



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN KENDALI MODUL BERODA DENGAN  
REMOTE DTMF**

**SKRIPSI**

**MUHAMMAD FACHRI  
0906602906**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2012**



**UNIVERSITAS INDONESIA**

**RANCANG BANGUN KENDALI MODUL BERODA DENGAN  
REMOTE DTMF**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar sarjana teknik**

**MUHAMMAD FACHRI  
0906602906**

**FAKULTAS TEKNIK  
PROGRAM STUDI TEKNIK ELEKTRO  
DEPOK  
JULI 2012**

## HALAMAN PERNYATAAN ORISINALITAS

Skripsi ini adalah hasil karya sendiri,  
dan semua sumber baik yang dikutip maupun dirujuk  
telah saya nyatakan dengan benar

Nama : Muhammad Fachri

NPM : 0906602906

Tanda Tangan : 

Tanggal : 5 Juli 2012

## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh,

Nama : Muhammad Fachri

NPM : 0906602906

Program Studi : Teknik Elektro

Judul Skripsi : Rancang Bangun Kendali Modul Beroda dengan Remote DTMF

Telah berhasil dipertahankan dihadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Teknik pada Program Studi Teknik Elektro, Fakultas Teknik, Universitas Indonesia

### DEWAN PENGUJI

Pembimbing : Dr. Ir. Arman Djohan Diponegoro

()

Penguji : Prof. Dr. Ir. Harry Sudibyo DEA

()

Penguji : Ir. Purnomo Sidi Priambodo M.Sc., Ph.D.

()

Ditetapkan di : Depok

Tanggal : 5 Juli 2012

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas berkat dan rahmat-Nya sehingga skripsi ini dapat diselesaikan dengan baik. Tak lupa penulis juga mengucapkan terima kasih kepada :

Dr. Ir. Arman Djohan Diponegoro

Selaku dosen pembimbing yang telah meluangkan waktu untuk memberikan saran, bimbingan, pengarahan, dan lainnya dalam penyelesaian skripsi ini. Juga kepada Orang tua, rekan-rekan semua yang tidak dapat disebutkan satu-persatu.

Depok, 5 Juli 2012  
Penulis

Muhammad Fachri  
NPM. 0906602906

**HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI  
TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Indonesia, saya yang bertanda tangan dibawah ini :

Nama : Muhammad Fachri  
NPM : 0906602906  
Program Studi : Teknik Elektro  
Departemen : Teknik Elektro  
Fakultas : Teknik  
Jenis Karya : Skripsi

demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Indonesia **Hak Bebas Royalti Noneksklusif** (*Non-exclusive Royalty-Free Right*) atas karya ilmiah saya yang berjudul :

**Rancang Bangun Kendali Modul Beroda dengan Remote DTMF**

beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneksklusif ini Universitas Indonesia berhak menyimpan, mengalihmedia / format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Depok  
Pada Tanggal : 5 Juli 2012

Yang Menyatakan



(Muhammad Fachri)

## ABSTRAK

Nama : Muhammad Fachri  
Program Studi : Teknik Elektro  
Judul : Rancang Bangun Kendali Modul Beroda Dengan Remote DTMF

Skripsi ini membahas mengenai rancang bangun kendali modul beroda nirkabel dengan menggunakan teknik DTMF (*Dual Tone Multy Frequency*). Dalam sistem ini, modul beroda dikendalikan dengan sebuah modul *remote* yang terdiri dari 4 tombol. *Remote* mengaplikasikan teknik DTMF yang umumnya digunakan pada pesawat telepon. DTMF mengubah kombinasi *array* kolom dan baris menjadi dua frekuensi yang berbeda, dan kemudian frekuensi tersebut dikonversi menjadi data digital. Modul beroda dikendalikan secara jarak jauh atau nirkabel. Sistem ini memiliki dua buah mikrokontroler, masing-masing pada sisi modul *remote* dan modul beroda. Penggerak modul beroda menggunakan dua buah motor DC. Metode komunikasi yang dilakukan antara *transmitter* dan *receiver* adalah komunikasi *simplex*. Komunikasi antara *transmitter* dengan *receiver* bekerja pada frekuensi yang sama yaitu 433,92 MHz.

