q.S Alam Nasyrak : 6-8)



KATA PENGANTAR

Puji syukur penulis panjatkan kehadirat Allah SWT dan Sholawat serta salam teruntuk Nabi Muhammad SAW yang telah memberikan rahmat dan hidayah kepada penulis sehingga dapat menyelesaikan tugas akhir ini dengan judul “Mobil Robot Berlengan Dikontrol Lewat Port Printer”

Tujuan penulisan tugasakhir ini adalah sebagai syarat kelulusan pada program strata 1 (S1) Jurusan Teknik Elektro Fakultas Teknologi Industri pada Universitas Islam Indonesia Jogjakarta.

Dengan selesainya tugas akhir ini penulis menyampaikan ucapan terimakasih dan penghargaan sebesar-besarnya kepada :

1. Bapak Fathul Wahid, ST, Msc selaku dekan Fakultas Teknologi Industri Universitas Islam Indonesia Jogjakarta.
2. Bapak Tito Yuwono, ST, Msc selaku kepala Jurusan Teknik Elektro Universitas Islam Indonesia.
3. Ibu Hj.Budi Astuti, Ir, MT selaku dosen pembimbing I.
4. Ibu Dwi Ana Ratnawati, ST selaku dosen pembimbing II.
5. Bapak dan Ibu dosen Teknik Elektro Universitas Islam Indonesia Jogjakarta yang telah mengajarkan dan memberikan ilmu dibidang Elektronika.
6. Kedua orang tuaku yang telah memberikan doa, membiayai kuliah dan sabar menunggu hingga dapat menyelesaikan pendidikan Strata 1.
7. Kakakku yang selalu memberikan motivasi.
8. Teman-teman seperjuangan di asrama Bontang yang selalu mendukung susah dan senang.
9. Teman-teman elektro angkatan 2002.

Bagaimanapun juga penulis adalah seorang manusia biasa dan penuh dengan kekurangan. Hanyalah terima kasih yang dapat kami berikan apabila pembaca turut memberikan saran dan kritik agar tugas akhir ini selalu mendekati sempurna. Tugas akhir ini akan sangat berarti jika dapat memberikan manfaat yang tiada pernah habis bagi penulis khususnya dan bagi semua pihak yang bersangkutan pada umumnya.

Wassalam

Jogjakarta, Mei 2007

Penulis



ABSTRAKSI

Dengan semakin berkembangnya peradaban manusia maka nyawa manusia sangatlah berharga, oleh karena itu manusia sering mencari solusi untuk mengatasi masalah-masalah yang dapat membahayakan nyawa manusia dengan menggantikan peran manusia dengan menggunakan robot.

Pada penelitian yang berjudul Mobil Robot Berlengan Dikontrol Lewat Port Paralel ini adalah robot dengan roda dibawahnya serta diatasnya terdapat lengan yang berfungsi untuk mengangkat sesuatu dan dapat bergerak dengan pengendalian melalui port paralel pada komputer dengan menggunakan bahasa pemrograman Borland Delphi 7, jadi untuk mengendalikan robot ini terdapat program yang dibuat untuk mempermudah pengguna dalam pengendaliannya. Pengguna tinggal mengklik dengan mouse bagian mana yang akan digerakkan.

Pada penelitian ini masih perlu lagi untuk dikembangkan menjadi lebih baik, seperti gerakan dari motor stepper yang lambat karena terhubung oleh mur dengan ulir berjarak 1 mm dan menggunakan motor dengan torsi kecil serta penempatan mur yang dipasang dimotor harus benar-benar berada di tengah agar gerakan robot tidak bergoyang-goyang.



DAFTAR ISI

Lembar Pengesahan Pembimbing.....	ii
Lembar Pengesahan Penguji.....	iii
Halaman Persembahan	iv
Halaman Motto.....	v
Kata Pengantar.....	vi
Abstrak.....	vii
Daftar Isi.....	ix
Daftar Tabel.....	xii
Daftar Gambar.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan.....	2
1.4 Batasan Masalah.....	2
1.5 Metodologi Penelitian.....	3
1.6 Sistematika Penulisan.....	3
BAB II DASAR TEORI.....	5
2.1 Pengertian robot.....	5
2.2 Struktur robot.....	5
2.3 Penggerak robot.....	6
2.4 Relay.....	6
2.5 IC ULN 2803.....	8
2.6 Regulator.....	9

2.7 Motor DC.....	10
2.7.1 Prinsip Kerja Motor Arus Searah.....	12
2.7.2 Konstruksi Motor DC.....	14
2.8 Program Borland Delphi.....	15
2.8.1 Menu dan Toolbar.....	16
2.8.2 Object Treeview.....	17
2.8.3 Object Inspector.....	19
2.8.4 Component Palette.....	20
2.8.5 Edit Window.....	21
2.9 Parallel Port.....	23
2.9.1 Komunikasi Paralel.....	24
2.9.2 Data DB 25.....	24
BAB III PERANCANGAN ALAT.....	26
3.1 Blok Diagram.....	26
3.2 Perancangan Perangkat keras.....	27
3.3.1 Rangkaian Power Supply.....	27
3.3.2 Rangkaian Interfacing.....	30
3.3.3 Rangkaian Driver Motor Stepper.....	31
3.3.4 Rangkaian Driver Motor DC.....	32
3.3 Perancangan Perangkat Lunak.....	33
3.3.1 Potongan Inisialisasi program penggerak robot.....	33
3.3.2 Potongan Program Untuk Mengclear Port Data Dan Kontrol.....	34
3.3.3 Program Delay Motor Stepper.....	34
3.3.4 Potongan Program Delay.....	35
3.3.5 Potongan Program Membuka Caput.....	36

3.3.6 Potongan Program Menutup Capit.....	37
3.3.7 Potongan Program Roda Maju.....	37
3.3.8 Potongan program roda mundur.....	38
3.3.9 Potongan program roda kiri.....	39
3.3.10 Potongan program roda kanan.....	40
3.3.11 Potongan program bahu kanan.....	40
3.3.12 Potongan program bahu kiri.....	41
3.3.13 Potongan program lengan bawah naik.....	42
3.3.14 Potongan program lengan bawah turun.....	43
3.3.15 Potongan program lengan atas naik.....	44
3.3.16 Potongan program lengan atas turun.....	45
3.4 Perancangan mekanik.....	46
BAB IV PENGUJIAN ALAT DAN PEMBAHASAN.....	49
4.1 Pengujian sistem elektronik.....	49
4.1.1 Pengujian rangkaian interfacing.....	49
4.1.2 Pengujian power supply.....	49
4.1.3 Pengujian Relay.....	50
4.1.4 Pengujian rangkaian driver stepper.....	50
4.2 Pengujian sistem software.....	52
4.3 Pengujian sistem robot.....	53
4.4 Pengujian keseluruhan.....	53
4.5 Pembahasan.....	57
BAB V KESIMPULAN DAN SARAN.....	58
5.1 Kesimpulan	58
5.2 Saran.....	58

DAFTAR TABEL

Tabel 2.1 Nomor dan alamat register paralel port	25
Tabel 2.2 Nomor dan nama sinyal LPT	25
Tabel 4.1 Pengujian IC I 74573	54
Tabel 4.2 Pengujian IC II 74573	55
Tabel 4.3 Pengujian IC III 74573	55
Tabel 4.4 Pengujian Power suplai	56
Tabel 4.5 Pengujian Relay	56
Tabel 4.6 Pengujian IC I ULN 2803	58
Tabel 4.7 Pengujian IC II ULN 2803	58
Tabel 4.8 Pengujian software	59



DAFTAR GAMBAR

Gambar 2.1 Relay	8
Gambar 2.2 IC ULN 2803	9
Gambar 2.3 IC Regulator	9
Gambar 2.4 Konstruksi motor DC	11
Gambar 2.5 IC ULN 2803	9
Gambar 2.6 Magnet permanen	14
Gambar 2.7 Stator motor stepper	16
Gambar 2.8 Lilitan motor stepper	16
Gambar 2.9 Rotor motor stepper	17
Gambar 2.10 Motor unipolar	18
Gambar 2.11 Tampilan awal memulai delphi	20
Gambar 2.12 Menu dan toolbar	21
Gambar 2.13 Oject treeview	22
Gambar 2.14 Object inspector	24
Gambar 2.15 Component palette	24
Gambar 2.16 Jendela editor	26
Gambar 2.17 Form kosong	27
Gambar 2.18 Paralel Port	27
Gambar 3.1 Diagram blok	29
Gambar 3.2 IC regulattor tegangan	31
Gambar 3.3 sistem power 5 volt	32
Gambar 3.4 Rangkaian power suplai	33
Gambar 3.5 Rangkaian interfacing	34

Gambar 3.6 Rangkaian driver motor DC	32
Gambar 3.7 Diagram alir delay motor stepper	35
Gambar 3.8 Diagram alir membuka caput	36
Gambar 3.9 Diagram alir menutup caput	37
Gambar 3.10 Diagram alir roda maju	38
Gambar 3.11 Diagram alir roda mundur	39
Gambar 3.12 Diagram alir roda kiri	39
Gambar 3.13 Diagram alir roda kanan	40
Gambar 3.16 Diagram alir bahu kanan	41
Gambar 3.17 Diagram alir bahu kiri	42
Gambar 3.18 Diagram alir lengan bawah naik	43
Gambar 3.19 Diagram alir lengan bawah turun	44
Gambar 3.20 Diagram alir lengan atas naik	45
Gambar 3.21 Diagram alir lengan atas turun	46
Gambar 3.22 Mekanik robot	46
Gambar 4.1 Pengujian IC ULN 2803	51
Gambar 4.2 Program mobil robot berlengan dikontrol lewat port printer	52
Gambar 4.3 Mobil robot berlengan	53
Gambar 4.4 Lengan atas turun dan naik	54
Gambar 4.5 Lengan bawah turun dan naik	55
Gambar 4.6 Bahu kekiri dan kekanan	55
Gambar 4.7 Caput buka dan tutup	56

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dengan semakin berkembangnya peradaban manusia , maka nyawa manusia sangatlah berharga, oleh karena itu manusia sering mencari solusi untuk mengatasi masalah-masalah yang dapat membahayakan nyawa manusia dengan menggantikan peran manusia yaitu dengan menggunakan robot. Penggantian peran manusia dengan robot sangatlah menguntungkan sebab robot dapat dikendalikan dengan program untuk melakukan berbagai aktifitas seperti yang dilakukan oleh manusia.

Aplikasi robot sering kita jumpai dalam kehidupan sehari-hari karena robot terdiri dari motor, rangkaian pengontrol dan mekanik. Jadi aplikasi yang menggunakan komponen robot pembuka kaca mobil (*power window*) ,*cd room* dan mesin di pabrik. Robot terdapat dalam berbagai jenis ukuran maupun pengendaliannya tergantung pada tujuan dari pembuatan.

Dengan perkembangan teknologi di bidang elektronika kemampuan suatu robot dapat hampir menyamai manusia atau memiliki kemampuan lebih dalam menjalankan aktifitas tertentu, karena robot dapat dilengkapi dengan sensor yang dapat berfungsi sama dengan indera manusia, bahkan kemampuannya dapat melebihi indera manusia dan dapat melakukan perhitungan secara akurat.

Dari opini tersebut penulis mencoba untuk membuat suatu mobil robot berlengan yang dikontrol lewat port paralel. Jadi robot ini dilengkapi dengan troli dibawah untuk dapat bergerak kesegala arah dan dilengkapi lengan di atasnya yang berfungsi sebagai tangan dengan pengendalian dan pemantauan melalui port printer pada komputer.

1.2 Rumusan Masalah

Bagaimana cara membuat dan mengendalikan robot yang dapat menggantikan peran manusia agar sesuai dengan yang dikehendaki dengan menggunakan komputer melalui port paralel.

1.3 Tujuan

Laporan tugas akhir ini disusun dengan tujuan sebagai berikut :

1. Mempelajari tentang port paralel, fungsi dan aplikasi.
2. Membuat rangkaian antarmuka melalui port paralel.
3. Dapat mengetahui mekanisme pembuatan robot.

1.4 Batasan masalah

Adapun batasan masalah pada penelitian ini yaitu :

1. Gerakan robot ini yang terbatas pada panjangnya kabel sepanjang 1 meter.
2. Tidak dapat mengangkat beban lebih dari $\frac{1}{4}$ kilogram karena torsi dari motor yang tidak terlalu besar.

3. Program Borland Delphi sebagai pemrograman antarmuka untuk mengatur gerakan.

1.5 Metodologi Penelitian

Metodologi yang digunakan dalam menyelesaikan tugas akhir ini yaitu :

1. Studi literatur mengenai komponen-komponen yang digunakan.
2. Mempelajari bahasa pemrograman Borland Delphi.
3. Perancangan dan perakitan elektronik.
4. Perancangan dan perakitan mekanik robot.
5. Mengaplikasikan perancangan software.
6. Melakukan pengujian.
7. Penyusunan laporan dan tugas akhir.

1.6 Sistematika Penulisan

Sistematika penulisan laporan tugas akhir ini akan di sajikan dalam bab-bab seperti yang digambarkan di bawah ini :

1. **Pendahuluan**
Berisi latar belakang masalah, perumusan masalah, batasan masalah, tujuan, metode penelitian serta sistematika penelitian.
2. **Dasar teori**
membahas teori-teori tentang komponen yang digunakan dan tentang bahasa pemrograman.
3. **Perancangan Alat**

membahas perancangan sistem secara keseluruhan dimulai dari perancangan hardware dan software.

4. Pengujian dan analisa

melakukan pengujian-pengujian terhadap sistem yang telah dirancang secara keseluruhan .

5. Kesimpulan

Membuat kesimpulan berdasarkan hasil-hasil pengujian yang telah dilakukan dan memberikan saran-saran.



BAB II

DASAR TEORI

2.1 Pengertian robot

Kemajuan dunia robot yang telah hadir sejak lama hingga sekarang peranannya makin diminati dan berkembang sangat pesat. Robot menurut carel capek berasal dari kata *robota* (bahasa Ceko) yang berarti budak atau pekerja. Defenisi robot secara umum adalah suatu alat multifungsi otomatis yang dapat diprogram untuk melakukan pekerjaan tertentu.

Unsur utama dalam kata robot adalah :

- a. seperangkat mesin atau peralatan (*device*).
- b. Dapat diprogram atau diatur.
- c. Bekerja dan bergerak secara manual dan otomatis.

2.2 Struktur robot

Pada umumnya sebuah robot memiliki struktur sebagai berikut:

- a. Manipulator
 - Badan (*body*).
 - Lengan (*arm*).
 - Pergelangan (*wrist*).
- b. End efecctor
 - Pemotong (*cutter*), penjepit (*gripper*), dan bor (*drill*).
- c. Penggerak (*driver*)

- Motor, hidraulik, pneumatic.
- d. Sensor
 - Panas, posisi, kecepatan dan jarak.
- e. pengendali (*controller*)
 - PC/komputer, mikrokontroler, dan mikroprosesor.

2.3 Penggerak robot

Macam-macam penggerak pada robot yang sering dijumpai dan sering digunakan adalah :

a. Elektris

Tenaga penggeraknya menggunakan jenis motor DC, motor AC, motor stepper dan motor servo.

b. Hidrolis

Penggerak robot pada jenis ini menggunakan semacam cairan untuk menggerakkan. Tenaganya diperoleh dari pompa hidrolis dan pergerakannya diatur oleh *solenoid valve* yang kemudian diteruskan ke bagian *actuator* melalui piston.

c. Pneumatis

Sama seperti penggerak hidrolis tetapi medium yang dipakai untuk meyalurkan tenaga berupa gas atau udara.

2.4 Relai

Relai merupakan saklar elektromagnetik yang dapat mengendalikan peralatan listrik maupun peralatan elektronis, selain digunakan sebagai saklar,

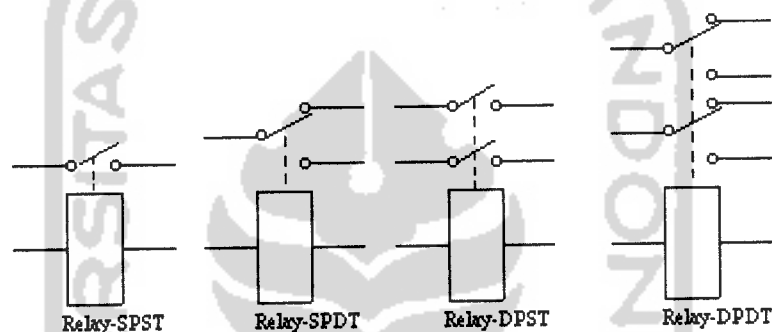
relai juga berfungsi sebagai isolator yaitu antara rangkaian digital yang bertegangan 5 volt dengan rangkaian listrik maupun elektronis yang bertegangan 220 volt dan berdaya besar (arus yang melewatinya besar) sehingga bila terjadi hubungan singkat pada rangkaian listrik maupun elektronis, rangkaian digital tidak akan rusak.

Pada relai biasanya terdapat nilai tegangan yang harus diberikan pada terminal kumparan supaya relai dapat bekerja dengan nilai tegangan dan arus maksimum yang dapat melalui terminal saklar, prinsip kerja relai secara umum adalah jika ada arus listrik yang mengalir dalam kumparan maka inti yang berada ditengah kumparan termagnetisasi berubah menjadi magnet dan mampu menarik plat logam sehingga terminal-terminal saklar yang awalnya *normally open* akan menutup dan sebaliknya terminal yang awalnya *normally close* akan terbuka.

Berdasarkan arus yang bekerja relai dapat dibedakan menjadi dua yaitu relai AC dan relai DC. Berdasarkan hubungan kontak relai dapat dibedakan menjadi empat yaitu :

1. Relai SPDT (*single pole double throw*) merupakan relai yang hanya memiliki satu plat logam dan memiliki dua buah kontak yaitu kontak NC (*normaly close*) dan kontak NO (*normally open*) sehingga apabila *coil* input pada relai diberi tegangan maka terminal saklar yang awalnya *normally close* akan berubah menjadi *normally open*.
2. Relai SPST (*single pole single throw*) merupakan jenis relai yang fungsinya mirip dengan saklar tunggal tetapi berbeda cara penoperasiannya.

3. Relai DPDT (*double pole double throw*) merupakan relai yang memiliki dua buah plat logam dan dua buah kontak NC dan NO. Jika pada koil input pada relai diberi masukan berupa tegangan maka kedua kontak yang awalnya *close* akan berubah menjadi *open*.
4. Relai DPST (*double pole single throw*) merupakan jenis relai yang fungsinya mirip dg dua buah saklar tunggal yang dijadikan satu namun cara pengoperasiannya mirip dengan jenis relai lainnya.



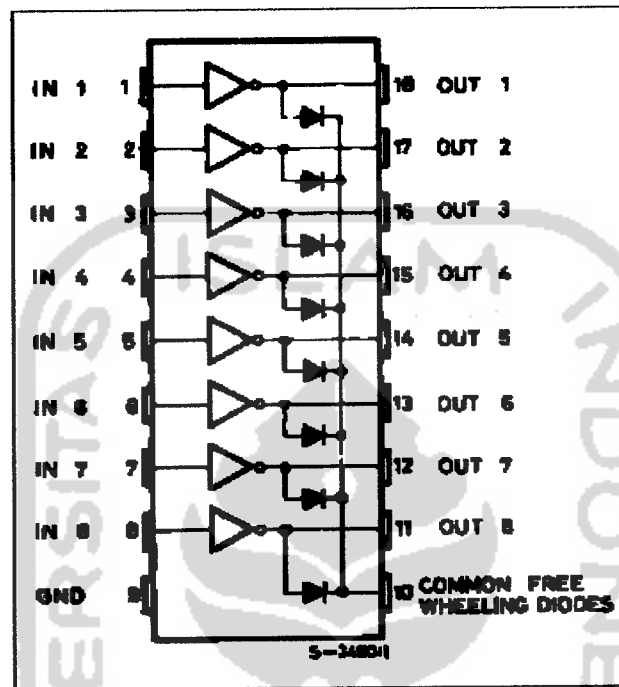
Gambar 2.1 Relay

2.5 IC ULN 2803

IC ULN 2803 adalah sebuah IC dengan ciri-ciri memiliki 8 bit input, tegangan maksimal 50 volt dan arus 500 mA. IC ini termasuk jenis TTL. Didalam IC ini terdapat transistor darlington. Transistor darlington merupakan dua buah transistor yang dirangkai dengan konfigurasi khusus untuk mendapatkan penguatan ganda sehingga dapat menghasilkan penguatan arus yang besar.

IC ULN 2803 merupakan IC yang mempunyai 18 pin. Pin-pin ini ada yang berfungsi sebagai input, output dan ada yang berfungsi sebagai pin catu daya. Catu daya ini terdiri dari catu daya (+) dan ground. Besar catu daya yang

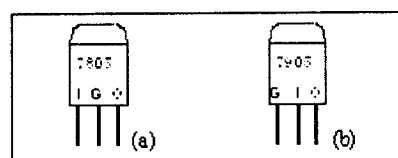
dibutuhkan tergantung pada tipe motor stepper. Sedangkan isi dari IC ULN 2803 adalah sebagai berikut :



gambar 2.2 IC ULN 2803 (Pemrograman IC PPI 8255)

2.6 Regulator

Pemakaian regulator pada pencatu daya berfungsi sebagai stabilitas tegangan. Komponen aktif ini mampu meregulasi tegangan menjadi stabil.



Gambar 2.3 IC Regulator

Komponen ini sudah dikemas dalam sebuah IC regulator tegangan tetap yang biasanya sudah dilengkapi dengan pembatas arus (*current limiter*) dan juga pembatas suhu (*thermal shutdown*). Jenis IC regulator tegangan tetap yang sering dipakai adalah jenis 78xx atau 79xx. IC regulator 78xx menghasilkan output tegangan dengan polaritas positif sedangkan 79xx menghasilkan output tegangan dengan polaritas negatif.

Hanya saja perlu diketahui supaya rangkaian regulator dengan IC tersebut bisa bekerja, tegangan input harus lebih besar dari tegangan output regulatormya. Biasanya perbedaan tegangan V_{in} terhadap V_{out} yang direkomendasikan ada di dalam datasheet komponen tersebut. Pemakaian aluminium pendingin (*heatsink*) dianjurkan jika komponen ini dipakai untuk mencatu arus yang besar.

2.6 Motor Arus Searah (DC)

Motor adalah mesin yang mengubah energi listrik menjadi energi mekanis yang berupa kecepatan atau putaran, dalam hal ini energi listrik yang diubah adalah listrik arus searah (DC). Dengan memperhatikan fungsi dari motor tersebut maka konstruksi motor DC sangat mirip dengan generator DC.

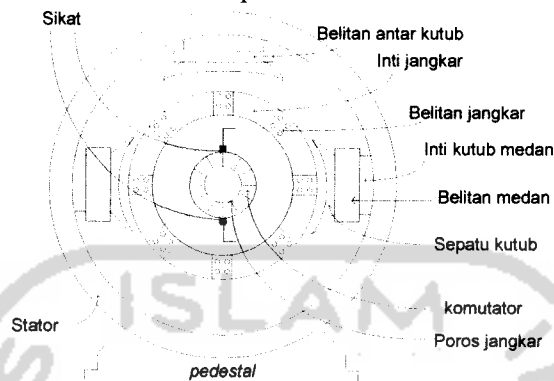
2.6.1 Prinsip Kerja Motor Arus Searah (DC)

Motor Dc bekerja berdasarkan hukum ampere dan hukum lorentz yaitu:

- Di sekitar penghantar yang dialiri arus listrik akan timbul medan magnet.
- Suatu penghantar yang dialiri arus listrik jika berada pada medan magnet akan mengalami suatu gaya yang disebut gaya lorentz.

2.6.2 Konstruksi motor DC

Konstruksi motor DC terlihat seperti Gambar 2.1



Gambar 2.4 Konstruksi motor DC

Rotor merupakan bagian yang berputar di motor. Rotor terdiri dari :

1. Poros jangkar, merupakan bagian yang membawa inti jangkar, kumparan jangkar, dan komutator untuk ikut berputar.
2. Inti jangkar, terbuat dari plat-plat baja yang masing-masing dilapisi isolator.
3. Kumparan jangkar, masing-masing kumparan terisolasi dari yang lain, terletak pada alur (slot) dan terhubung secara elektrik dengan komutator.
4. Komutator, merupakan bagian yang terbuat dari segmen-segmen tembaga yang masing-masing diisolasi dengan bahan mika.
5. Sikat, terbuat dari karbon dan grafit. Sikat terletak pada rumah atau pemegang sikat dengan pegas yang mempunyai kekuatan tekan antara 150-250 gr/cm².

Stator merupakan bagian yang tidak berputar di motor. Stator terdiri dari :

1. Gandar (*yoke*), kerangka motor yang biasanya terbuat dari baja.
2. Kutub, merupakan bagian yang terdiri dari inti kutub, sepatu kutub dan kumparan kutub.

2.6.3 Jenis Motor DC

Motor DC diklasifikasikan berdasarkan pada jenis penguatan atau eksitasinya, yaitu berdasarkan cara pengaliran arus medan. Menurut jenis penguatannya, motor dibagi menjadi :

1. Motor DC dengan penguatan terpisah.
2. Motor DC *shunt*, motor DC dengan penguatan paralel.
3. Motor DC seri, motor DC dengan penguatan seri.
4. Motor DC kompon, motor DC dengan penguatan campuran (seri dan paralel).

2.6.4 Karakteristik motor DC

Karakteristik motor DC ditunjukkan pada Tabel 2.1.

Tabel 2.1 Pemakaian motor DC berdasarkan karakteristiknya

No.	Tipe motor	Karakteristik	Pemakaian
1.	Motor DC <i>shunt</i>	<ul style="list-style-type: none"> • Kecepatan mendekati konstan • Torsi start medium • Kecepatan dapat diatur 	<ul style="list-style-type: none"> • Pompa sentrifugal • Mesin perkakas • Kipas dan <i>blower</i> • Mesin bubut
2.	Motor DC seri	<ul style="list-style-type: none"> • Kecepatan bervariasi • Torsi start tinggi • Pengaturan kecepatan yang bervariasi 	<ul style="list-style-type: none"> • Trem listrik • Pesawat pengangkat (<i>crane</i>) • Mobil <i>trolley</i>
3.	Motor DC kompon	<ul style="list-style-type: none"> • Kecepatan bervariasi • Torsi start tinggi • Pengaturan kecepatan yang bervariasi 	<ul style="list-style-type: none"> • Elevator • Mesin potong dan pelubang • Mesin penggiling • Mesin penggilas

2.7 Motor Stepper

Motor stepper mempunyai karakteristik yang berbeda-beda sesuai dengan jenisnya. Ada tiga tipe dasar dari motor stepper yaitu tipe variabel reluktansi, magnet permanen dan hybrid. Tipe-tipe ini berbeda konstruksi berdasar pada magnet permanen dan besi rotor dengan stator baja berlapis.

2.7.1 Variabel reluktansi

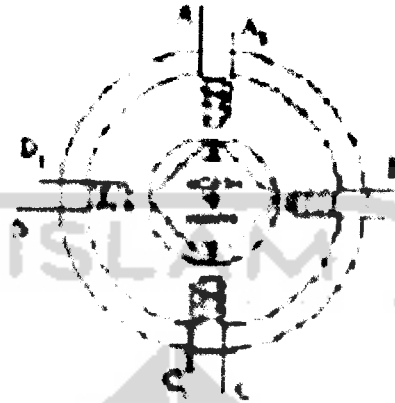
Motor variabel reluktansi tidak menggunakan magnet permanen. Hasilnya rotor motor dapat bergerak tanpa batasan atau penahan torsi. Tipe konstruksi ini baik dalam aplikasi non industri yang tidak membutuhkan torsi motor dalam derajat yang tinggi.

2.7.2 Magnet permanen

Motor magnet permanen juga dikenal sebagai motor canstack. Seperti tersirat dalam namanya, tipe ini mempunyai sebuah rotor permanen. Hal ini memberikan kecepatan rendah secara relatif. Alat torsi rendah dengan sudut step besar, yaitu 45° atau 90° . Motor ini mempunyai konstruksi sederhana dan biaya rendah sehingga membuatnya menjadi pilihan yang ideal untuk aplikasi non industri .

Tidak sama dengan motor stepper lain, rotor motor magnet permanen tidak mempunyai gigi dan dirancang untuk menjadi bermagnet pada sudut siku ke porosnya. Ilustrasi di atas merupakan gambar motor magnet permanen sederhana 90 derajat motor magnet permanen dengan empat fasa (a- d). Penerapan arus ke masing-masing fasa secara berurutan akan menyebabkan rotor berputar dengan

menyesuaikan pada perubahan medan magnet. Walaupun beroperasi pada kecepatan rendah, motor magnet permanen mempunyai karakteristik torsi relatif tinggi.



Gambar 2.6 Motor magnet permanen (Pemrograman IC PPI 8255)

Cara paling mudah untuk membedakan antara tipe motor stepper diatas adalah dengan memutar rotor dengan tangan ketika tak dihubungkan ke suplai. Pada motor stepper yang memiliki magnet permanen, ketika diputar dengan tangan akan tersendat-sendat karena adanya gaya yang ditimbulkan oleh magnet permanen tersebut. Tetapi ketika menggunakan motor dengan variabel reluktansi maka ketika diputar akan terasa halus karena sisa reluktansinya cukup kecil.

2.7.3 Hybrid

Motor hybrid mengkombinasikan karakteristik terbaik dari motor variabel reluktansi dan motor magnet permanen. Motor ini dibangun dengan kutub stator banyak gigi dan rotor magnet permanen. Motor hybrid standar mempunyai 200 gigi rotor dan berputar pada 1,8 derajat sudut step. Motor hybrid lain tersedia

dalam 0,9-3,6 derajat konfigurasi sudut step. Karena memperlihatkan torsi tinggi statis dan dinamis dan berputar pada kecepatan step yang sangat tinggi, motor hybrid digunakan dalam aplikasi industri yang sangat luas.

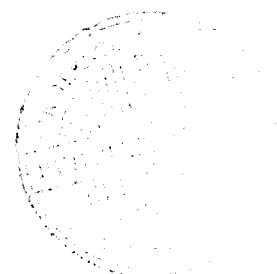
2.7.4 Komponen–komponen motor stepper

Konstruksi motor stepper lebih sederhana daripada motor DC tipe lain karena motor stepper membutuhkan komponen yang lebih sedikit. Pada motor stepper tidak ada komponen sikat arang seperti pada motor DC yang lain. Motor stepper dibentuk oleh beberapa komponen pokok. Komponen pokok motor stepper tersebut adalah stator, lilitan stator dan rotor. Komponen-komponen ini membentuk satu kesatuan tugas, yaitu untuk memutar rotor motor yang dapat difungsikan sebagai magnet dengan kutub tertentu sesuai arah dari pemberian arusnya. Rotor akan berputar jika magnet dari lilitan akan menolak kutub yang sama dan akan menarik kutub yang berbeda. Dengan demikian melalui pemberian pulsa dengan urutan tertentu akan mengakibatkan rotor berputar satu putaran.

2.7.5 Stator

Stator adalah bagian yang diam dari motor stepper dan sebagai tempat untuk lilitan-lilitan motor stepper. Stator terdiri dari besi-besi berlapis. Jumlah lilitannya tergantung tipe dari motor stepper.

Contoh gambar sebuah stator dapat anda lihat pada gambar dibawah ini :

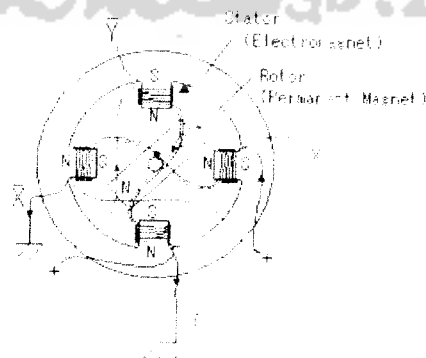




Gambar 2.7 Stator motor stepper (Pemrograman IC PPI8255)

2.7.6 Lilitan stator motor stepper

Motor stepper mempunyai beberapa lilitan yang berfungsi sebagai jalur untuk menerima pulsa-pulsa dari driver motor stepper. Motor stepper dengan jenis yang berbeda memiliki jumlah lilitan yang berbeda pula.



Gambar 2.8 Lilitan motor stepper (Pemrograman IC PPI 8255)

2.7.7 Rotor

Rotor merupakan besi baja yang mengandung sifat magnet. Untuk menghaluskan gerakan perputaran motor stepper biasanya rotor dilengkapi dengan laker dengan bola besi didalamnya. Biasanya laker ini ada di bagian atas dan di bagian bawah magnet rotor sehingga perputarannya dapat seimbang. Motor stepper juga dilengkapi dengan sebuah poros. Poros motor stepper dihubungkan dengan beban dari motor stepper.



Gambar 2.9 Rotor motor stepper(Pemrograman IC PPI 8255)

Rotor adalah bagian motor stepper yang bergerak. Rotor merupakan magnet yang akan menolak kutub yang sama dengannya dan akan menarik kutub yang berbeda dengan polaritasnya. Yang menyebabkan rotor berputar adalah sifat magnet yang ditimbulkan oleh stator motor stepper dan magnet permanen dari rotor motor stepper seperti yang telah dijelaskan .

2.7.8 Lilitan motor

Ada dua macam lilitan motor yaitulilitan unipolar dan lilitan bipolar. Keterangan selengkapnya ada seperti dibawah ini.

2.7.9. Motor stepper unipolar

Motor stepper dengan tipe unipolar mempunyai 2 buah lilitan yang masing-masing lilitan ditengahnya diberikan sebuah tap.

2.7.10 Motor stepper bipolar

pada lilitan bipolar ada dua pasang lilitan yang sama pada masing-masing kutub stator. Tipe konfigurasi lilitan ini menyederhanakan operasi dalam pengiriman arus dari satu lilitan ke yang lain. Lilitan dengan arah kebalikan akan membalikkan arah perputaran rotor motor. Dalam aplikasi unipolar, untuk mengubah arah diperlukan pembalik arus dalam lilitan yang sama.

Konfigurasi kawat yang paling umum pada lilitan bipolar motor stepper adalah 8 ujung, demi fleksibilitas, baik untuk rangkaian seri maupun paralel. Bagaimanapun terdapat banyak motor stepper yang mempunyai 6 ujung untuk aplikasi hubungan seri.

Motor bertipe bipolar ini mempunyai konstruksi hampir sama dengan motor stepper tipe unipolar namun tidak terdapat tap pada lilitannya.

Penggunaan motor tipe ini memerlukan rangkaian sedikit lebih rumit guna mengatur agar motor dapat berputar dalam dua arah. Biasanya untuk menggerakkan motor stepper jenis ini dibutuhkan sebuah driver motor yang dikenal dengan H-bridge. Rangkaian ini akan mengontrol tiap lilitan secara independen, termasuk polaritas untuk tiap-tiap lilitan.

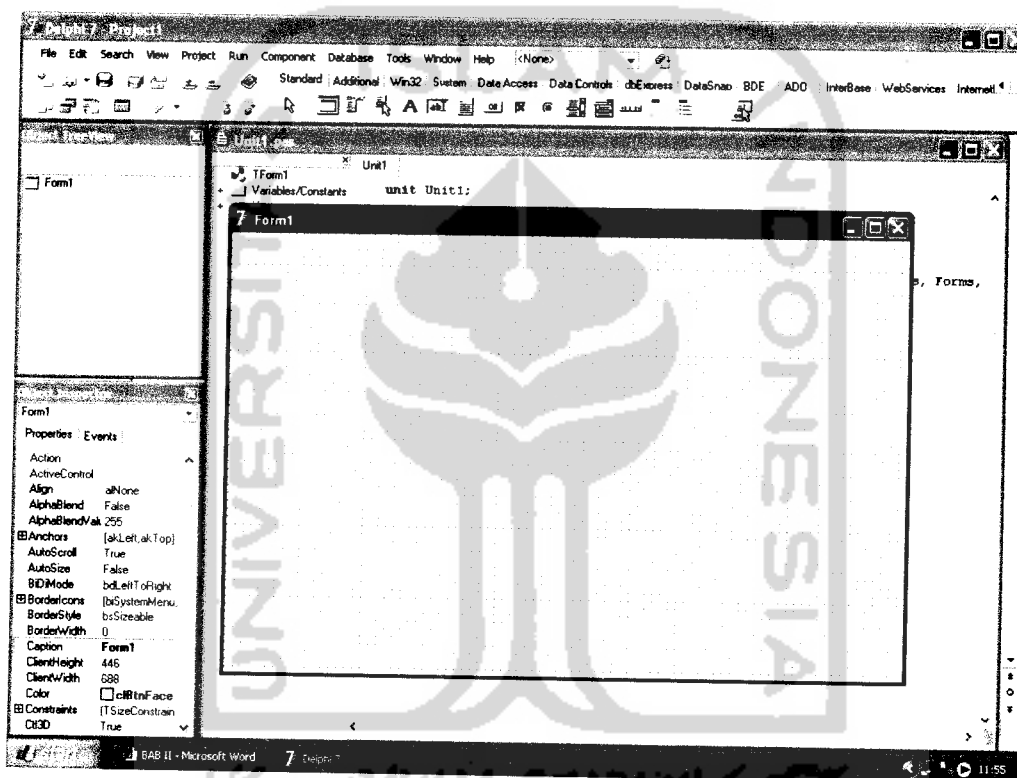
2.8 Program Borland Delphi

Pada awal pemrograman delphi dihadapkan pada suatu lingkungan delphi dimana semua tool yang akan digunakan untuk mendesain dan menjalankan program aplikasi terhubung menjadi satu kesatuan yang memudahkan pengembangan program. Lingkungan ini dinamakan IDE (*Integrated*

Development Environment). IDE terdiri dari toolbar, debugger, editor program dan lain-lain. Semua fasilitas ini akan digunakan pada saat membuat program.

Di dalam tampilan ini dapat dilihat bahwa menu delphi terdiri dari *menu* dan *toolbar*, *object inspector*, *object tree view* serta *komponen palette*. Masing-masing bagian ini mempunyai fungsi ,yaitu sebagai berikut:

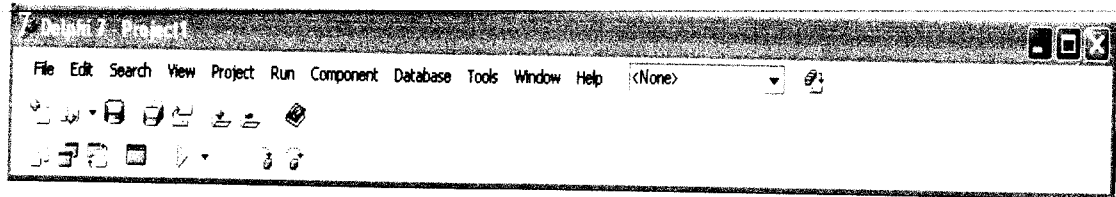
Berikut adalah tampilan awal saat memulai pemrograman delphi.



Gambar 2.11 Tampilan awal saat memulai delphi

2.8.1 Menu dan Toolbar

gambar dibawah ini adalah gambar menu utama delphi dan gambar speedbar. Menu utama delphi berisi tool untuk mengakses lingkungan delphi. Speedbar membantu untuk menjalankan fungsi-fungsi dengan cepat dan mudah.



Gambar 2.12 Menu dan toolbar

2.8.2 Object Treeview

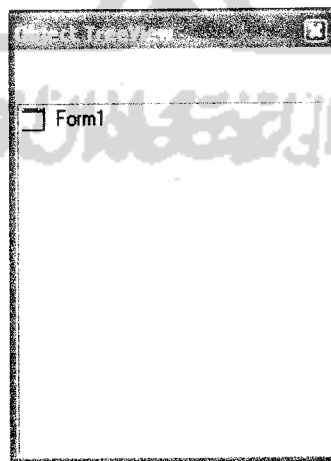
Object treeview menampilkan sebuah diagram pohon dari komponen visual dan non visual yang ditempatkan pada sebuah form, modul data, atau frame. Object treeview menampilkan komponen logika yang berhubungan, yang dikenal dengan istilah hubungan saudara kandung (misalnya antara sebuah komponen dengan komponen palette yang sama), beberapa berhubungan secara lengkap, seperti sebuah komponen data sheet ke propertinya. Dapat membuat hubungan lain dengan menarik dan memasukkan satu komponen diatas yang lain, hanya jika antar komponen itu memiliki kemungkinan untuk berhubungan. sebagai contoh, dapat menarik :

1. komponen berhubungan yang dapat anda kombinasikan, seperti sebuah panel dan sebuah checkbox.
2. sumber data dari satu tabel ke yang lain.
3. basis data dari satu tabel ke bagian yang lain.
4. dataset (seperti *tabel* dan *query*) dari satu basis data ke yang lain.