Kata kunci : *Remote DTMF, Nirkabel, Mikrokontroler*

## ABSTRACT

Name : Muhammad Fachri  
Study Program: Electrical Engineering  
Title : Design and Implementation of Wheels Module Control Using DTMF Remote

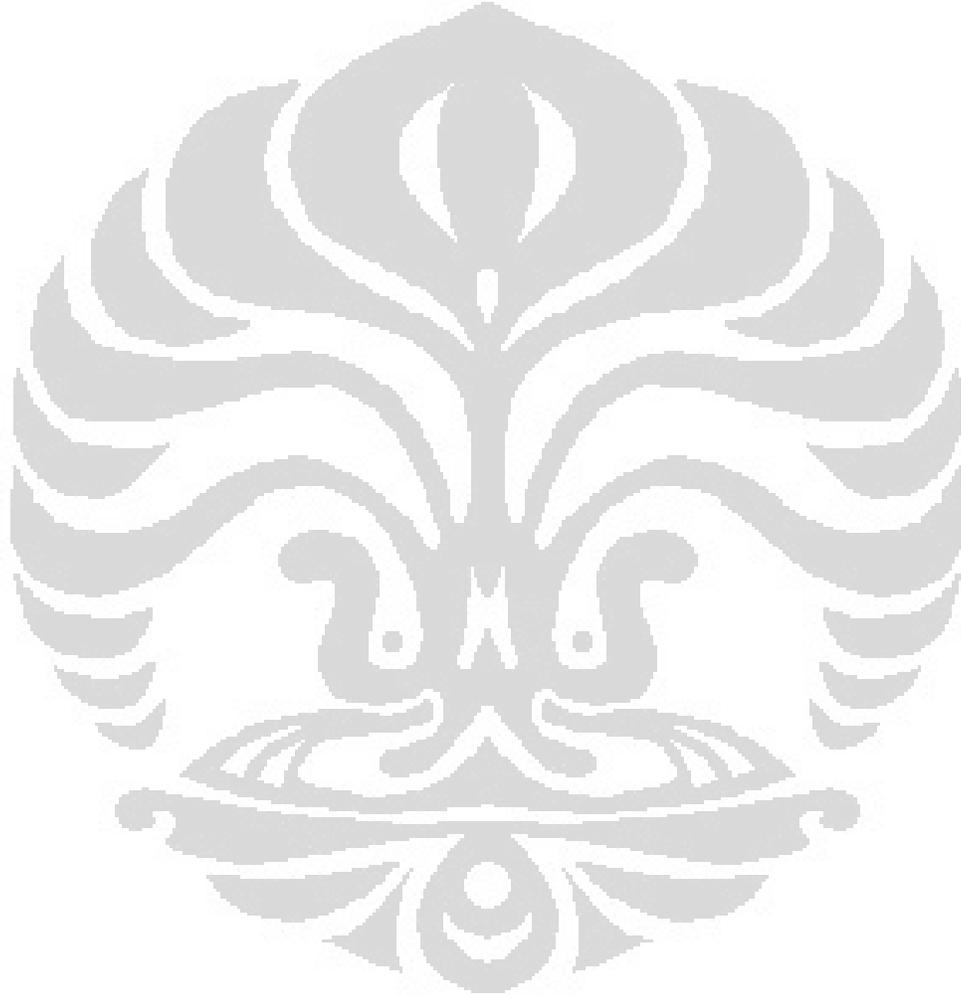
This paper discusses about the design and implementation of wheels modul with DTMF remote. In this system, wheels module controlled by remote module. Remote module using DTMF techniques. DTMF combine array of row and column change into two different frquency, and then covert frequency to digital data. In this system, wheels modul controlled by remote with four push button. Remote use DTMF technic which commonly used in telephone device. Wheels modul controlled by wireless. This system has two microcontroller, in remote modul and wheels modul. Actuator of wheels modul are couple of dc motor. Communication method between transmitter and receiver is simplex. Communication between transmitter and receiver operate on the same frequency is 433,92 MHz.