Beberapa titik dalam diagram pohon ditunjukkan dengan tanda hitam dan putih. Titik ini melambangkan komponen secara penuh. Contohnya, sebuah dataset mempunyai bagian default yang dihubungkan dengannya, bagian default dibuat pada runtime.

Jika sebuah titik dalam diagram pohon mempunyai tanda pertanyaan merah dalam lingkaran kuning, hal itu berarti item itu tidak dapat didefinisikan secara lengkap atau bermasalah. Contohnya, sebuah sumber data milik dataset mempunyai nilai yang muncul dengan tanda pertanyaan merah. Klik dua kali item itu untuk melompat ke tempat dalam kode editor dimana anda dapat menuliskan sebuah event handle. Jika melakukan klik kanan pada item dalam diagram pohon, akan terlihat terjemahan dari menu konteks komponen. Untuk mengakses menu secara penuh, klik kanan komponen yang sama dalam sebuah form, modul data, atau frame.

Untuk membuat sebuah diagram pada halaman diagram kode editor, tarik elemen dari object treeview ke halaman diagram.



Gambar 2.13 Object treeview

2.8.3 Object Inspector

Object inspector adalah hubungan antara penampilan visual aplikasi dan kode yang membuat aplikasi berjalan object inspector menghubungkan pada:

1. Pengaturan desain properti untuk komponen yang di tempatkan pada sebuah form (atau untuk form itu sendiri).
2. Membuat dan membantu mengendalikannya melalui event handler.
3. Menyaring properti dan event yang kelihatan.

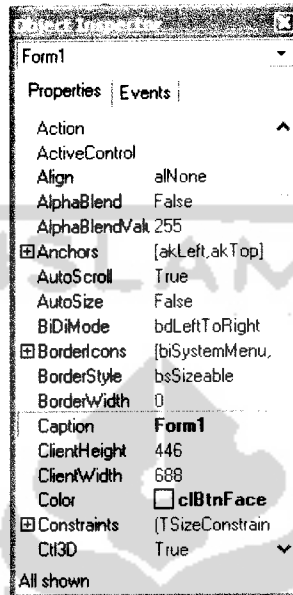
Object selector, atau daftar kejadian, yang ada pada bagian atas object inspector adalah daftar menurun yang mengandung semua komponen pada form aktif dan juga menampilkan tipe object atau kelas dari komponen yang dipilih. Hal ini mengijinkan untuk menampilkan property dan event secara cepat untuk komponen yang berbeda pada form ini.

Dapat juga mengubah ukuran kolom object inspector dengan menarik garis pemisah ke posisi yang baru. Object inspector mempunyai dua tab, yaitu tab properties dan tab events. Tab object inspector memberikan kemungkinan untuk memindahkan tab property dan tab event dari object inspector. Untuk mengubah halaman, klik tab properti atau events.

Menampilkan dan menyaring properti dan event dengan kategori. Dengan menyaring properti, dapat mengurangi jumlah properti yang terlihat dalam object inspector. Juga dapat lebih mudah menempatkan properti yang berhubungan dengan menampilkannya menurut katagori.

Tab properti menyediakan fasilitas untuk melihat dan mengubah properti dari setiap item. Tab event adalah tab yang berisi event-event yang dapat direspon

oleh sebuah object. Misalnya ingin agar sesuatu dilakukan pada saat form ditutup maka dapat diklik tab event dan menyatakan tindakan tersebut sebagai sebuah prosedur pada event on close



Gambar 2.14 Object inspector

2.8.4 Component palette

component palette adalah inventori dari *visual component library* (VCL). Komponen-komponen visual dibagi menjadi beberapa group. Secara default komponen-komponen dikelompokkan sesuai dengan fungsinya. Pengelompokkan ini dinyatakan dengan tab.



Gambar 2.15 Component palette

Komponen merupakan blok bangunan dari setiap aplikasi delphi, dan basis dari visual component library dan borland component library untuk lambang program. Setiap halaman tab dalam komponen palette menampilkan sekelompok ikon yang melambangkan komponen yang dapat di gunakan untuk desain interface aplikasi anda. Untuk menambahkan komponen ke form terbuka, klik dua kali ikon itu dan kemudian atur properti dan kode *event handler*nya.

Component palette membantu untuk menggambar tampilan fitur jendela pop-up kecil yang mengandung nama atau deksripsi singkat dari tombol ketika kursor anda berada di atas tombol lebih dari satu detik.

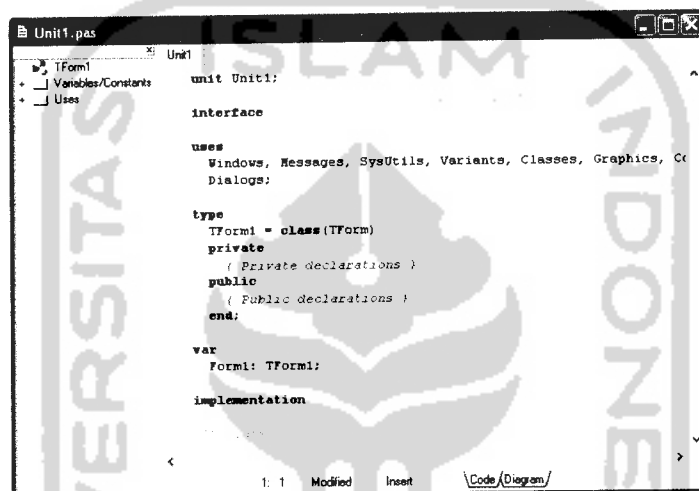
2.8.5 Edit window

jendela editor ini merupakan tempat untuk menuliskan program yang akan anda buat dalam delphi. Editor delphi mempunyai fasilitas hightlight untuk memudahkan dalam menemukan kesalahan. Juga sudah tersedia kerangka program sehingga tidak perlu menuliskan seluruh program.

Bagian kiri jendela editor adalah *code explorer*. *Code explorer* dipakai untuk memudahkan navigasi dialam file unit. Juga dapat mengabaikan code explorer dengan mengklik tanda x di pojok kanan atas code explorer tersebut sehingga code explorer tertutup dan jendela editor hanya akan menampilkan editor program.

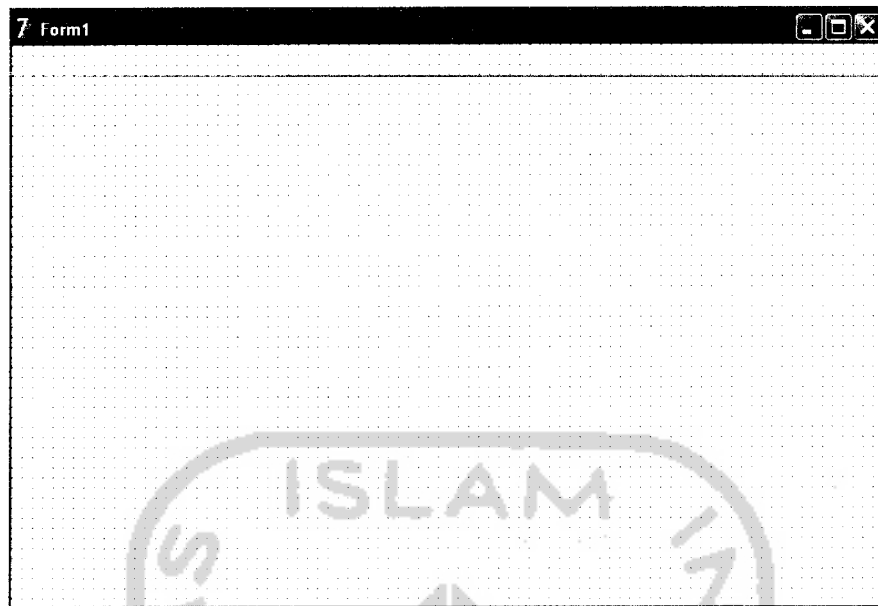
Untuk menampilkan code explorer adalah dengan memilih menu view, code explorer atau klik kanan pada code editor dan pilih view explorer

Title bar berisi nama file yang sedang di edit. Di bagian atas jendela terdapat tab-tab yang menyatakan halaman-halaman yang anda buka. Anda dapat membuka beberapa file dalam sebuah jendela editor dan jga dapat mengaktifkannya dengan menggunakan tab tersebut. Dibagian kiri bawah terdapat penunjuk baris atau kolom yang berfungsi untuk menunjukkan posisi kursor dalam program.



Gambar 2.16 Jendela editor

Setelah mengetahui fungsi masing-masing menu, komponen dan fasilitas yang ada dalam delphi, tempatkan komponen-komponen yang di butuhkan, yang terdapat di dalam component palette, sesuai kebutuhan. Berikut ini contoh sebuah form kosong yang belum diberi komponen.

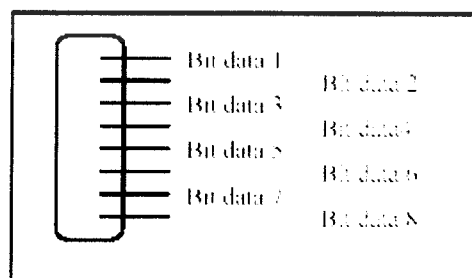


Gambar 2.17 Form kosong

Untuk membuat form kosong ini sebagai papan kendali, perlu menempatkan tombol-tombol ataupun checkbox yang sesuai. Juga dapat memberikan keterangan untuk nama tombol-tombol tersebut sesuai fungsinya sehingga pemakai tidak bingung bila menjalankan program.

2.9 Paralel port

Komunikasi paralel adalah komunikasi yang mengirimkan data secara bersamaan.



Gambar 2.17 Paralel port

Perhatikan gambar diatas, pada penggunaan komunikasi paralel semua bit dikirim secara bersamaan pada waktu yang sama. Oleh karena itu pada komunikasi ini dibutuhkan banyak kabel. Hal memang sering menjadi kelemahan komunikasi paralel akibat banyaknya kabel yang dibutuhkan, dan panjang kabel ini tidak boleh lebih dari 20 m, untuk menjaga keaslian data. Namun kelebihan komunikasi paralel adalah lebih cepat dan kapasitas yang dibawa juga banyak serta pemrograman yang lebih mudah.

2.9.1 Komunikasi paralel

Komunikasi paralel yang digunakan adalah komunikasi paralel lewat kabel data untuk printer (saat mengeluarkan data). Pada keadaan normal (tidak aktif) tegangan pada pin-pin ini adalah 0 volt, namun bila di beri high, maka tegangannya akan berubah menjadi 5 volt. Pada tugas akhir ini, komunikasi paralel hanya digunakan untuk mengeluarkan data, yang bisa berguna untuk menyalakan relay atau stepper motor untuk menjalankan atau mengontrol hardware.

2.9.2 Data DB 25

ada port paralel ada 3 jalur data, yaitu :

- a. Jalur kontrol

Memiliki arah bidirectional

- b. Jalur status

Memiliki satu arah, yaitu arah input.

- c. Jalur data

Memiliki 2 arah. Dapat juga berfungsi sebagai pengirim Address dan data, masing-masing 8 bit, dimana keduanya melakukan transfer data dengan *protokol handshaking* serta diakses dengan register yang berbeda.

Bila kita menggunakan jalur LPT1 maka alamat yang ditentukan oleh PC adalah:

Tabel 2.2 Nomor dan alamat register paralel port

Register kontrol	Register Status	Register Data	Register Address
37A	379	378	37B

Tabel 2.3 Nomor dan nama sinyal LPT

Nomor Pin	Nomor Bit	Jalur	Sinyal
1	1	Kontrol	Strobe
2	1	Data	Data 1
3	2	Data	Data 2
4	3	Data	Data 3
5	4	Data	Data 4
6	5	Data	Data 5
7	6	Data	Data 6
8	7	Data	Data 7
9	8	Data	Data 8
10	7	Status	Ack
11	8	Status	Busy
12	6	Status	P Error
13	5	Status	Select
14	2	Kontrol	Auto FD
15	4	Status	Fault
16	3	Kontrol	Int
17	4	Kontrol	Select In
18 - 25			Ground

BAB III

PERANCANGAN ALAT

3.1 Blok Diagram Sistem

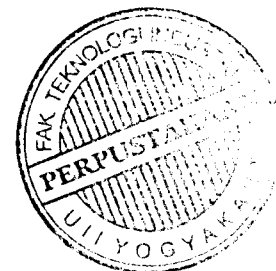
blok diagram mobil robot berlengan ini terdiri dari :

1. Input mouse
2. Personal computer (PC)
3. Rangkaian penggerak motor
4. Motor
5. Mekanik



Gambar 3.1 Blok diagram

Dari gambar 3.1 blok diagram diatas sistem mobil robot berlengan dapat diketahui bahwa setiap ruas mekanik terdapat motor yang berfungsi sebagai penggerak. Robot ini menggunakan 6 motor yaitu : 3 motor dc untuk menggerakkan roda kiri, roda kanan dan capit dan 3 motor stepper untuk menggerakkan bahu, lengan bawah dan lengan atas.



Untuk rangkaian penggerak motor terdapat 4 rangkaian yaitu rangkaian interfacing sebagai pengunci data dan memperbesar jalur data dari port printer, rangkaian relay sebagai *switch power* untuk motor dc dan rangkaian driver stepper untuk memberikan data ke motor stepper.

Rangkaian penggerak ini kemudian terhubung dengan port printer ke pc yang dikendalikan oleh program borland delphi untuk mengubah data masukan yaitu berupa klik mouse pada *push button* untuk selanjutnya memberikan sinyal-sinyal ke port printer.

3.2 Perancangan perangkat keras (*hardware*)

Untuk rangkaian penggerak motor terdapat 4 rangkaian yaitu rangkaian power supply, interfacing, driver untuk motor dc dan rangkaian driver stepper.

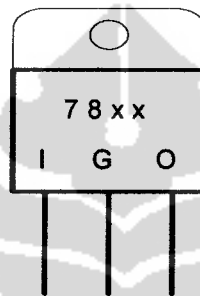
3.2.1 Rangkaian power suplai

Suplai daya adalah sumber tenaga yang dibutuhkan suatu rangkaian elektronik untuk bekerja. Besar suplai daya ini tergantung oleh spesifikasi dari alat masing-masing. Pada perancangan ini suplai daya digunakan untuk mengaktifkan rangkaian driver motor dc dan rangkaian driver motor stepper.

Pada rangkaian Power Suplai pada umumnya sering digunakan IC Regulator dalam mengontrol tegangan yang di inginkan. Regulator tegangan menjadi sangat penting gunanya apabila mengaplikasikan system power tersebut untuk rangkaian – rangkaian yang membutuhkan tegangan yang sangat stabil.

Misalkan untuk system digital, terutama untuk Minimum system (*MikroProsesor/MikroKontroler*) yang sangat membutuhkan tegangan dan arus yang sangat stabil.

IC Regulator yang umum digunakan untuk, mengontrol tegangan adalah IC keluarga 78XX. IC ini dapat mengontrol tegangan dengan baik. Keluaran tegangan yang diinginkan tinggal melihat tipe yang ada. Misalkan tipe 7805 dapat memberikan keluaran tegangan 5 Volt dengan Toleransi +1, dengan arus keluaran maksimal 1500 mA.

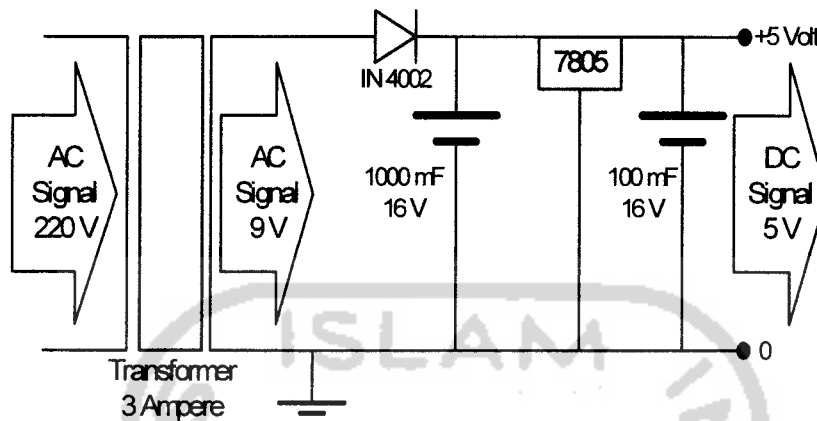


Gambar 3.2 IC Regulator tegangan

3.2.1.1 Rangkaian Tegangan 5 Volt

Rangkaian ini merupakan aplikasi dari regulator tegangan IC 7805, yang dapat mengeluarkan tegangan 5 Volt DC. Rangkaian ini dibangun dari beberapa Komponen yakni, Diode 1N 4002 yang merupakan diode yang dapat melewatkan arus maksimal 3 ampere, selain itu diode ini juga berfungsi untuk menjadikan sinyal AC sinusoidal yang melewatinya menjadi sinyal DC setengah gelombang. Kemudian selain itu dibangun oleh Kapasitor yang berfungsi untuk memperhalus sinyal DC keluaran dari diode. Setelah itu sinyal DC keluaran dari Kapasitor akan di inputkan pada Regulator 7805. Hasil keluaran dari IC 7805 adalah tegangan 5

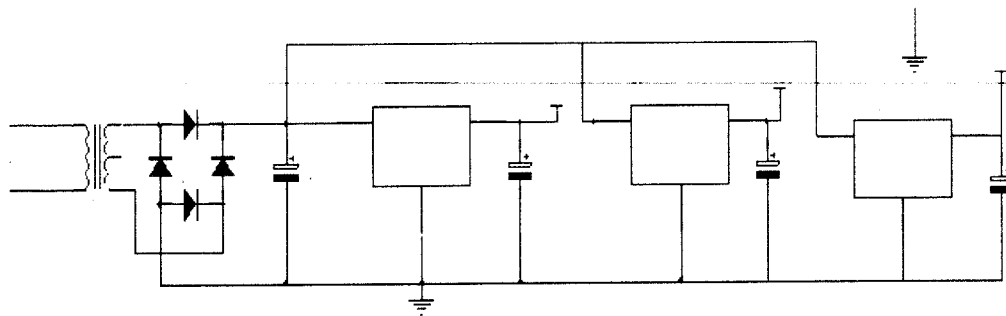
volt dengan arus 1,5 A. Rangkaian ini nantinya akan digunakan untuk memberikan tegangan sebagai suplai driver motor.



Gambar 3.3 Sistem power 5 volt

3.2.1.1 Rangkaian Tegangan 12 Volt

Rangkaian driver motor membutuhkan suplai daya sebesar +12V sedangkan untuk IC membutuhkan suplai daya sebesar 5V. Rangkaian dibawah ini adalah rangkaian suplai daya yang dapat menyediakan keperluan dari rangkaian diatas, masukan rangkaian ini adalah tegangan 220 V yang diubah menjadi tegangan 15 V, dengan menggunakan 4 buah dioda 3A untuk menyearahkan teggangan AC menjadi tegangan DC. Dengan menggunakan IC Regulator untuk mengubah tegangan menjadi tegangan yang diinginkan menggunakan 7805 dan 7812. Rangkaian ini menghasilkan keluaran +5V dan 12V.

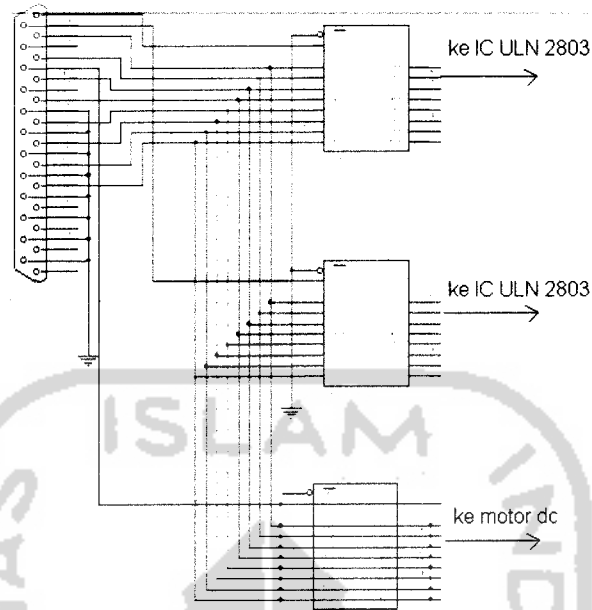


Gambar 3.4 Rangkaian power suplai

Dari gambar 3.4 rangkaian power suply yang digunakan menggunakan trafo 1 ampere yang dihubungkan dengan tegangan 220 volt, 3 buah kapasitor sebesar $100\mu\text{F}$ dan 1 buah kapasitor sebesar $4700\mu\text{F}$ serta menggunakan LM 7805 dan LM 7812 sebagai regulator arah dan menghasilkan tegangan keluaran sebesar 5 volt dan 12 volt

3.2.2 Rangkaian interfacing

rangkaiannya ini terdiri dari 3 IC dimana setiap port data pada DB 25 dihubungkan dengan jalur data pada 3 IC 74573, sehingga jalur data pada port data yang semula 8 diperbanyak oleh IC 74573 menjadi 24 agar dapat memenuhi input ke seluruh driver motor dan relay. Port kontrol (PC0, PC1, PC2) dihubungkan dengan pin c (*select*) untuk memilih IC 74573 mana yang akan diaktifkan. Sedangkan pin oc dihubungkan ke ground agar data dari port printer yang dihubungkan ke output IC diteruskan sampai ke output. IC 74573 yang pertama (D0, D1, D2, D3) dihubungkan dengan driver stepper1, D4, D5, D6, D7 dihubungkan ke driver stepper 2, IC 74573 yang kedua (D0, D1, D2, D3) dihubungkan ke driver stepper 3 dan IC 74573 yang ketiga dihubungkan ke relay dan diteruskan ke motor DC.



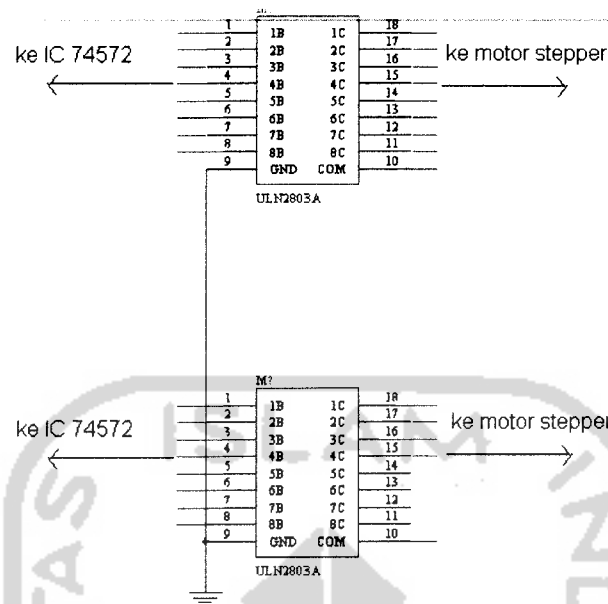
Gambar 3.5 Rangkaian interfacing

3.2.3 Rangkaian driver motor stepper

Gambar dibawah adalah skema rangkaian dari driver motor stepper yang telah dibuat dengan menggunakan IC ULN 2803 dan komponen-komponen pendukungnya. Bentuk jadi dari rangkaian driver motor stepper menggunakan IC adalah sebagai berikut :



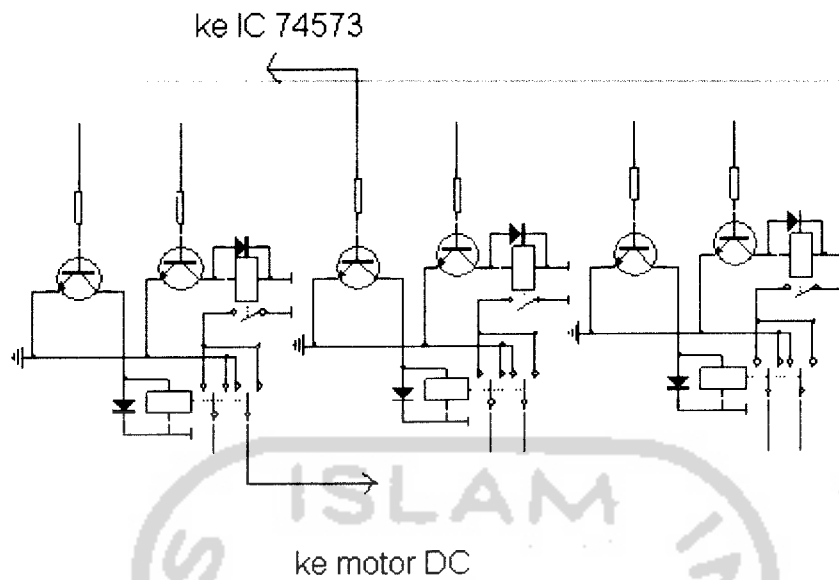
Gambar 3.6 Bentuk jadi driver motor stepper



Gambar 3.7 Rangkaian driver motor stepper

3.2.4 Rangkaian driver motor dc

Driver motor diatas digunakan untuk menggerakkan roda agar bisa maju dan mundur serta menggerakkan caput agar bisa membuka dan menutup, Rangkaian driver motor ini menggunakan 2 buah relay untuk satu buah motor dc. Kedua buah relai tersebut digunakan untuk saklar on/off dan untuk putar kiri dan putar kanan.



Gambar 3.8 Rangkaian driver motor dc

3.3 Perancangan perangkat lunak (*software*)

Untuk menggerakkan robot yang dibuat dengan program delphi, dimana komputer akan melakukan eksekusi perintah sebagai berikut :

1. komputer pertama akan mengirim data ke IC 74573.
2. komputer akan mengaktifkan port kontrol (PC/PC1/PC2) untuk mengaktifkan pin select (c) pada IC 74573 untuk memilih jalur data yang akan dikirim dan mengaktifkan driver atau relai.

3.3.1 Potongan inisialisasi program penggerak robot

Potongan program ini adalah fungsi yang harus ditambahkan untuk mengakses parallel port, karena borland delphi 7 tidak menyediakan fungsi khusus. Oleh karena itu harus ditambahkan fungsi seperti listing program dibawah ini :

```

//fungsi baca port
function TForm1.inport(addr:word):byte;
var tmp:byte;
begin
asm
mov dx,addr;
in ax,dx;
mov tmp,al
end;
inport:=tmp;
end;
//procedure tulis port
procedure TForm1.outport(addr:word;data:byte);
begin
asm
mov dx,addr;
mov al,data;
out dx,al;
end;
end;

```

3.3.2 Potongan program untuk mereset semua port data dan port kontrol

Untuk potongan program mengclear semua port data dan port kontrol, agar konsisi awal pada DB 25 tidak ada data sama sekali pada port data isi=0 sehingga tidak ada lagi gerakan pada robot. Listing program untuk mengclear semua port data dan port control dapat dilihat dibawah ini:

```

//matikan motor
procedure TForm1.kill_motor;
begin
outport($378,$00);
outport($37a,$cc);
outport($37a,$cb);
isi:=0;
end;

```

3.3.3 Potongan program delay motor stepper

Untuk potongan program delay motor stepper sama dengan potongan program untuk memberikan delay disetiap transfer data melalui port data yang akan digunakan untuk memberikan masukan ke driver motor. Listing program delay motor stepper dan digram alir dapat dilihat dibawah ini :

```

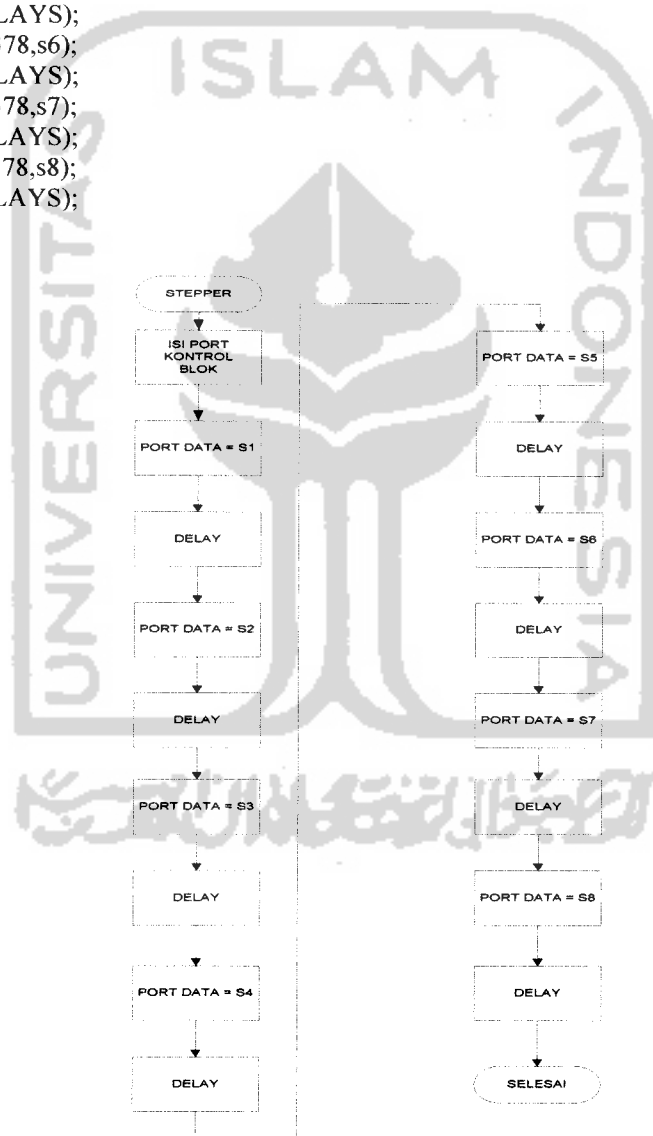
/stepper semua

```

```

procedure TForm1.stepper(blok,s1,s2,s3,s4,s5,s6,s7,s8:integer);
VAR DELAYS:INTEGER;
begin
DELAYS:=inp01.Value;
outport($37a,blok);
outport($378,s1);
delay(DELAYS);
outport($378,s2);
delay(DELAYS);
outport($378,s3);
delay(DELAYS);
outport($378,s4);
delay(DELAYS);
outport($378,s5);
delay(DELAYS);
outport($378,s6);
delay(DELAYS);
outport($378,s7);
delay(DELAYS);
outport($378,s8);
delay(DELAYS);
end;

```



Gambar 3.9 Diagram alir delay motor stepper

3.3.4 Potongan program delay

Untuk potongan program delay digunakan untuk mengatur kecepatan data transfer ke motor stepper, karena motor stepper membutuhkan 4 jalur masukan secara periodik, delay ini akan digunakan pada lamanya transfer data pada saat perpindahan jalur sehingga menyebabkan putaran motor menjadi cepat atau lambat. Listing program dapat dilihat dibawah ini:

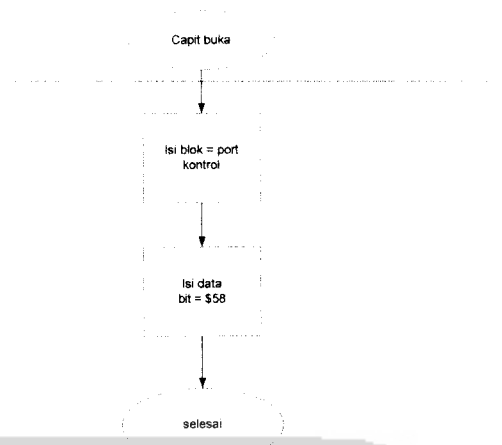
```
//procedure delay
procedure TForm1.delay(detik:longint);
var mulai,stop:dword;
begin
mulai:=gettickcount;
repeat
stop:=gettickcount;
application.ProcessMessages
until(stop-mulai)>=detik;
end;
```

3.3.5 Potongan program membuka caput

Dari program dibawah adalah mengirim data ke port data (\$58) dan memanggil fungsi untuk pengaktifan motor servo dan untuk pengaktifan port kontrol sesudah mengirim data.

```
// caput buka

procedure TForm1.Button1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
PW($58);
end;
```



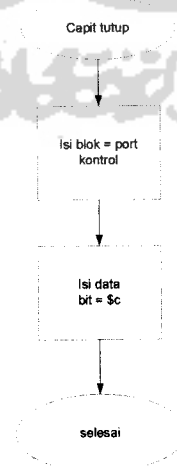
Gambar 3.10 Diagram alir membuka capit

3.3.6 Potongan program menutup capit

Dari program dibawah adalah mengirim data ke port data (\$c) dan memanggil fungsi untuk pengaktifan motor servo dan untuk pengaktifan port kontrol sesudah mengirim data.

```
// capit tutup
```

```
procedure TForm1.Button12MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  PW($C);
end;
```

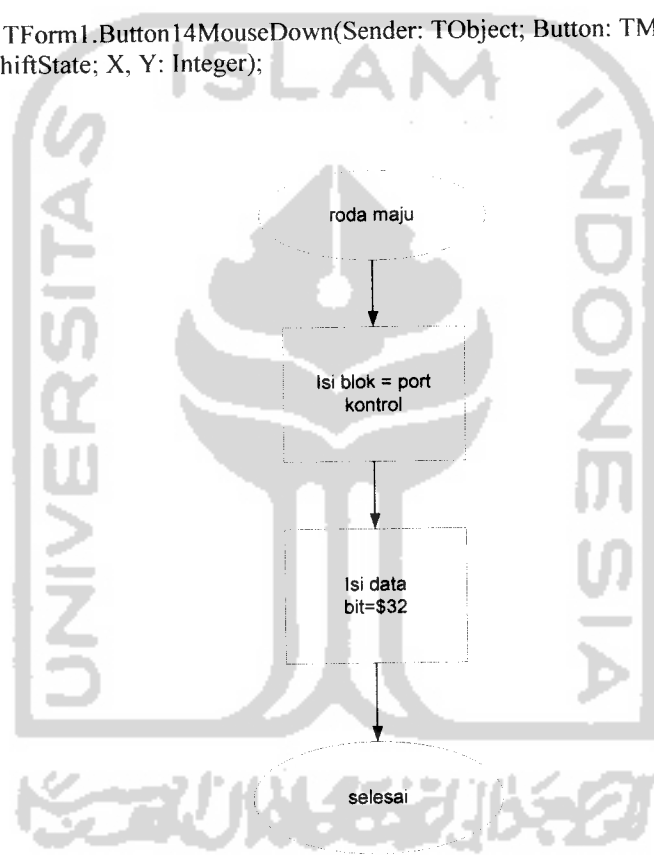


Gambar 3.11 Diagram alir menutup capit

3.3.7 Potongan program roda maju

Dari program dibawah adalah mengirim data ke port data (\$32) dan memanggil fungsi untuk pengaktifan motor servo dan untuk pengaktifan port kontrol sesudah mengirim data.

```
// maju
procedure TForm1.Button14MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  PW($32);
```



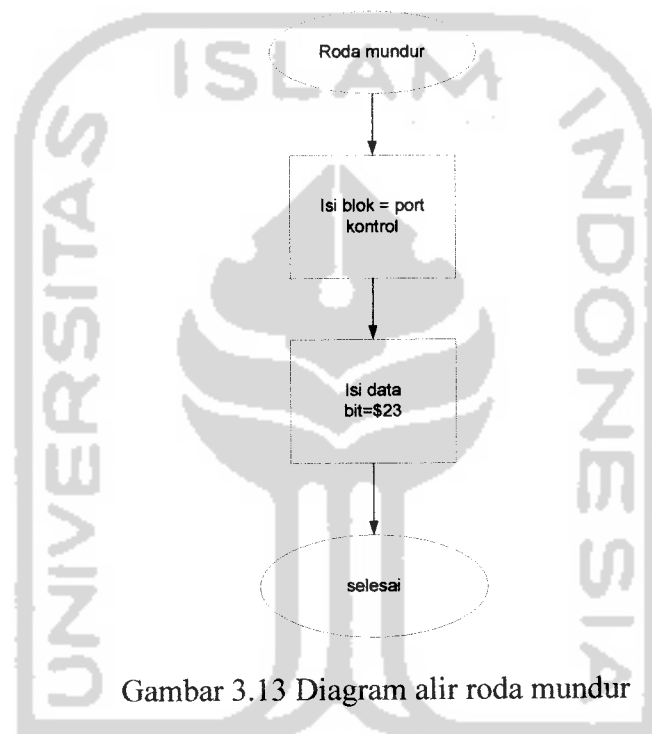
Gambar 3.12 Diagram alir roda maju

3.3.8 Potongan program roda mundur

Dari program dibawah adalah mengirim data ke port data (\$23) dan memanggil fungsi untuk pengaktifan motor servo dan untuk pengaktifan port kontrol sesudah mengirim data.

```
// mundur
```

```
procedure TForm1.Button15MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  PW($23);
end;
```



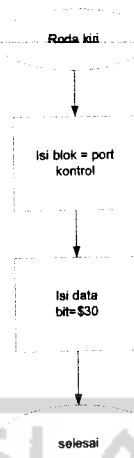
Gambar 3.13 Diagram alir roda mundur

3.3.9 Potongan program roda kiri

Dari program dibawah adalah mengirim data ke port data (\$30) dan memanggil fungsi untuk pengaktifan motor servo dan untuk pengaktifan port kontrol sesudah mengirim data.

```
// kiri
```

```
procedure TForm1.Button13MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  PW($30);
end;
```

Gambar 3.14 Diagram alir roda kiri

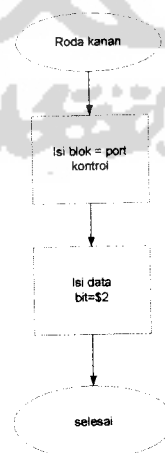
3.3.10 Potongan program roda kanan

Dari program dibawah adalah mengirim data ke port data (\$2) dan memanggil fungsi untuk pengaktifan motor servo dan untuk pengaktifan port kontrol sesudah mengirim data.

/ kanan

```

procedure TForm1.Button16MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  PW($2);
end;
  
```



Gambar 3.15 Diagram alir roda kanan

3.3.11 Potongan program bahu kanan

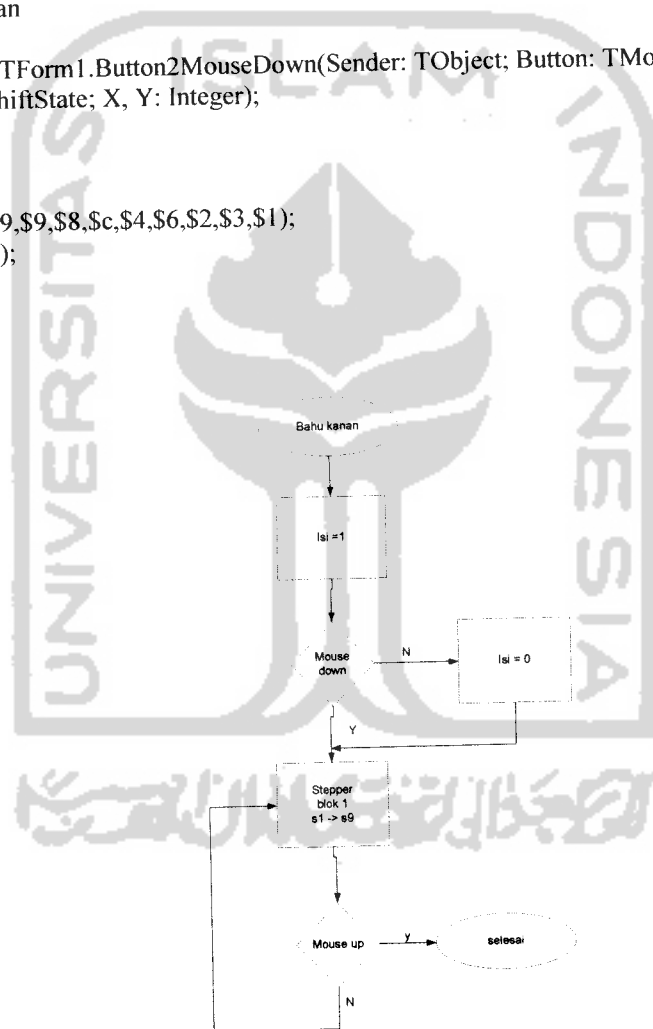
Program dibawah adalah pengaktifan port kontrol PC-1 (autofeed) melalui kaki pin port printer nomor 14 sesudah mengirim data (\$9,\$8,\$c,\$4,\$6,\$2,\$3,\$1) ke IC 2. Sehingga membuat motor stepper yang menggerakkan bahu akan berputar kekanan.