Keyword : *Remote DTMF, Wireless, Microcontroller*

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PERNYATAAN ORSINALITAS.....	ii
HALAMAN PENGESAHAN.....	iii
KATA PENGANTAR.....	iv
HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI.....	v
ABSTRAK.....	vi
DAFTAR ISI.....	vii
DAFTAR GAMBAR.....	ix
DAFTAR TABEL.....	x
<b>BAB 1 PENDAHULUAN.....</b>	<b>1</b>
1.1. Latar Belakang.....	1
1.2. Tujuan.....	2
1.3. Batasan Masalah.....	2
1.4. Sistematika Penulisan.....	2
<b>BAB 2 TINJAUAN PUSTAKA.....</b>	<b>4</b>
2.1. DTMF ( <i>Dual Tone Multy Frequency</i> ).....	4
2.1.1. <i>Encoder</i> .....	5
2.1.2. <i>Decoder</i> .....	8
2.2. Mikrokontroler.....	10
2.2.1. Konfigurasi Pin ATMega 8.....	11
2.2.2. Peta Memori ATMega 8.....	13
2.3. Komunikasi Data Serial.....	15
2.3.1. Komunikasi Data Serial USART.....	18
2.3.2. Inisialisasi USART.....	19
2.4. <i>Transmitter</i> TLP-434.....	24
2.3.1. Konfigurasi Pin <i>Transmitter</i> TLP-434.....	24
2.3.2. Karakteristik <i>Transmitter</i> TLP-434.....	25
2.5. <i>Receiver</i> RLP-434.....	25
2.5.1. Konfigurasi Pin <i>Receiver</i> RLP-434.....	26
2.5.2. Karakteristik <i>Receiver</i> RLP-434.....	27
2.6. Aktuator.....	27
2.5.1. Motor DC.....	27
2.5.2. <i>Driver</i> Motor DC.....	28
<b>BAB 3 PERANCANGAN ALAT.....</b>	<b>30</b>
3.1. Deskripsi Sistem.....	30
3.2. Perancangan Modul <i>Remote</i> .....	32
3.3. Perancangan Modul Modul beroda.....	35
3.4. Perancangan Algoritma Sistem.....	35
3.4. Perancangan <i>Driver</i> Motor.....	37
<b>BAB 4 PENGUJIAN DAN ANALISA DATA.....</b>	<b>38</b>
4.1. Deskripsi Pengujian.....	38

4.2. Pengujian Penekanan Tombol <i>Remote</i> Bersamaan .....	38
4.3. Pengukuran Data yang Dikirim <i>Transmitter</i> .....	40
4.4. Pengukuran Frekuensi Kerja Sebuah <i>Transmitter</i> .....	44
4.5. Pengukuran JangkauanMaksimum Sebuah <i>Transmitter</i> .....	46
<b>BAB 5 KESIMPULAN DAN SARAN</b> .....	47
5.1. Kesimpulan .....	47
<b>DAFTAR ACUAN</b> .....	55
<b>LAMPIRAN</b> .....	56