/ bahu kanan

```

procedure TForm1.Button2MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  isi:=1;
  repeat
    stepper($c9,$9,$8,$c,$4,$6,$2,$3,$1);
  until(isi=0);
end;

```

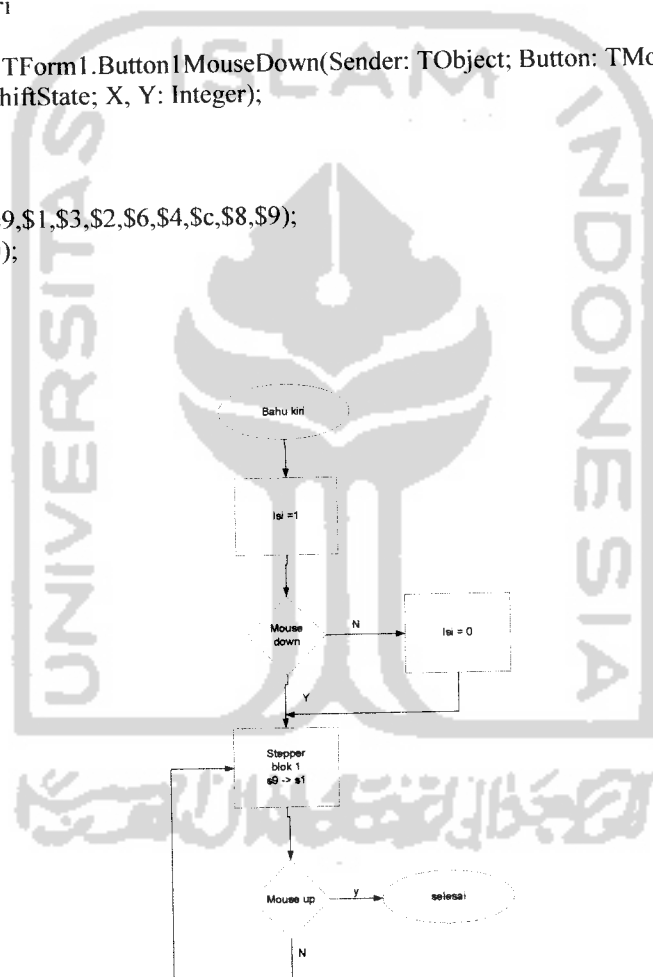


Gambar 3.16 Diagram alir bahu kanan

3.3.12 Potongan program bahu kiri

Program dibawah adalah pengaktifan port kontrol PC-1 (autofeed) melalui kaki pin port printer nomor 14 sesudah mengirim data (\$9,\$8,\$c,\$4,\$6,\$2,\$3,\$1) ke IC 2 sehingga membuat motor stepper yang menggerakkan bahu akan berputar kekiri

```
// bahu kiri
procedure TForm1.Button1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  isi:=1;
  repeat
  stepper($9,$1,$3,$2,$6,$4,$c,$8,$9);
  until(isi=0);
end;
```



Gambar 3.17 Diagram alir bahu kiri

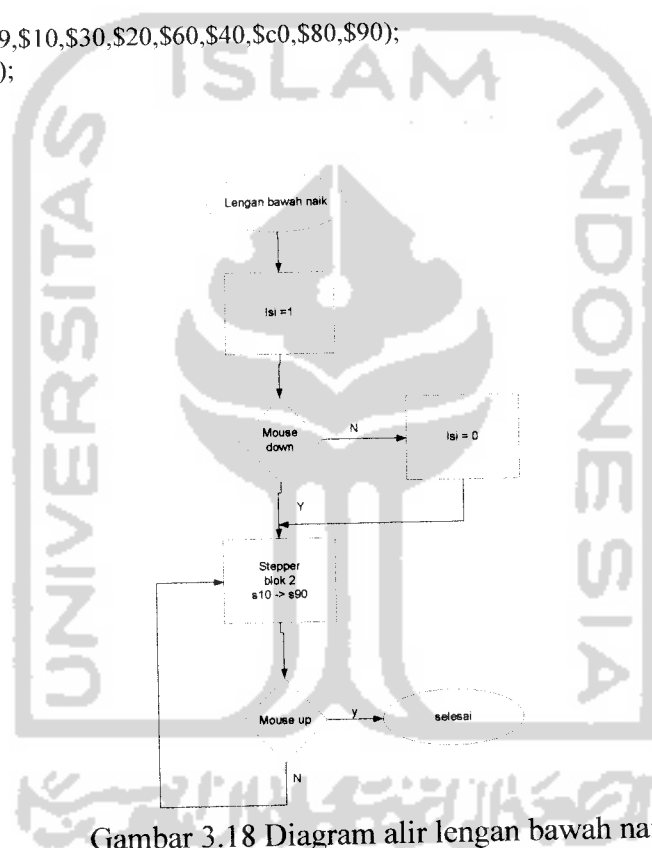
3.3.13 Potongan program lengan bawah naik

Program dibawah adalah pengaktifan port kontrol PC-1 (autofeed) melalui kaki pin port printer nomor 14 sesudah mengirim data

(\$10,\$30,\$20,\$60,\$40,\$c0,\$80,\$90) ke IC 2. Sehingga membuat motor stepper yang menggerakkan lengan bawah akan berputar kekiri

```
// lengan bawah naik
```

```
procedure TForm1.Button4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  isi:=1;
  repeat
  stepper($c9,$10,$30,$20,$60,$40,$c0,$80,$90);
  until(isi=0);
end;
```



Gambar 3.18 Diagram alir lengan bawah naik

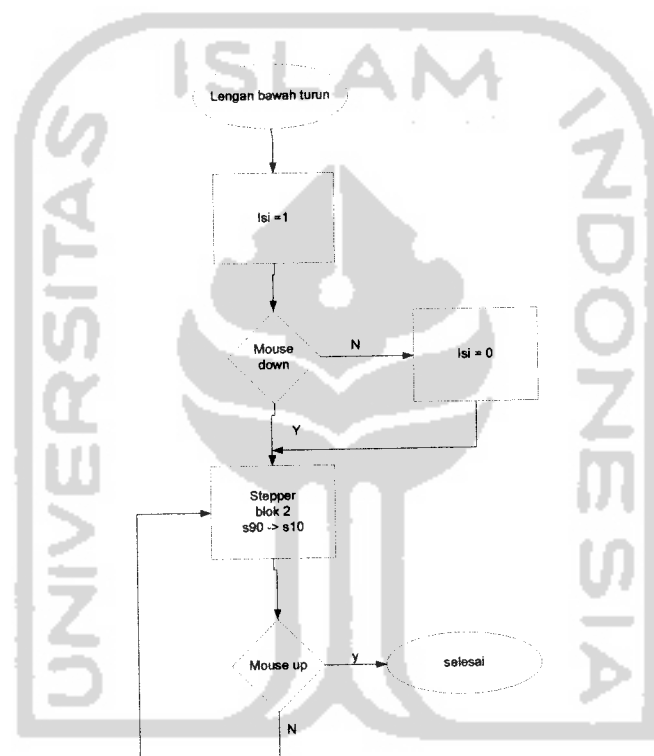
3.3.14 Potongan program lengan bawah turun

Program dibawah adalah pengaktifan port kontrol PC-1 (autofeed) melalui kaki pin port printer nomor 14 sesudah mengirim data (\$90,\$80,\$c0,\$40,\$60,\$20,\$30,\$10) ke IC 2. Sehingga membuat motor stepper yang menggerakkan lengan bawah akan berputar kekanan.

```

// lengan bawah turun
procedure TForm1.Button3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
isi:=1;
repeat
stepper($c9,$90,$80,$c0,$40,$60,$20,$30,$10);
until(isi=0);
end;

```



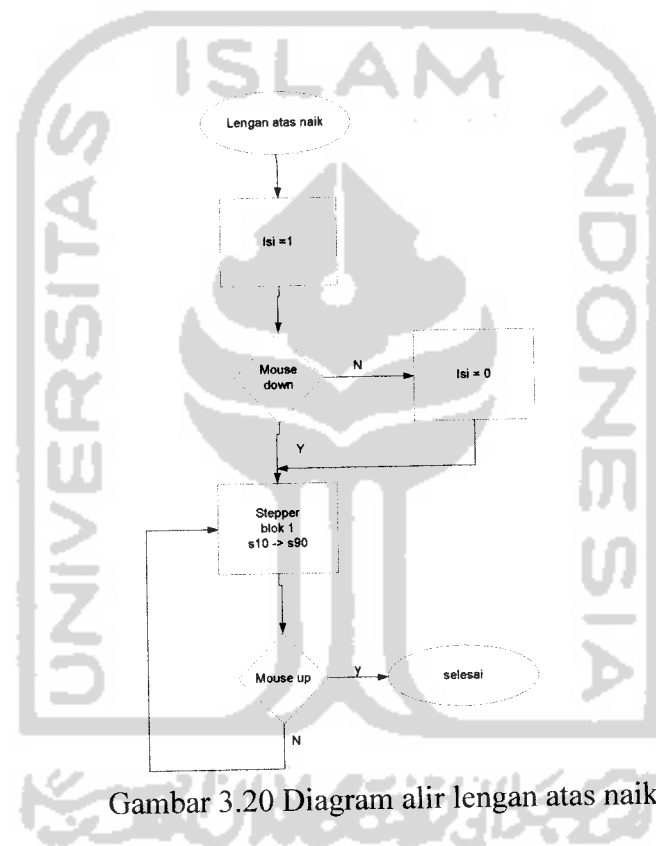
Gambar 3.19 Diagram alir lengan bawah turun

3.3.15 Potongan program lengan atas naik

Program dibawah adalah pengaktifan port kontrol PC-0 (strobe) melalui kaki pin port printer nomor 1 sesudah mengirim data (\$1,\$3,\$2,\$6,\$4,\$c,\$8,\$9) ke IC 1. sehingga membuat motor stepper yang menggerakkan lengan atas akan berputar kekiri.

```
// lengan atas naik
```

```
procedure TForm1.Button7MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  isi:=1;
  repeat
  stepper($ca,$1,$3,$2,$6,$4,$c,$8,$9);
  until(isi=0);
end;
```



Gambar 3.20 Diagram alir lengan atas naik

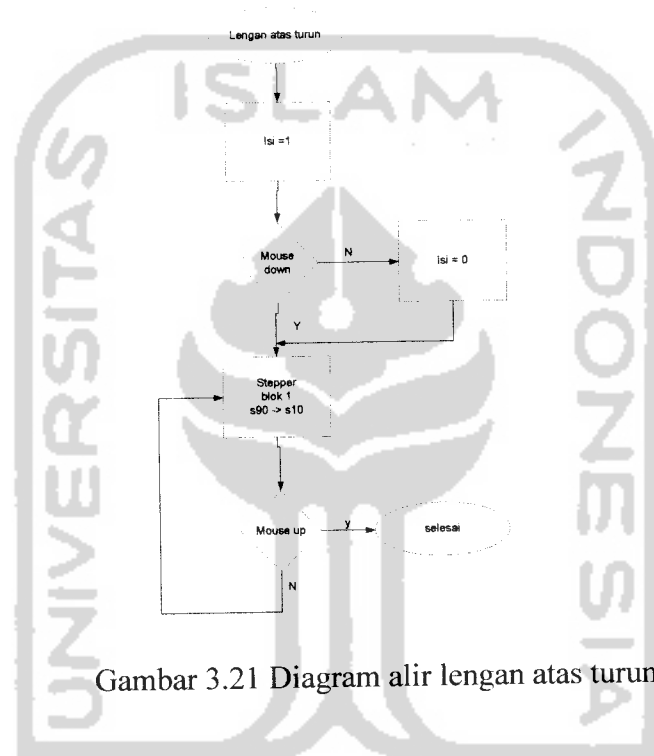
3.3.16 Potongan program lengan atas turun

Program dibawah adalah pengaktifan port kontrol PC-0 (strobe) melalui kaki pin port printer nomor 1 sesudah mengirim data (\$9,\$8,\$c,\$4,\$6,\$2,\$3,\$1) ke IC 1. Sehingga membuat motor stepper yang menggerakkan lengan atas akan berputar kekanan.

```

// lengan atas turun
procedure TForm1.Button8MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
  isi:=1;
  repeat
  stepper($ca,$9,$8,$c,$4,$6,$2,$3,$1);
  until(isi=0);
end;

```



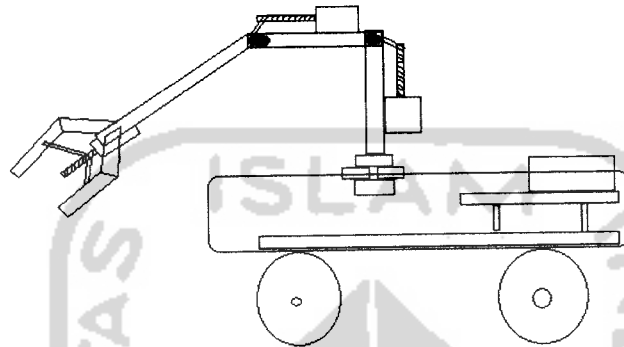
Gambar 3.21 Diagram alir lengan atas turun

3.4 Perancangan mekanik

Sistem perancangan mekanik menggunakan :

- Batang alumunium
- Mur
- Baut
- Roda kursi
- Engsel

- Papan kayu

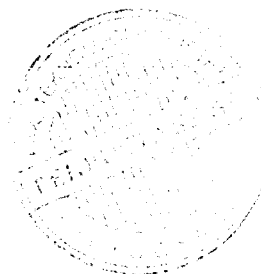


Gambar 3.22 Mekanik robot

Dari gambar 3.22 perancangan mekanik dari robot dapat dijelaskan sebagai berikut :

- Capit terhubung oleh baut dan digerakkan oleh mur yang dihubungkan ke motor dc. Apabila motor dc bergerak kekanan maka baut akan tertarik sehingga capit tertutup dan sebaliknya bila motor bergerak kekiri maka baut akan terdorong sehingga capit akan terbuka
- Lengan atas terhubung dengan baut dan digerakkan oleh mur yang dihubungkan oleh motor stepper apabila motor stepper bergerak kekanan maka baut akan tertarik sehingga lengan atas akan bergerak keatas dan sebaliknya apabila motor stepper bergerak kekiri maka baut akan terdorong sehingga lengan atas akan bergerak kebawah.

- c. Lengan bawah terhubung dengan baut dan digerakkan oleh mur yang dihubungkan oleh motor stepper apabila motor stepper bergerak kekanan maka baut akan tertarik sehingga lengan bawah akan bergerak naik keatas dan sebaliknya apabila motor stepper bergerak kekiri maka baut akan terdorong dan lengan bawah akan bergerak kebawah.
- d. Untuk bahu menggunakan motor stepper dapat langsung dihubungkan dengan mekanik yang telah dibuat sedemikian rupa agar dapat memutar lengan bergerak kekiri dan kekanan.
- e. Motor dc untuk menggerakkan roda dapat langsung dihubungkan dengan roda kursi tanpa harus menggunakan mur atau baut dan untuk gerakan maju ,mundur,kiri dan kanan dari roda mobil tergantung dari masukan yang masuk ke motor dc.



BAB IV

PENGUJIAN ALAT DAN PEMBAHASAN

Pengujian alat

Proses pengujian dilakukan untuk menguji apakah alat yang dirancang sudah berjalan dengan baik atau tidak. Adapun pengujian alat ada 4 yaitu :

1. Pengujian sistem elektronik
2. Pengujian sistem software
3. Pengujian sistem robot
4. Pengujian keseluruhan

4.1. Pengujian sistem elektronik

4.1.1. Pengujian rangkaian interfacing

Pengujian dilakukan dengan cara menghubungkan pin select (c) dan input jalur data pada IC 74573 ke power suplai dari komputer melalui kabel port paralel dengan dilapisi resistor serta output jalur data dihubungkan ke led jika led menyala maka rangkaian interfacing normal.

Tabel 4.1 Pengujian IC I 74573

no	pin	output	led
1	1	5 volt	nyala
2	2	4,97 volt	nyala
3	3	5 volt	nyala
4	4	5 volt	nyala
5	5	4,97 volt	nyala
6	6	5 volt	nyala
7	7	5 volt	nyala
8	8	4,97 volt	nyala

Tabel 4.2 Pengujian IC II 74573

no	pin	output	led
1	1	5 volt	nyala
2	2	4,97 volt	nyala
3	3	5 volt	nyala
4	4	5 volt	nyala
5	5	4,97 volt	nyala
6	6	5 volt	nyala
7	7	5 volt	nyala
8	8	4,97 volt	nyala

Tabel 4.3 Pengujian IC III 74573

no	pin	output	led
1	1	5 volt	nyala
2	2	4,97 volt	nyala
3	3	5 volt	nyala
4	4	5 volt	nyala
5	5	4,97 volt	nyala
6	6	5 volt	nyala
7	7	5 volt	nyala
8	8	4,97 volt	nyala

4.1.2 Pengujian Power suplai

Pada pengujian power suplai tegangan keluaran yang dibutuhkan adalah +12 V , +5 V, dan 0 V dengan tegangan masukan yang dibutuhkan adalah 15 V, dan 0 V. Tegangan keluaran yang dihasilkan oleh power suplai untuk keluaran +12 V adalah +11,88 V, dan tegangan keluaran untuk +5 V adalah +5,04 V, walaupun keluaran power suplai berbeda namun masih dalam batas yang diinginkan. Hal ini bisa terjadi dikarenakan lilitan pada transformator yang tidak sesuai sehingga hasil keluaran berbeda dengan yang diinginkan, hal lainnya yang dapat mempengaruhi tegangan keluaran adalah pada saat pemasangan komponen-komponen elektronika, seperti regulator, dioda dan kapasitor bila pada saat pemasangan atau penyolderan mengalami pemanasan yang berlebih maka

komponen tersebut akan rusak atau hasil keluarannya tidak sesuai dengan harganya.

Pengujian ini dilakukan dilakukan dengan cara pengukuran, tegangan pada keluaran yang telah di searahkan sebesar 12 volt yang akan digunakan untuk tegangan masukan motor dc dan pengukuran pada tegangan keluaran LM 7805 dan LM 7812 yaitu sebesar 5 volt dan 12 volt yang akan digunakan untuk suply driver.

Tabel 4.4 Pengujian power suplai

no	regulator	output
1	7805	5,04 volt
2	7812	12,01 volt
3	7812	12,01 volt

4.1.3 Pengujian Relay

Pengujian rangkaian relay pada gambar dilakukan menghubungkan dengan motor dc dengan menghubungkan transistor ke power supply dan input output switch relay ke motor dc jika motor bergerak berarti rangkaian berhasil ,jika motor tidak bergerak maka ada kesalahan.