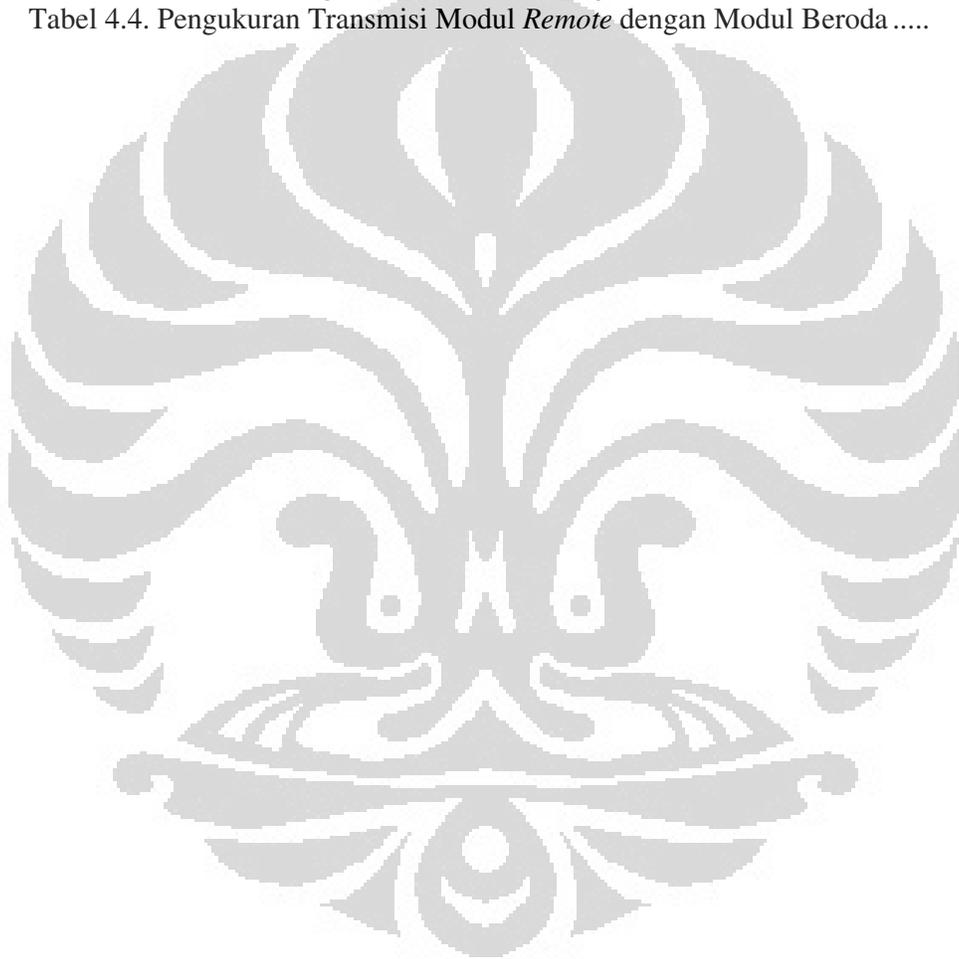


## DAFTAR GAMBAR

Gambar 2.1. Kombinasi Nada DTMF .....	5
Gambar 2.2. Blok diagram <i>Pulse/Tone Dialer</i> UM91214/15 <i>Series</i> .....	6
Gambar 2.3. Susunan Pin IC UM91215C .....	7
Gambar 2.4. Blok diagram CM8870PI .....	8
Gambar 2.5. Konfigurasi Pin IC CM8870PI .....	9
Gambar 2.6. Konfigurasi Pin ATMEGA 8 .....	13
Gambar 2.7. Blok Diagram ATMEGA 8 .....	13
Gambar 2.8. Peta Memori ATMEGA 8 .....	15
Gambar 2.9. Paket Data Komunikasi Serial Asinkron .....	16
Gambar 2.10. Diagram Blok USART .....	18
Gambar 2.11. <i>Register</i> UDR .....	19
Gambar 2.12. <i>Register</i> UCSRA .....	20
Gambar 2.13. <i>Register</i> UCSRB .....	21
Gambar 2.14. <i>Register</i> UCSRC .....	23
Gambar 2.15. <i>Transmitter</i> TLP-434 .....	24
Gambar 2.16. <i>Receiver</i> RLP-434 .....	25
Gambar 2.17. Motor DC .....	27
Gambar 2.18. Diagram Blok IC L298 .....	29
Gambar 3.1. Diagram Blok Modul <i>Remote</i> .....	31
Gambar 3.2. Diagram Blok Modul beroda .....	31
Gambar 3.3. Rangkaian Modul <i>Remote</i> .....	33
Gambar 3.4. Modul <i>Remote</i> .....	33
Gambar 3.5. Rangkaian Modul Beroda .....	35
Gambar 3.6. Modul Beroda .....	37
Gambar 3.7. Diagram Alir Algoritma Sistem .....	38
Gambar 4.1. Konfigurasi Pengujian Penekan Tombol .....	39
Gambar 4.2. Konfigurasi Pengujian <i>Transmitter</i> .....	41
Gambar 4.3. Paket Data Serial .....	42
Gambar 4.4. Data Serial pada Modul <i>Remote</i> .....	42
Gambar 4.5. Konfigurasi Pengukuran Frekuensi Kerja Sebuah Transmitter .....	44
Gambar 4.6. Data Hasil Frekuensi Kerja <i>Transmitter</i> pada Osiloskop .....	42

## DAFTAR TABEL

Tabel 2.1. Penentuan Ukuran Karakter .....	22
Tabel 3.1. Konfigurasi <i>Push-Button</i> Modul <i>Remote</i> .....	33
Tabel 3.2. Konfigurasi Pin UM 91215 dengan <i>Push-Button</i> .....	33
Tabel 3.3. Frekuensi <i>Output Encoder</i> UM 91215 .....	34
Tabel 3.4. Hasil Pengkodean <i>Decoder</i> M8870 .....	34
Tabel 4.1. Hasil Pengujian Penekanan Tombol .....	39
Tabel 4.2. Peralatan Pendukung Pengujian <i>Transmitter</i> .....	41
Tabel 4.3. Peralatan Pengukuran Frekuensi Kerja Sebuah <i>Transmitter</i> .....	45
Tabel 4.4. Pengukuran Transmisi Modul <i>Remote</i> dengan Modul Beroda .....	46



## BAB 1 PENDAHULUAN

### 1.1. Latar Belakang

Kemajuan teknologi elektronika memberikan kemudahan dan kemajuan dalam proses telekomunikasi atau pengiriman informasi, dan bidang otomatisasi. Proses pengiriman informasi dalam dunia telekomunikasi yang semula menggunakan media perantara (kabel), sekarang berkembang menuju komunikasi data tanpa kabel (nirkabel). Dengan teknologi nirkabel atau *wireless*, instalasi menjadi lebih praktis karena tidak memerlukan kabel-kabel yang menghubungkan divais satu ke divais yang lain. Teknologi nirkabel cukup menggunakan pemancar (transmitter) dan penerima (receiver), tidak memerlukan media perantara kabel. Adanya teknologi nirkabel, didapatkan kemudahan mengakses sebuah sistem elektronika dalam memantau ataupun mengendalikan sistem tersebut pada tempat yang jauh dan sulit dijangkau. Untuk mengaktifkan sebuah sistem atau yang semula menggunakan kabel, kini mampu diaktifkan menggunakan teknologi *wireless* sehingga tidak memerlukan instalasi kabel antara *transmitter* dengan *receiver* serta menghemat biaya kabel dan biaya instalasi.

Dalam bidang telekomunikasi sinyal informasi yang berupa *tone*, dapat ditransmisikan dengan teknologi *wireless*. Aplikasi sinyal *tone* terdapat pada sistem elektronik pada pesawat telepon. Dalam divais telepon terdapat sebuah fenomena DTMF (*Dual Tone Multy Frequency*), yaitu penggabungan beberapa *tone* yang menghasilkan keluaran frekuensi yang berbeda. Serta dalam bidang otomatisasi, pengembangan divais elektronika semakin pesat menuju ke pembuatan robot yang dapat memudahkan pekerjaan manusia. Dengan memadukan teknologi *wireless* dan penggabungan *tone* tombol-tombol, dapat menciptakan sebuah sistem modul beroda sebagai model dari robot yang diaktifkan dengan menggunakan penggabungan *tone*. Dengan kata lain, teknologi ini menciptakan sebuah modul beroda yang menggunakan *remote wireless* dan berbasis DTMF.

Dengan menggabungkan DTMF, teknologi wireless, dan mikrokontroler dapat diaplikasikan sebuah *wireless remote control* berbasis DTMF untuk mengaktifkan modul beroda. Modul beroda diaktifkan dengan empat keadaan, yaitu maju, mundur, belok kanan, dan belok kiri.. Modul beroda diaktifkan menggunakan *remote wireless* yang berbasis DTMF, dengan *modul remote* memiliki rangkaian *transmitter*. Rangkaian *transmitter* pada *modul remote* berfungsi untuk mengirimkan data atau mengaktifkan modul beroda, sedangkan rangkaian *receiver* pada modul beroda berfungsi untuk menerima data dari *modul remote*. Modul *transmitter wireless* dan *receiver wireless* terintegrasi dengan mikrokontroler sebagai pengolah data. Frekuensi kerja yang digunakan pada modul remote dan modul beroda adalah sama, yaitu 433,92 MHz. Proses pengiriman atau komunikasi data antara *transmitter* dengan *receiver* menggunakan komunikasi serial asinkron dan berlangsung secara *single duplex*.

## **1.2. Tujuan**

Skripsi ini bertujuan untuk membuat modul beroda dengan kendali jarak jauh nirkabel menggunakan *remote* berbasis *Dual Tone Multy Frequency*. Komunikasi yang dilakukan antara *transmitter* dan *receiver* menggunakan *wireless*. *Transmitter* dan *receiver* bekerja pada frekuensi yang sama yaitu 433,92 MHz.

## **1.3. Batasan Masalah**

Dalam laporan skripsi ini, permasalahan dibatasi pada pengolahan *Dual Tone Multy Frequency*, metode komunikasi yang digunakan antara *transmitter* dan *receiver*, dan protokol komunikasi yang digunakan.

## **1.4. Sistematika Penulisan**

Penulisan skripsi ini disusun berdasarkan sistematika penulisan sebagai berikut :

Bab pertama membahas pendahuluan yang berkenaan dengan latar belakang masalah, tujuan, batasan masalah, dan sistematika penulisan. Bab kedua menjelaskan mengenai tinjauan pustaka berupa bahan-bahan materi yang berhubungan dan mendukung dalam penulisan skripsi ini. Bab ketiga menjelaskan perancangan sistem yang akan dibuat dalam tugas akhir berupa deskripsi sistem, tujuan sistem, spesifikasi sistem, perancangan modul *remote* dan modul beroda, perancangan rangkaian *driver*. Bab keempat menjelaskan hasil dan analisa alat. Terakhir pada bab kelima menerangkan kesimpulan dan saran.