Tabel 4.5 Pengujian relay

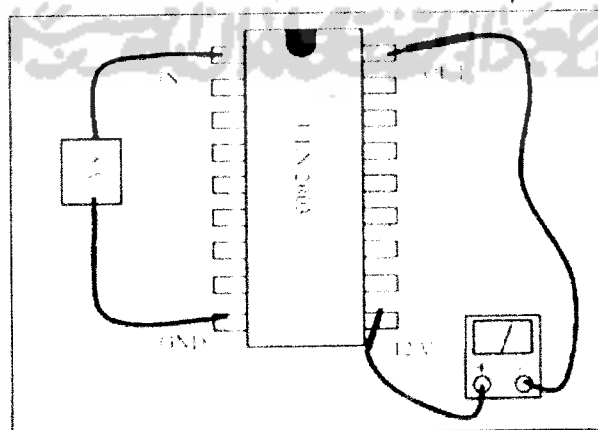
no	motor DC	output relay	output motor
1	roda 1	12 volt	gerak
2	roda 2	12,01 volt	gerak
3	capit	12,01 volt	gerak

4.1.4 Pengujian rangkaian driver stepper

Didalam IC ULN 2803 ada bermacam-macam komponen ,yaitu resistor,transistor dan diioda. Masing-masing komponen ini tidak bisa di uji satu persatu karena masing–masing komponen sudah terangkai menjadi satu dalam IC. Pengujian komponen rangkaian driver ini adalah dengan menguji masing-masing pin dan melihat keluarannya.

Langkah pengujiannya adalah dengan memberikan sinyal berupa tegangan kurang dari 5 volt ke input rangkaian dan kemudian memperhatikan keluaran dari rangkaian, apakah rangkaian sudah mampu mengeluarkan tegangan sebesar catu dayanya.

Dalam pengujian ini, sinyal input tidak harus di ambil dari keluaran IC 74573, tetapi anda juga dapat uji dengan menggunakan batere. Untuk melihat hasil pengujiannya, tegangan dari keluaran IC diukur apakah besarnya sam adengan besar tegangan catu daya.



Gambar 4.1 Pengujian ic ULN 2803

Tabel 4.6 pengujian IC I ULN 2803

no	pin	output
1	1	5 volt
2	2	4,97 volt
3	3	5 volt
4	4	5 volt
5	5	4,97 volt
6	6	5 volt
7	7	5 volt
8	8	4,97 volt

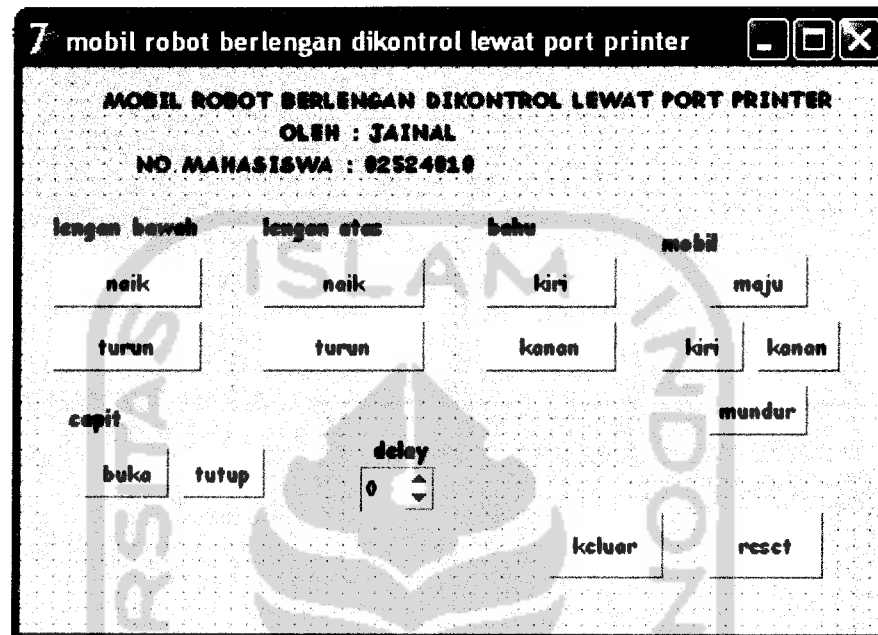
Tabel 4.7 pengujian IC II ULN 2803

no	pin	output
1	1	5 volt
2	2	4,97 volt
3	3	5 volt
4	4	5 volt
5	5	4,97 volt
6	6	5 volt
7	7	5 volt
8	8	4,97 volt

4.2. Pengujian sistem software

Sebelum alat dihubungkan langsung dengan komputer melalui port paralel, terlebih dahulu dilakukan pengujian pada port paralel dengan mengukur tegangan pada pin-pin yang akan digunakan terhadap ground untuk memastikan bahwa port dan program berfungsi dengan baik, untuk menguji port data dan port kontrol dengan cara menjalankan program yang telah dibuat. Kemudian klik push button dan menghubungkan semua kaki port paralel ke led jika led menyala maka program sudah benar. Untuk menyalakan semua led maka tegangan yang terukur adalah 5 volt yang menandakan logika tinggi .sebaliknya pada saat push button

tidak di klik maka semua led mati dan tegangan yang terukur adalah 0 volt untuk lebih jelasnya proses pengujian dapat dilihat pada gambar program berikut :



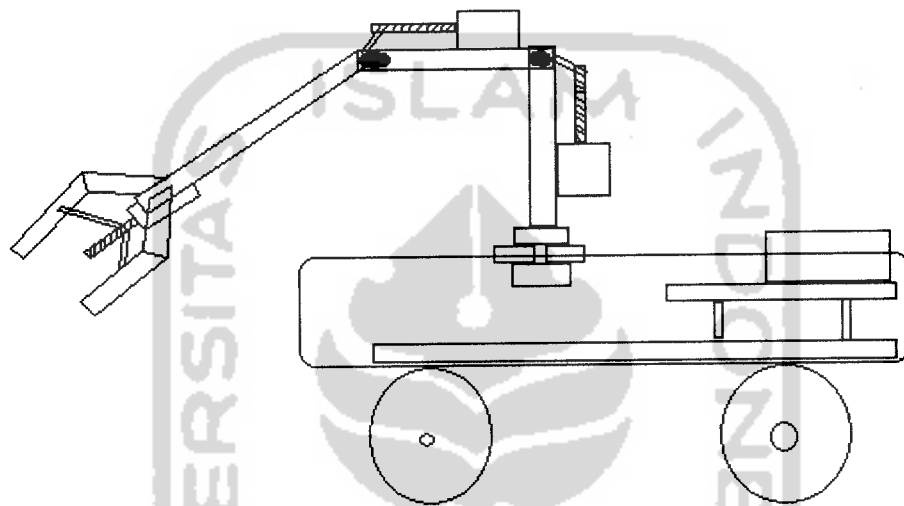
Gambar 4.2 Program mobil robot berlengan dikontrol lewat port printer

Tabel 4.8 Pengujian software

no	pushbutton	output	led
1	lengan bawah naik	5 volt	nyala
2	lengan bawah turun	4,97 volt	nyala
3	lengan atas naik	5 volt	nyala
4	lengan atas turun	4,97 volt	nyala
5	bahu kiri	5 volt	nyala
6	bahu kanan	5 volt	nyala
7	mobil maju	5 volt	nyala
8	mobil mundur	5 volt	nyala
9	mobil kiri	5 volt	nyala
10	mobil kanan	5 volt	nyala
11	capit buka	5 volt	nyala
12	capit tutup	5 volt	nyala

4.3 Pengujian sistem robot

Pengujian sistem robot dengan cara menghubungkan semua motor ke power suplai . Jika mekanik bergerak dengan baik maka robot tidak ada kerusakan sistem akhir mekanik dapat dilihat pada gambar berikut :



Gambar 4.3 Mobil robot berlengan

4.4 Pengujian keseluruhan

Pengujian keseluruhan dari rangkaian ini adalah dengan cara menghubungkan DB 25 dari komputer ke rangkaian yang telah dibuat dan program dijalankan, untuk menggerakkan robot tinggal mengklik push button pada program dan memilih mekanik mana yang akan digerakkan, jika gerakan sesuai dengan program maka pengujian pengujian secara keseluruhan telah berhasil. Pada push button yang menggerakkan mekanik menggunakan motor stepper kecepatannya dapat dikendalikan dengan menaikkan atau menurunkan

delay semakin besar angka yang diberikan pada delay maka gerakan dari motor stepper akan semakin lambat ini disebabkan karena delay membuat sinyal yang diberikan secara periodik dari komputer memiliki selang waktu yang lama dan apabila kita ingin menaikkan kecepatan dari motor stepper angka di kolom delay tinggal dikurangi saja. Untuk menggerakkan lengan atas terdapat dua push button pada program borland delphi yang telah dibuat yaitu tombol naik dan tombol turun.



Gambar 4.4 lengan atas turun dan lengan atas naik

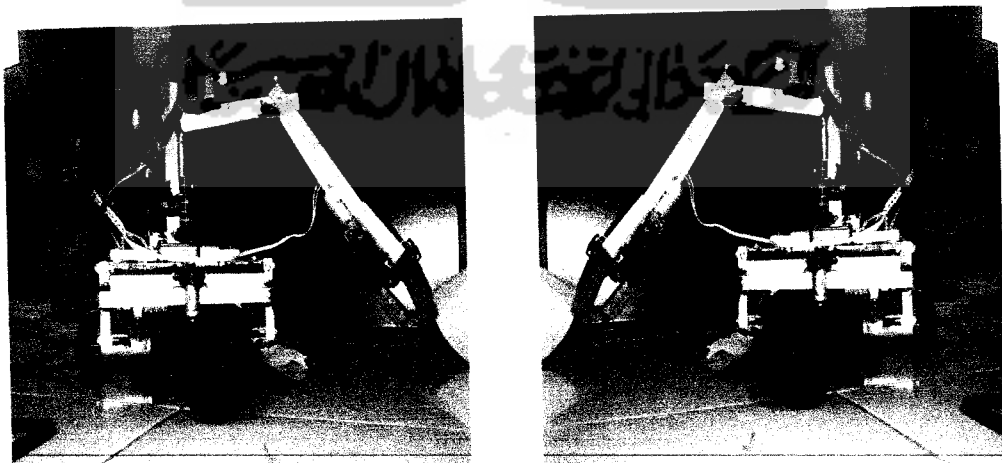
Dari gambar diatas untuk menggerakkan lengan atas tersebut kita hanya memberikan delay sebanyak 1 saja karena beban yang diangkat untuk menggerakkan lengan atas tidak terlalu berat, gerakan dari lengan atas untuk naik dan untuk turun tidak sempurna alias bergoyang-goyang hal ini disebabkan karena mur yang dilekatkan pada motor stepper tidak tepat pada posisi tengah. Sedangkan untuk menggerakkan lengan bawah terdapat dua push button yang digunakan yaitu tombol naik dan tombol turun. Untuk menggerakkan lengan

bawah ini kita juga hanya memberikan delay sebesar 1 saja sama seperti menggerakkan lengan atas.



Gambar 4.5 Lengan bawah turun dan lengan bawah naik

Untuk menggerakkan bahu program pada borland delphi telah dibuat dua push button yaitu tombol kiri dan tombol kanan, pengujian dilakukan dengan mengklik push button kiri dan kanan apabila gerakan sesuai dengan apa yang di klik maka pengujian telah berhasil .



Gambar 4.6 Bahu kekiri dan bahu kekanan

Untuk menggerakkan bahu kekiri dan menggerakkan bahu kekanan pada kolom delay kita berikan sebesar 45 karena beban untuk menggerakkan bahu cukup berat dan dalam pengujian ini bahu dapat digerakkan sesuai apa yang di program. Pada program untuk membuka capit dan menutup capit telah dibuat dua push button yaitu tombol buka dan tombol tutup, untuk lebih jelasnya dapat dilihat pada gambar 4.6



Gambar 4.7 Capit pada posisi buka dan posisi tutup

Pada gerakan ini capit juga tidak bergerak halus seperti yang diinginkan hal ini juga disebabkan oleh penempatan mur yang tidak tepat pada posisi tengah dan tingginya kecepatan yang diberikan oleh motor DC. Untuk pengujian roda mobil maju, mundur, kiri dan kanan sudah sesuai dengan apa yang diinginkan tetapi untuk menggerakkan mobil keiri dan kekanan masih cenderung berat atau lambat ini disebabkan oleh beban yang dibawa oleh mobil cukup berat.

4.5 Pembahasan

Sistem mobil robot berlengan ini dikendalikan oleh komputer melalui port paralel dengan menggunakan kabel data DB 25 male dan female pada tugas akhir

ini kita menggunakan semua jalur data dan 3 jalur kontrol yaitu pin 1, pin 14 dan pin 16, ketiga pin tersebut digunakan untuk mengontrol 3 IC 74573 yang disebut rangkaian interfacing. IC yang pertama digunakan untuk menggerakkan lengan atas, IC yang kedua digunakan untuk menggerakkan bahu dan lengan bawah, lalu IC yang ketiga digunakan untuk menggerakkan roda mobil dan capit.

Pada rangkaian interfacing ini setiap port data pada DB 25 dihubungkan dengan jalur data pada port paralel yang semula 8 diperbanyak menjadi 24 agar dapat memenuhi input ke seluruh driver motor dan relay. Port kontrol (PC-0, PC-1, PC-2) dihubungkan pada pin select yaitu kaki pin nomor 11 untuk memilih IC 74573 mana yang akan diaktifkan. Sedangkan pin OC dihubungkan ke ground agar data dari port printer yang dihubungkan ke D dapat diteruskan sampai ke output. Rangkaian interfacing ini selanjutnya dihubungkan ke DB 25 yang akan melakukan eksekusi perintah sebagai berikut :

1. komputer pertama-tama akan menolak semua port data dan kontrol.
2. komputer akan mengirim data ke IC 74573 melalui kabel data DB 25.
3. komputer mengaktifkan port kontrol dan memilih IC mana yang akan diaktifkan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

1. Setiap mekanik yang digerakkan oleh motor stepper gerakannya lambat sebab setiap motor stepper terhubung dengan mur yang memiliki ulir berjarak 1 mm.
2. Lengan dari robot ini hanya dapat mengangkat benda ringan sebab torsi dari motor kecil.
3. Gerakan dari robot yang bergoyang-goyang disebabkan oleh penempatan mur yang dipasang di motor tidak benar-benar berada pada posisi tengah.

5.2 Saran

1. Mobil robot berlengan ini dapat dikembangkan lagi dengan menambah 1 lengan lagi mengingat masih ada port kontrol yang belum digunakan.
2. Dalam perancangan selanjutnya motor yang digunakan memiliki putaran atau kecepatan yang lebih cepat dari motor yang digunakan dalam tugas akhir ini sehingga robot dapat mengangkat beban yang lebih berat.

DAFTAR PUSTAKA

- Barry G. Wollard, M phil, C Eng, MIERE, M Inst MC, 1999, *Elektronika Praktis*, PT.Pradnya Paramita, Jakarta.
- Eko Putra, Agfianto, 2005, *Pengendalian Mobil Robot Dengan Mobotsim v1.0*, Gava Media, Jogjakarta.
- MADCOMS, 2003, *Pemrograman Borland Delphi 7*, CV.AndiOffset, Jogjakarta.
- Sugiri A.md, S.pd., 2006, *Pemrograman Sistem Pengendali dengan Delphi*, CV.Andi Offset, Jogjakarta.
- Supriadi, Muhammad, 2005, *Pemrograman IC PPI 8255*, CV.Andi Offset, Jogjakarta.
- , 1999, *Precision 1 A regulators*, ST, www.tranzistoare.ro/datasheet/228/390068_ds.pdf.
- , 1990, *Octal D-type transparent lath; 3-state*, philips semiconductor, www.ortodoxism.ro/datasheet/philips/74hc573_cnv_2.pdf.
- , 2003, *NPN general purpose transistor*, Philips semi conductor, www.tranzistoare.ro/datasheet/105/366825_ds.pdf.
- , 1999, *eight darlington arrays*, philips semiconductor, www.tranzistoare.ro/datasheet/150/128380_ds.pdf.

LAMPIRAN




```
init tt;
```

```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls, Forms,  
Dialogs, StdCtrls, Spin;
```

```
var isi:byte;
```

```
type
```

```
TForm1 = class(TForm)
```

```
GroupBox1: TGroupBox;
```

```
Button1: TButton;
```

```
Button2: TButton;
```

```
GroupBox2: TGroupBox;
```

```
Button3: TButton;
```

```
Button4: TButton;
```

```
GroupBox3: TGroupBox;
```

```
Button7: TButton;
```

```
Button8: TButton;
```

```
inp01:TSpinEdit;
```

```
GroupBox4: TGroupBox;
```

```
Button11: TButton;
```

```
Button12: TButton;
```

```
Label1: TLabel;
```

```
GroupBox5: TGroupBox;
```

```
Button13: TButton;
```

```
Button14: TButton;
```

```
Button15: TButton;
```

```
Button16: TButton;
```

```
Button17: TButton;
```

```
Button18: TButton;
```

```
Label2: TLabel;
```

```
Label3: TLabel;
```

```
Label4: TLabel;
```

```
function inport(addr:word):byte;
```

```
procedure outport(addr:word;data:byte);
```

```
procedure delay(detik:longint);
```

```
procedure FormCreate(sender:TObject);
```

```
procedure PW(data:byte);
```

```
Procedure stepper(blok,s1,s2,s3,s4,s5,s6,s7,s8:integer);
```

```
procedure kill_motor;
```

```
procedure Button1MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button2MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button3MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button4MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button7MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button8MouseDown(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button1MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button2MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button3MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button4MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button18Click(Sender: TObject);
```

```
procedure Button17Click(Sender: TObject);
```

```
procedure Button15MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button14MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button16MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button13MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button7MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button8MouseUp(Sender: TObject; Button: TMouseButton;  
Shift: TShiftState; X, Y: Integer);
```

```
procedure Button12MouseUp(Sender: TObject; Button: TMouseButton;
```

```

    Shift: TShiftState; X, Y: Integer);
procedure Button11MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button12MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button11MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button15MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button14MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button16MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button13MouseDown(Sender: TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
procedure Button5Click(Sender: TObject);
procedure Button6Click(Sender: TObject);
procedure Button9Click(Sender: TObject);
procedure Button10Click(Sender: TObject);

```

```

private
    { Private declarations }
public
    { Public declarations }
end;

```

```

var
    Form1: TForm1;

```

implementation

```

{$R *.dfm}
//fungsi baca port
function TForm1.inport(addr:word):byte;
var tmp:byte;
begin
    asm
        mov dx,addr;
        in ax,dx;
        mov tmp,al
    end;
    inport:=tmp;
end;
//procedure tulis port
procedure TForm1.outport(addr:word;data:byte);
begin
    asm
        mov dx,addr;
        mov al,data;
        out dx,al;
    end;
end;
//procedure delay
procedure TForm1.delay(detik:longint);
var mulai,stop:dword;
begin
    mulai:=gettickcount;
    repeat
        stop:=gettickcount;
        application.ProcessMessages
    until (stop-mulai)>=detik;
end;
//form inisialisasi
procedure TForm1.FormCreate(sender:TObject);
begin
    inp01.Value:=10;
    kill_motor;
    isi:=0;
end;
//pengaktifan pw
procedure TForm1.PW(data:byte);
begin
    outport($378,data);
    outport($37a,$cf);
    outport($37a,$cb);
end;