## BAB 2

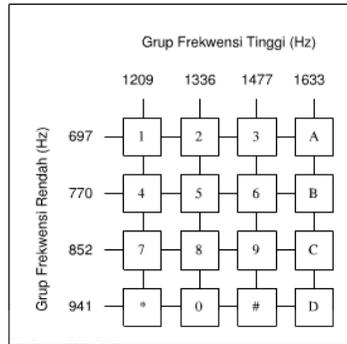
### TINJAUAN PUSTAKA

#### 2.1. DTMF (*Dual Tone Multy Frequency*)

*Dual Tone Muty Frequency* merupakan sebuah proses pengolahan data digital menjadi data analog, dan kemudian terjadi pengolahan kembali dari data analog menjadi data digital, dengan demikian DTMF terdapat dua proses pengolahan yaitu *encoder* dan *decoder*.

DTMF adalah suatu teknik mengirimkan angka-angka pembentuk nomor telepon yang dikodekan dengan 2 nada yang dipilih dari 8 buah frekuensi yang sudah ditentukan. Delapan frekuensi tersebut adalah 697 Hz, 770 Hz, 852 Hz, 941 Hz, 1209 Hz, 1336 Hz, 1477 Hz, dan 1633 Hz. Frekuensi-frekuensi tersebut dibagi ke dalam 2 grup, yaitu grup frekuensi baris/*Low Group* (697, 770, 852, 941 Hz) dan grup frekuensi kolom/*High Group* (1209, 1336, 1477, 1633 Hz). Seperti terlihat dalam gambar 2.1 angka 1 dikodekan dengan frekuensi 697 Hz dan 1209 Hz. Kombinasi dari 8 frekuensi tersebut bisa dipakai untuk mengkodekan 16 tanda.

Pada pesawat telepon biasanya tombol A, B, C, dan D tidak dipakai. Sedangkan simbol “\*” biasanya disebut star atau asterisk dan simbol “#” disebut pagar. Walaupun banyak pemakai telepon tidak mempergunakan digit ini, tetapi biasanya dalam *dialing* nomor telepon kedua simbol tersebut digunakan untuk keperluan pengontrol seperti pada mesin penjawab telepon, *control repeater*, *electronic banking* dan lain-lain. Standar *dial* DTMF adalah saat dimana nada selama tombol telepon ditekan, tak peduli berapa lamanya, nada dikodekan sebagai satu digit. Pengiriman digit dalam durasi yang pendek  $\pm 100$  ms dapat juga dilakukan, tapi tidak oleh tangan manusia karena hal tersebut tidak mungkin, dan hanya dilakukan dengan cara otomatis.



**Gambar 2.1.** Kombinasi Nada DTMF

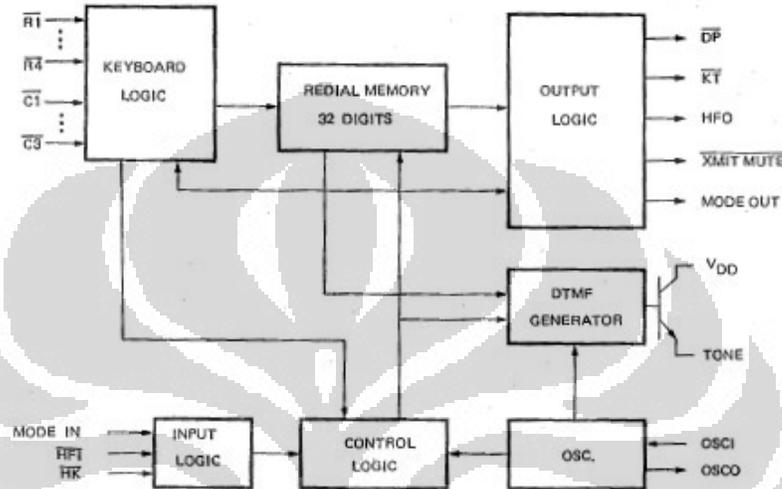
Pada teknik DTMF meskipun mempunyai banyak keunggulan dibanding dengan cara memutar piringan angka, tapi secara teknis lebih sulit diselesaikan. Alat pengirim kode DTMF merupakan 8 rangkaian osilator yang masing-masing membangkitkan frekuensi tinggi, ditambah dengan rangkaian pencampur frekuensi untuk mengirimkan 2 nada yang terpilih. Sedangkan penerima kode DTMF lebih rumit lagi, dibentuk dari 8 buah filter yang tidak sederhana dan rangkaian tambahan lainnya.

DTMF juga digunakan pada penyiaran TV kabel untuk mengindikasikan waktu mulai dan akhir iklan komersial. Adanya sistem DTMF ini memungkinkan terjadinya hubungan komunikasi baik digunakan untuk jaringan telepon ataupun dapat digunakan untuk komunikasi khusus lainnya. Oleh karena kegunaan dari DTMF yang mudah untuk diaplikasikan, maka saat ini telah banyak dibuat *Integrated Circuit (IC)* untuk kebutuhan DTMF baik untuk *encode*, *decode*, ataupun keduanya.

### 2.1.1. Encoder DTMF

DTMF dapat dimanfaatkan untuk sistem kontrol jarak jauh, dimana DTMF digunakan sebagai informasi utama untuk membedakan sinyal yang dikirimkan. Diperlukan dua buah sistem yaitu sistem pemancar DTMF dan penerima DTMF. Sinyal DTMF dapat dihasilkan dari sebuah IC DTMF *encoder* yang sudah banyak dijual di pasaran. Sinyal DTMF yang dihasilkan merupakan hasil dari pencampuran 2 buah frekuensi yaitu dari pin frekuensi baris dan pin frekuensi kolom yang ditekan atau dipilih.

IC UM91215C merupakan salah satu jenis dari IC DTMF *encoder*. UM91215C adalah jenis dari *Tone/Pulse Dialer UM91214/15 Series* keluaran Unicorn Microelectronics (UMC). Dengan melihat blok diagram rangkaian *Pulse/Tone Dialer UM91214/15 Series* pada gambar 2.2, dapat diketahui cara kerja dari UM91215C.

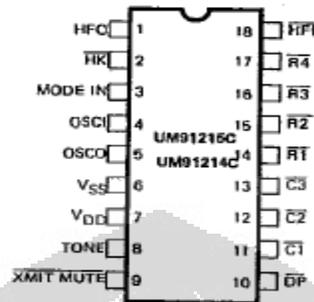


**Gambar 2.2.** Blok diagram *Pulse/Tone Dialer UM91214/15 Series*

Berdasarkan blok diagram di atas, terdapat dua bagian yang berfungsi sebagai input, yaitu *keyboard logic* dan *input logic*. *Keyboard logic* biasanya menggunakan keypad sebagai *interface*-nya. Pada bagian *keyboard logic* dihasilkan kombinasi sinyal DTMF yang akan dikirimkan yang merupakan kombinasi dari frekuensi-frekuensi baris (*Low Group Frequency*) dan frekuensi-frekuensi kolom (*High Group Frequency*). IC UM91215C ini dapat dihubungkan dengan keypad matriks 4x4 sehingga dapat menghasilkan 16 sinyal DTMF.

Sedangkan pada *input logic* terdapat parameter yang harus diberikan nilai tertentu (aktif *low/high*) agar IC dapat bekerja sesuai dengan kondisi yang diinginkan. Parameter kerja untuk IC UM91215C adalah *MODE IN*,  $\overline{HK}$ , dan  $\overline{HFI}$  yang akan menentukan mode kerja dari IC sesuai dengan logic yang diberikan. Selanjutnya terdapat bagian *control logic* yang mengontrol seluruh masukan. Kemudian bagian *DTMF generator* yang menghasilkan sinyal DTMF. Terdapat pula bagian *oscillator* yang berupa

osilator internal dan membutuhkan kristal 3,58 Mhz sebagai input dan output osilator.