```



```

stepper semua
procedure TForm1.stepper(blok,s1,s2,s3,s4,s5,s6,s7,s8:integer);
var DELAYS:INTEGER;
begin
DELAYS:=inp01.Value;
outport($37a,blok);
outport($378,s1);
delay(DELAYS);
outport($378,s2);
delay(DELAYS);
outport($378,s3);
delay(DELAYS);
outport($378,s4);
delay(DELAYS);
outport($378,s5);
delay(DELAYS);
outport($378,s6);
delay(DELAYS);
outport($378,s7);
delay(DELAYS);
outport($378,s8);
delay(DELAYS);
end;
//matikan motor
procedure TForm1.kill_motor;
begin
outport($378,$00);
outport($37a,$cc);
outport($37a,$cb);
isi:=0;
end;

// bahu kiri

procedure TForm1.Button1MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
isi:=1;
repeat
stepper($c9,$1,$3,$2,$6,$4,$c,$8,$9);
until (isi=0);
end;

// bahu kanan

procedure TForm1.Button2MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
isi:=1;
repeat
stepper($c9,$9,$8,$c,$4,$6,$2,$3,$1);
until (isi=0);
end;

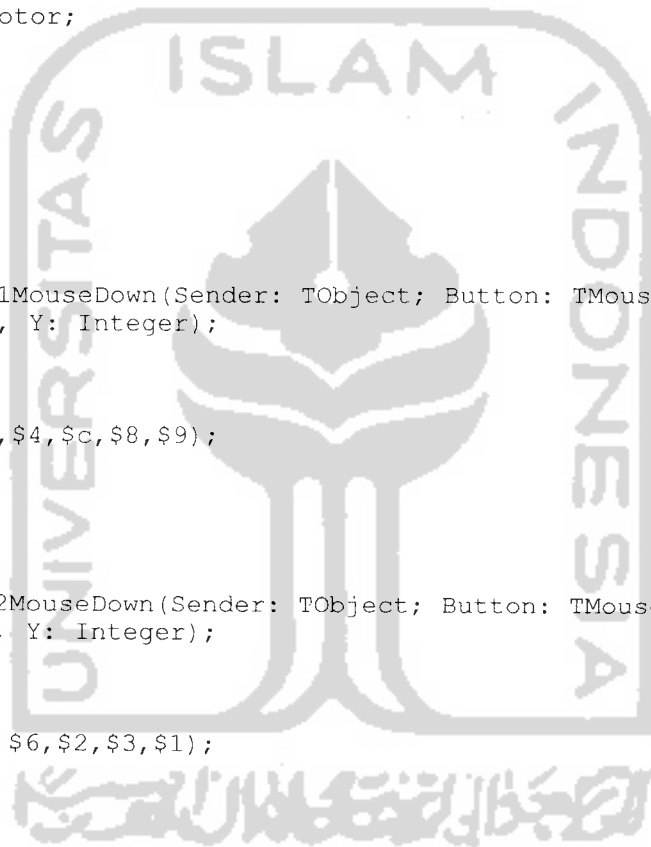
// lengan bawah turun

procedure TForm1.Button3MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
isi:=1;
repeat
stepper($c9,$90,$80,$c0,$40,$60,$20,$30,$10);
until (isi=0);
end;

// lengan bawah naik

procedure TForm1.Button4MouseDown(Sender: TObject; Button: TMouseButton;
  Shift: TShiftState; X, Y: Integer);
begin
isi:=1;
repeat
stepper($c9,$10,$30,$20,$60,$40,$c0,$80,$90);
until (isi=0);
end;

```



lengan atas naik

```
procedure TForm1.Button7MouseDown(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);  
begin  
  i:=1;  
repeat  
  pepper($ca,$1,$3,$2,$6,$4,$c,$8,$9);  
until(isi=0);  
end;
```

lengan atas turun

```
procedure TForm1.Button8MouseDown(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);  
begin  
  i:=1;  
repeat  
  pepper($ca,$9,$8,$c,$4,$6,$2,$3,$1);  
until(isi=0);  
end;
```

```
procedure TForm1.Button1MouseDown(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button7MouseUp(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button8MouseUp(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button2MouseUp(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button3MouseUp(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button4MouseUp(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button18Click(Sender: TObject);
```

```
begin  
  kill_motor;  
end;
```

```
procedure TForm1.Button17Click(Sender: TObject);
```

```
begin  
  kill_motor;  
close;  
end;
```

```
procedure TForm1.Button15MouseUp(Sender: TObject; Button: TMouseButton;  
  Shift: TShiftState; X, Y: Integer);
```

```
begin  
  kill_motor;  
end;
```



```

cedure TForm1.Button14MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  l_motor;
;

cedure TForm1.Button16MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  l_motor;
;

cedure TForm1.Button13MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  ll_motor;
;

cedure TForm1.Button12MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  ll_motor;
;

cedure TForm1.Button11MouseUp(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  ll_motor;
;

/ capit tutup

cedure TForm1.Button12MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  V($C);
;

/ capit buka

cedure TForm1.Button11MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  W($58);
;

/ mundur

cedure TForm1.Button15MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  W($23);
;

/ maju

cedure TForm1.Button14MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  W($32);
;

nd;

/ kanan

cedure TForm1.Button16MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
  V($2);
;

nd;

/ kiri

```

```
procedure TForm1.Button13MouseDown(Sender: TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
in
$30);
;
```

engan bawah naik terus

```
procedure TForm1.Button5Click(Sender: TObject);
in
:=1;
repeat
upper($c9,$10,$30,$20,$60,$40,$c0,$80,$90);
until(isi=0);
;
```

lengan bawah turun terus

```
procedure TForm1.Button6Click(Sender: TObject);
begin
i:=1;
repeat
upper($c9,$90,$80,$c0,$40,$60,$20,$30,$10);
until(isi=0);
end;
```

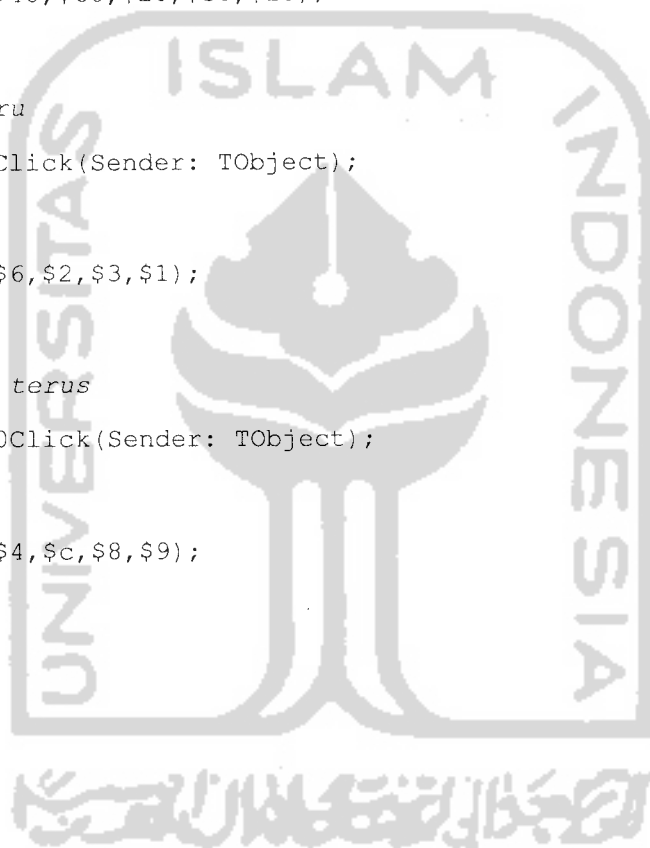
/ lengan atas turun terus

```
procedure TForm1.Button9Click(Sender: TObject);
begin
isi:=1;
repeat
upper($ca,$9,$8,$c,$4,$6,$2,$3,$1);
until(isi=0);
end;
```

/ lengan atas naik terus

```
procedure TForm1.Button10Click(Sender: TObject);
begin
isi:=1;
repeat
upper($ca,$1,$3,$2,$6,$4,$c,$8,$9);
until(isi=0);
end;
```

end.



Octal D-type transparent latch; 3-state

74HC/HCT573

FEATURES

- Inputs and outputs on opposite sides of package allowing easy interface with microprocessors
- Useful as input or output port for microprocessors/microcomputers
- 3-state non-inverting outputs for bus oriented applications
- Common 3-state output enable input
- Functionally identical to the "563" and "373"
- Output capability: bus driver
- I_{CC} category: MSI

GENERAL DESCRIPTION

The 74HC/HCT573 are high-speed Si-gate CMOS devices and are pin compatible with low power Schottky TTL (LSTTL). They are specified in compliance with JEDEC standard no. 7A.

The 74HC/HCT573 are octal D-type transparent latches featuring separate D-type inputs for each latch and 3-state outputs for bus oriented applications.

A latch enable (LE) input and an output enable (\overline{OE}) input are common to all latches.

The "573" consists of eight D-type transparent latches with 3-state true outputs. When LE is HIGH, data at

the D_n inputs enter the latches. In this condition the latches are transparent, i.e. a latch output will change state each time its corresponding D-input changes.

When LE is LOW the latches store the information that was present at the D-inputs a set-up time preceding the HIGH-to-LOW transition of LE.

When \overline{OE} is LOW, the contents of the 8 latches are available at the outputs. When \overline{OE} is HIGH, the outputs go to the high impedance OFF-state.

Operation of the \overline{OE} input does not affect the state of the latches.

The "573" is functionally identical to the "563" and "373", but the "563" has inverted outputs and the "373" has a different pin arrangement.

QUICK REFERENCE DATA

GND = 0 V; T_{amb} = 25 °C; t_r = t_f = 6 ns

SYMBOL	PARAMETER	CONDITIONS	TYPICAL		UNIT
			HC	HCT	
t _{PHL} / t _{PLH}	propagation delay	C _L = 15 pF; V _{CC} = 5 V			
	D _n to Q _n		14	17	ns
	LE to Q _n		15	15	ns
C _I	input capacitance		3.5	3.5	pF
C _{PD}	power dissipation capacitance per latch	notes 1 and 2	26	26	pF

Notes

1. C_{PD} is used to determine the dynamic power dissipation (P_D in μW):

$$P_D = C_{PD} \times V_{CC}^2 \times f_i + \sum (C_L \times V_{CC}^2 \times f_o) \text{ where:}$$

f_i = input frequency in MHz; f_o = output frequency in MHz

∑ (C_L × V_{CC}² × f_o) = sum of outputs

C_L = output load capacitance in pF; V_{CC} = supply voltage in V

2. For HC the condition is V_I = GND to V_{CC}; for HCT the condition is V_I = GND to V_{CC} - 1.5 V

ORDERING INFORMATION

See "74HC/HCT/HCU/HCMOS Logic Package Information".

Octal D-type transparent latch; 3-state

74HC/HCT573

PIN DESCRIPTION

PIN NO.	SYMBOL	NAME AND FUNCTION
2, 3, 4, 5, 6, 7, 8, 9	D ₀ to D ₇	data inputs
11	LE	latch enable input (active HIGH)
1	\overline{OE}	3-state output enable input (active LOW)
10	GND	ground (0 V)
19, 18, 17, 16, 15, 14, 13, 12	Q ₀ to Q ₇	3-state latch outputs
20	V _{CC}	positive supply voltage

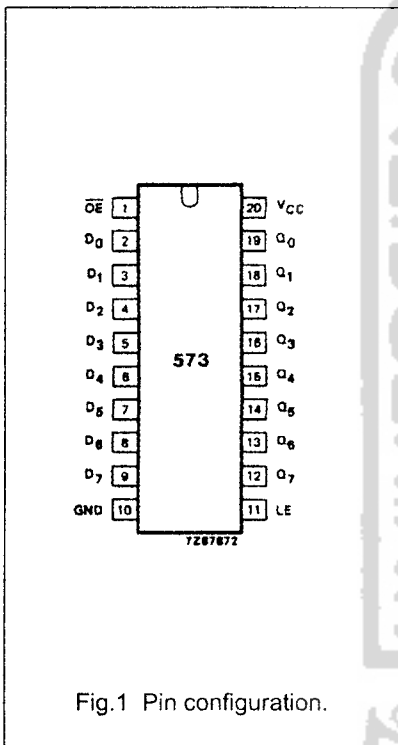


Fig.1 Pin configuration.

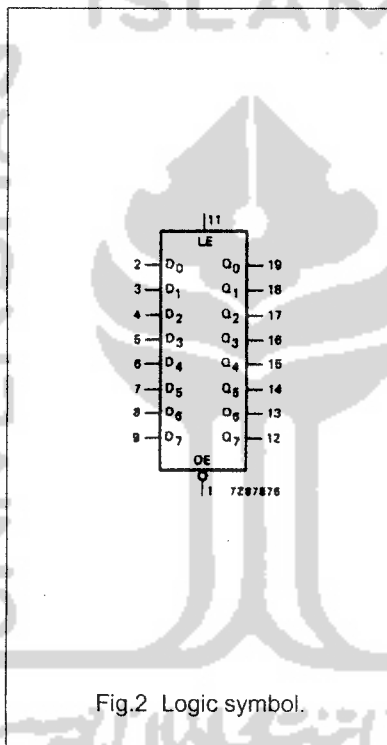


Fig.2 Logic symbol.

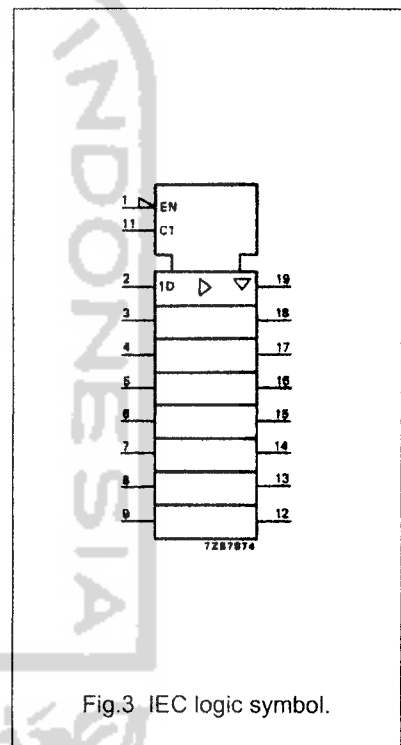


Fig.3 IEC logic symbol.

Octal D-type transparent latch; 3-state

74HC/HCT573

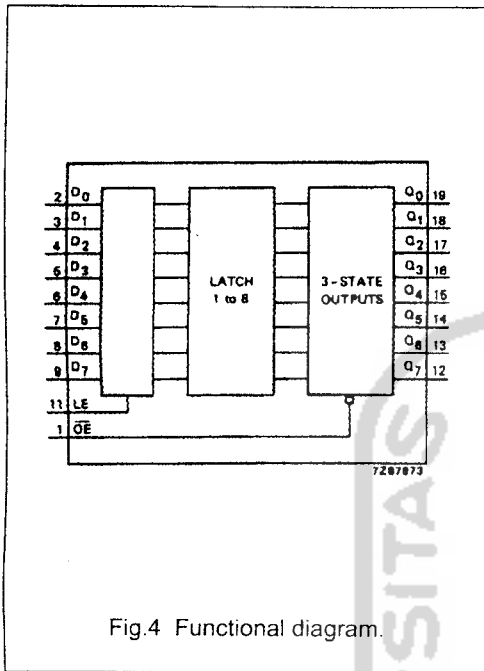


Fig.4 Functional diagram.

FUNCTION TABLE

OPERATING MODES	INPUTS			INTERNAL LATCHES	OUTPUTS Q ₀ to Q ₇
	\overline{OE}	LE	D _N		
enable and read register (transparent mode)	L	H	L	L	L
	L	H	H	H	H
latch and read register	L	L	l	L	L
	L	L	h	H	H
latch register and disable outputs	H	L	l	L	Z
	H	L	h	H	Z

Notes

- H = HIGH voltage level
 h = HIGH voltage level one set-up time prior to the HIGH-to-LOW LE transition
 L = LOW voltage level
 l = LOW voltage level one set-up time prior to the HIGH-to-LOW LE transition
 Z = high impedance OFF-state

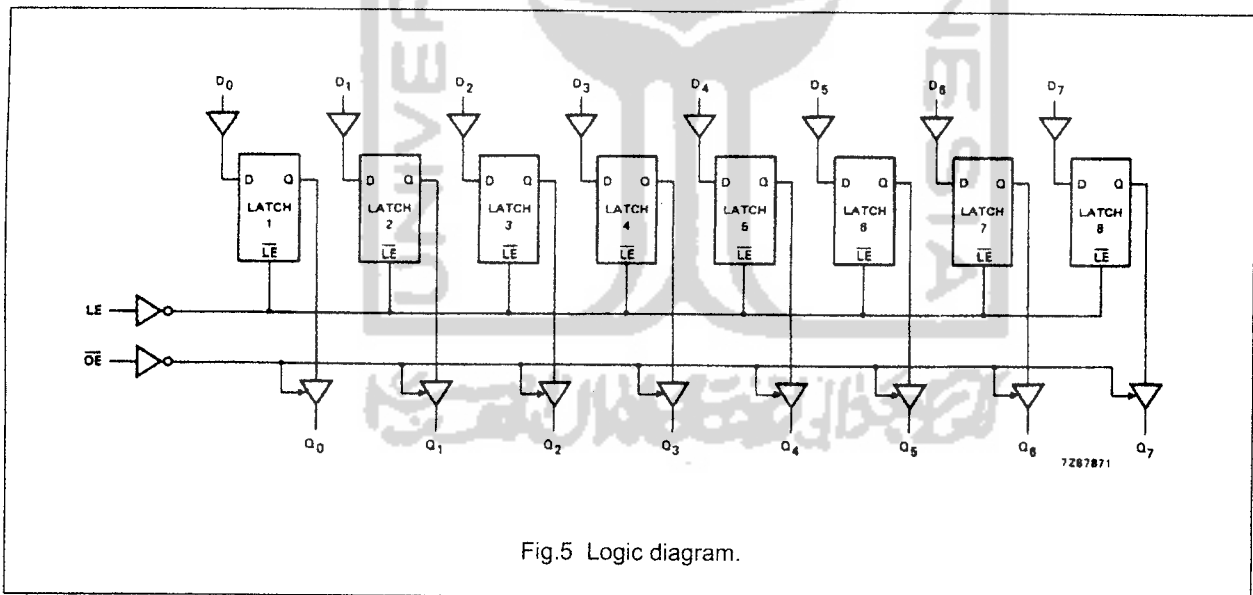


Fig.5 Logic diagram.

Octal D-type transparent latch; 3-state

74HC/HCT573

DC CHARACTERISTICS FOR 74HC

For the DC characteristics see "74HC/HCT/HCU/HCMOS Logic Family Specifications".

Output capability: bus driver

I_{CC} category: MSI

AC CHARACTERISTICS FOR 74HC

GND = 0 V; t_r = t_f = 6 ns; C_L = 50 pF

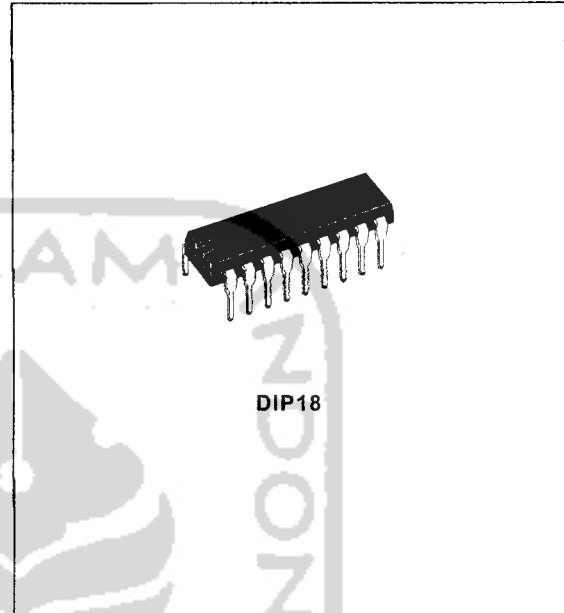
SYMBOL	PARAMETER	T _{amb} (°C)								UNIT	TEST CONDITIONS	
		74HC									V _{CC} (V)	WAVEFORMS
		+25			-40 to +85		-40 to +125					
		min.	typ.	max.	min.	max.	min.	max.				
t _{PHL} / t _{PLH}	propagation delay D _n to Q _n	47	150		190		225	ns	2.0	Fig.6		
		17	30		38		45					
		14	26		33		38					
t _{PHL} / t _{PLH}	propagation delay LE to Q _n	50	150		190		225	ns	2.0	Fig.7		
		18	30		38		45					
		14	26		33		38					
t _{PZH} / t _{PZL}	3-state output enable time \overline{OE} to Q _n	44	140		175		210	ns	2.0	Fig.8		
		16	28		35		42					
		13	24		30		36					
t _{PHZ} / t _{PLZ}	3-state output disable time \overline{OE} to Q _n	55	150		190		225	ns	2.0	Fig.8		
		20	30		38		45					
		16	26		33		38					
t _{THL} / t _{TLH}	output transition time	14	60		75		90	ns	2.0	Fig.6		
		5	12		15		18					
		4	10		13		15					
t _w	enable pulse width HIGH	80	14		100		120	ns	2.0	Fig.7		
		16	5		20		24					
		14	4		17		20					
t _{su}	set-up time D _n to LE	50	11		65		75	ns	2.0	Fig.9		
		10	4		13		15					
		9	3		11		13					
t _h	hold time D _n to LE	5	3		5		5	ns	2.0	Fig.9		
		5	1		5		5					
		5	1		5		5					



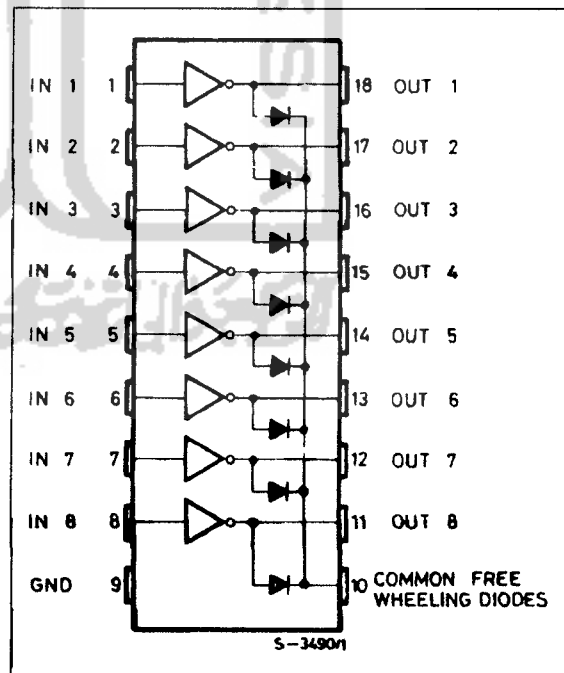
ULN2801A
ULN2802A - ULN2803A
ULN2804A - ULN2805A

EIGHT DARLINGTON ARRAYS

- EIGHT DARLINGTONS WITH COMMON EMITTERS
- OUTPUT CURRENT TO 500 mA
- OUTPUT VOLTAGE TO 50 V
- INTEGRAL SUPPRESSION DIODES
- VERSIONS FOR ALL POPULAR LOGIC FAMILIES
- OUTPUT CAN BE PARALLELED
- INPUTS PINNED OPPOSITE OUTPUTS TO SIMPLIFY BOARD LAYOUT



PIN CONNECTION (top view)



DESCRIPTION

The ULN2801A-ULN2805A each contain eight darlington transistors with common emitters and integral suppression diodes for inductive loads. Each darlington features a peak load current rating of 600mA (500mA continuous) and can withstand at least 50V in the off state. Outputs may be paralleled for higher current capability.

Five versions are available to simplify interfacing to standard logic families : the ULN2801A is designed for general purpose applications with a current limit resistor ; the ULN2802A has a 10.5k Ω input resistor and zener for 14-25V PMOS ; the ULN2803A has a 2.7k Ω input resistor for 5V TTL and CMOS ; the ULN2804A has a 10.5k Ω input resistor for 6-15V CMOS and the ULN2805A is designed to sink a minimum of 350mA for standard and Schottky TTL where higher output current is required.

All types are supplied in a 18-lead plastic DIP with a copper lead from and feature the convenient input-opposite-output pinout to simplify board layout.

ULN2801A - ULN2802A - ULN2803A - ULN2804A - ULN2805A

ABSOLUTE MAXIMUM RATINGS

Symbol	Parameter	Value	Unit
V_o	Output Voltage	50	V
V_i	Input Voltage for ULN2802A, UL2803A, ULN2804A for ULN2805A	30 15	V
I_C	Continuous Collector Current	500	mA
I_B	Continuous Base Current	25	mA
P_{tot}	Power Dissipation (one Darlington pair) (total package)	1.0 2.25	W
T_{amb}	Operating Ambient Temperature Range	- 20 to 85	°C
T_{stg}	Storage Temperature Range	- 55 to 150	°C
T_j	Junction Temperature Range	- 20 to 150	°C

THERMAL DATA

Symbol	Parameter	Value	Unit
$R_{th\ j-amb}$	Thermal Resistance Junction-ambient Max.	55	°C/W

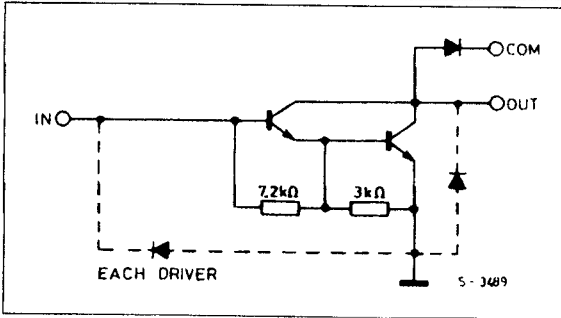
ELECTRICAL CHARACTERISTICS ($T_{amb} = 25^\circ\text{C}$ unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit	Fig.
I_{CEX}	Output Leakage Current	$V_{CE} = 50\text{V}$ $T_{amb} = 70^\circ\text{C}$, $V_{CE} = 50\text{V}$ $T_{amb} = 70^\circ\text{C}$ for ULN2802A			50 100	μA μA	1a 1a
		$V_{CE} = 50\text{V}$, $V_i = 6\text{V}$ for ULN2804A			500	μA	1b
		$V_{CE} = 50\text{V}$, $V_i = 1\text{V}$			500	μA	1b
$V_{CE(sat)}$	Collector-emitter Saturation Voltage	$I_C = 100\text{mA}$, $I_B = 250\mu\text{A}$		0.9	1.1	V	2
		$I_C = 200\text{mA}$, $I_B = 350\mu\text{A}$		1.1	1.3	V	
		$I_C = 350\text{mA}$, $I_B = 500\mu\text{A}$		1.3	1.6	V	
$I_{i(on)}$	Input Current	for ULN2802A $V_i = 17\text{V}$		0.82	1.25	mA	3
		for ULN2803A $V_i = 3.85\text{V}$		0.93	1.35	mA	
		for ULN2804A $V_i = 5\text{V}$		0.35	0.5	mA	
		for ULN2805A $V_i = 12\text{V}$		1	1.45	mA	
		for ULN2805A $V_i = 3\text{V}$		1.5	2.4	mA	
$I_{i(off)}$	Input Current	$T_{amb} = 70^\circ\text{C}$, $I_C = 500\mu\text{A}$	50	65		μA	4
$V_{i(on)}$	Input Voltage	$V_{CE} = 2\text{V}$ for ULN2802A $I_C = 300\text{mA}$			13	V	5
		for ULN2803A $I_C = 200\text{mA}$			2.4	V	
		$I_C = 200\text{mA}$			2.7	V	
		$I_C = 250\text{mA}$			3	V	
		for ULN2804A $I_C = 300\text{mA}$			5	V	
		$I_C = 125\text{mA}$			6	V	
		$I_C = 200\text{mA}$			7	V	
		$I_C = 275\text{mA}$			8	V	
		$I_C = 350\text{mA}$				V	
		for ULN2805A $I_C = 350\text{mA}$			2.4	V	
h_{FE}	DC Forward Current Gain	for ULN2801A $V_{CE} = 2\text{V}$, $I_C = 350\text{mA}$	1000			-	2
C_i	Input Capacitance			15	25	pF	-
t_{PLH}	Turn-on Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	μs	-
t_{PHL}	Turn-off Delay Time	$0.5 V_i$ to $0.5 V_o$		0.25	1	μs	-
I_R	Clamp Diode Leakage Current	$V_R = 50\text{V}$ $T_{amb} = 70^\circ\text{C}$, $V_R = 50\text{V}$			50 100	μA μA	6 6
V_F	Clamp Diode Forward Voltage	$I_F = 350\text{mA}$		1.7	2	V	7

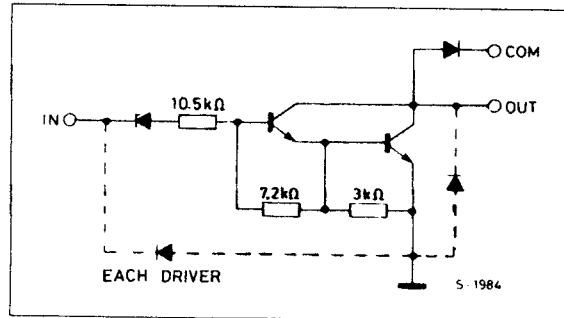


SCHEMATIC DIAGRAM AND ORDER CODES

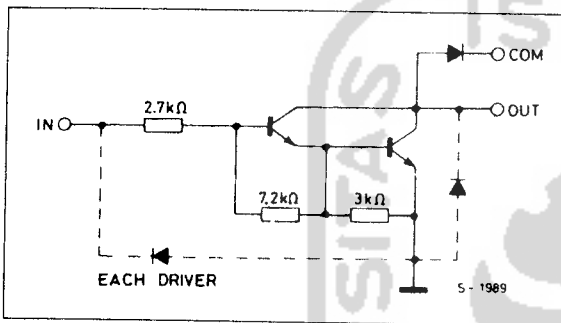
For ULN2801A (each driver for PMOS-CMOS)



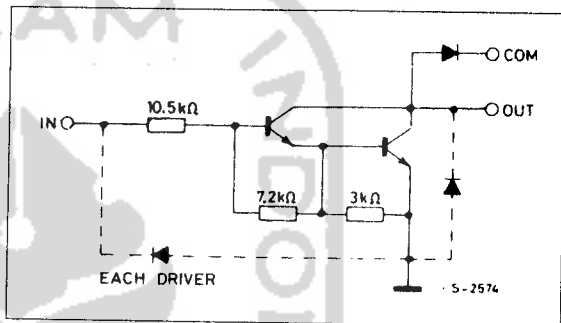
For ULN2802A (each driver for 14-15 V PMOS)



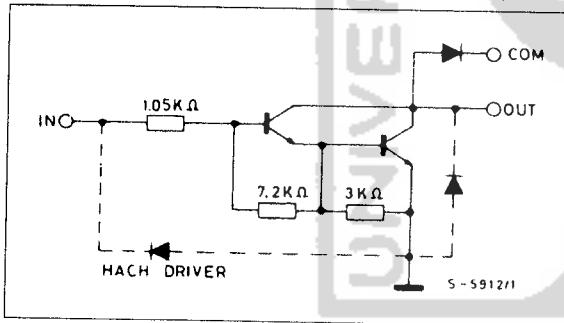
For ULN2803A (each driver for 5 V, TTL/CMOS)



For ULN2804A (each driver for 6-15 V CMOS/PMOS)



For ULN2805A (each driver for high out TTL)



Mechatronics II Laboratory Exercise 7

STEPPER MOTORS

In this laboratory exercise, you will look at several properties of stepper motors. The motor used is a two-phase motor. You will look at the pulse pattern of the motor, the holding torque and the stall frequency of the motor. There is no lab report required for this exercise, but you are expected to come to your scheduled lab and answer all the questions to your T.A.

Brief Background

The term "stepper motor" is a short descriptive name for a wound synchronous motor. These motors operate on the principle of a rotating magnetic field. A permanently magnetized rotor sits within an envelope of stator windings, as seen in Figure 1. When current is introduced to the windings, a magnetic field in the windings results because of Faraday's law. The poles of the rotor line up with the induced magnetic field (the equilibrium point or position of lowest potential energy), resulting in a net torque on the rotor. It would not be very interesting or useful to have the rotor go to one position and stay there when the current is switched on. To gain usefulness, current is switched between the windings in a circular pattern, thus causing a rotating magnetic field. The rotor chases the magnetic field resulting in a net speed. Most stepper motors have two windings, but some have three. The stepper motor of Figure 1 does not have a very desirable resolution since it has only one pole pair on the rotor. In reality, the rotors of stepper motors contain many pole pairs, thus improving the resolution (step size) according to the following relation:

$$\Delta\theta = \frac{360^\circ}{m \cdot N}$$

where $\Delta\theta$ is the resolution, m is the number of stator phases, and N is the number of rotor teeth.

This formula works for full-step configuration. The half-step configuration doubles the resolution. Note that each pole pair has two teeth. Thus, for the figure below, $m=2$ and $N=2$. The class web page has a link to an animated tutorial of these principles in the "Lab Handouts" section.

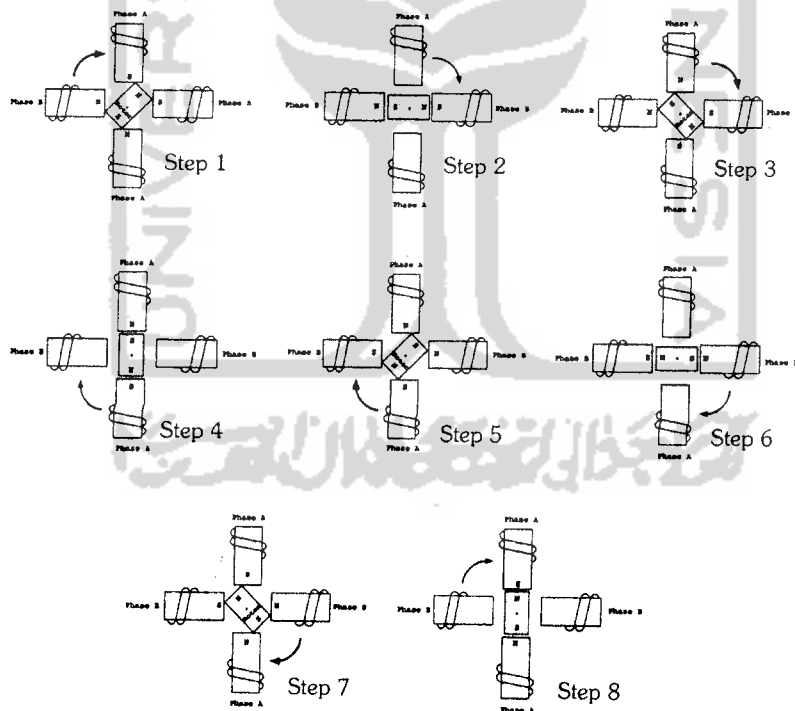


Figure 1: Schematic of a simple wound synchronous motor taking half steps.



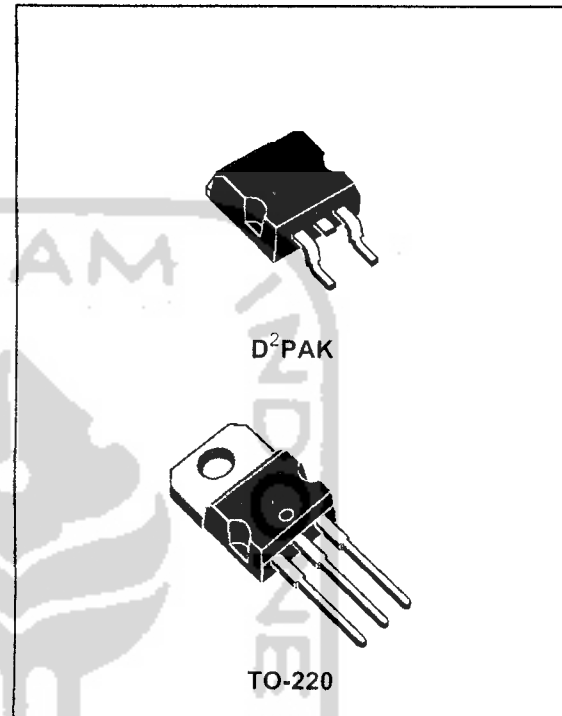
L7800AB/AC SERIES

PRECISION 1A REGULATORS

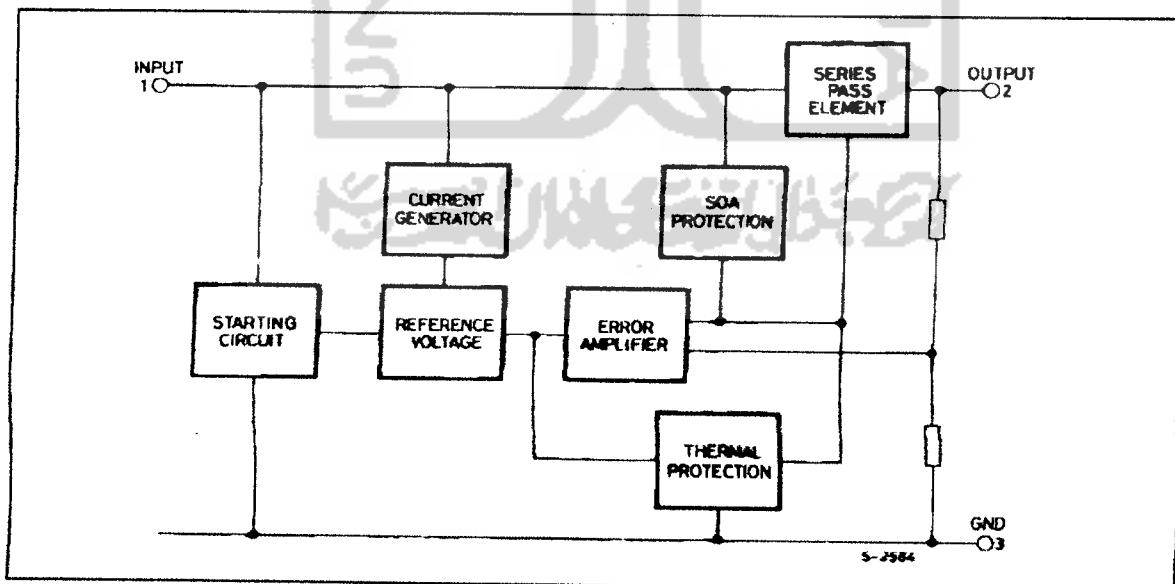
- OUTPUT CURRENT IN EXCESS OF 1 A
- OUTPUT VOLTAGES OF 5; 6; 8; 9; 12; 15; 18; 20; 24V
- THERMAL OVERLOAD PROTECTION
- OUTPUT TRANSITION SOA PROTECTION
- 2% OUTPUT VOLTAGE TOLERANCE
- GUARANTEED IN EXTENDED TEMPERATURE RANGE

DESCRIPTION

The L7800A series of three-terminal positive regulators is available in TO-220 and D²PAK packages and several fixed output voltages, making it useful in a wide range of applications. These regulators can provide local on-card regulation, eliminating the distribution problems associated with single point regulation. Each type employs internal current limiting, thermal shut-down and safe area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



BLOCK DIAGRAM



NPN general purpose transistors

BC546; BC547

FEATURES

- Low current (max. 100 mA)
- Low voltage (max. 65 V).

APPLICATIONS

- General purpose switching and amplification.

DESCRIPTION

NPN transistor in a TO-92; SOT54 plastic package.
PNP complements: BC556 and BC557.

PINNING

PIN	DESCRIPTION
1	emitter
2	base
3	collector

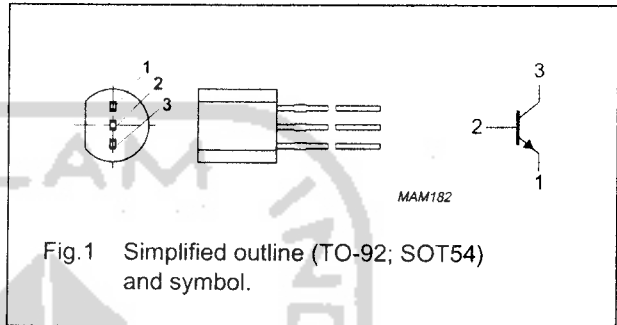


Fig.1 Simplified outline (TO-92; SOT54) and symbol.

LIMITING VALUES

In accordance with the Absolute Maximum Rating System (IEC 134).

SYMBOL	PARAMETER	CONDITIONS	MIN.	MAX.	UNIT
V _{CBO}	collector-base voltage	open emitter			
	BC546		-	80	V
	BC547		-	50	V
V _{CEO}	collector-emitter voltage	open base			
	BC546		-	65	V
	BC547		-	45	V
V _{EBO}	emitter-base voltage	open collector			
	BC546		-	6	V
	BC547		-	6	V
I _C	collector current (DC)		-	100	mA
I _{CM}	peak collector current		-	200	mA
I _{BM}	peak base current		-	200	mA
P _{tot}	total power dissipation	T _{amb} ≤ 25 °C; note 1	-	500	mW
T _{stg}	storage temperature		-65	+150	°C
T _j	junction temperature		-	150	°C
T _{amb}	operating ambient temperature		-65	+150	°C

Note

1. Transistor mounted on an FR4 printed-circuit board.