**Gambar 2.3.** Susunan Pin IC UM91215C

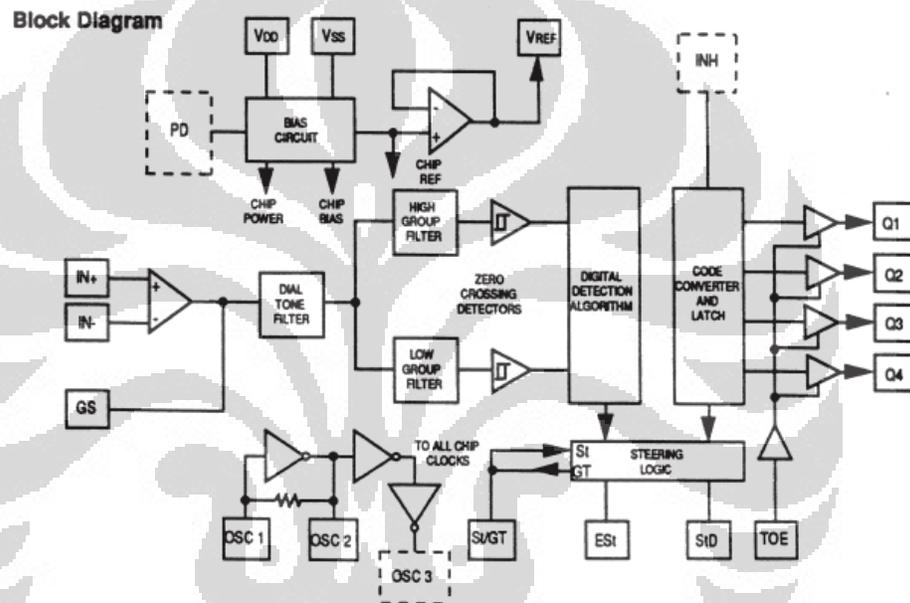
Susunan pin IC UM91215C ditunjukkan pada Gambar 2.3 di atas. Penjelasan dari pin-pin UM91215C adalah sebagai berikut :

- Pin  $\overline{HK}$  (*Hook switch input*) mendeteksi keadaan *switch*. Apabila dalam keadaan “*On Hook*”, maka dikondisikan pada kondisi  $V_{DD}$ , sedangkan keadaan “*Off Hook*” dinyatakan dengan kondisi  $V_{SS}$ .
- Pin *MODE IN* (*Mode select*) merupakan pilihan untuk mode *dialing pulse/tone* pada tiap tombol yang dipilih.
- Pin *OSCO* dan *OSCI* merupakan pin masukan dan keluaran osilator (kristal 3,58 MHz).
- Pin *VSS* adalah pin yang terhubung ke *Ground*.
- Pin *VDD* adalah pin masukan +2V sampai +5V.
- Pin *TONE* (*Tone dialing output*) merupakan pin keluaran sinyal DTMF yang berasal dari *DTMF generator*.
- Pin  $\overline{XMIT MUTE}$  (*Dialing transmission mute output*) dipakai untuk operasi *mute* pada saat transmit.
- Pin  $\overline{R1}$ ,  $\overline{R2}$ ,  $\overline{R3}$ , dan  $\overline{R4}$  adalah pin-pin frekuensi baris ( *Low Group Frequency*).
- Pin  $\overline{C1}$ ,  $\overline{C2}$ , dan  $\overline{C3}$  adalah pin-pin frekuensi kolom ( *High Group Frequency* ).

- Pin  $\overline{DP}$  (*Dialing Pulse Output*) merupakan output pada saat *pulse mode*.

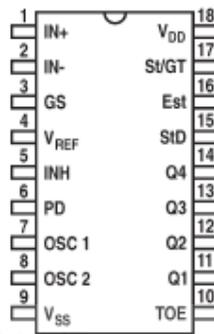
### 2.1.2. Decoder DTMF

DTMF *decoder* merupakan sebuah rangkaian yang berfungsi untuk mengkonversi sinyal DTMF yang berupa sinyal analog menjadi output data yang berbentuk digital. Salah satu IC DTMF *decoder* yang umum di pasaran adalah CM8870PI keluaran *California Micro Devices*.



Gambar 2.4. Blok diagram CM8870PI

IC CM8870PI merupakan gabungan dari *bandsplit filter* dan *digital decoder function*. Blok diagram dari CM8870PI ditunjukkan seperti pada Gambar 2.4. Tampak pada blok diagram bahwa CM8870PI berisi filter pemisah gelombang dan fungsi pengubah ke sinyal digital. Bagian filter menggunakan teknik *switched capacitor* untuk *Low Group Filter* dan *High Group Filter* serta *penolakan nada dial* (*Tone Dial Rejection*). Sedangkan untuk *decoder* menggunakan teknik *counter* digital yang akan memeriksa dan mengkodekan 16 sinyal DTMF ke dalam suatu kode 4 bit. MT8870DE juga membutuhkan kristal 3,58 Mhz sebagai input dan output untuk osilator. Susunan pin dari CM8870PI dapat dilihat pada Gambar 2.5.



**Gambar 2.5.** Konfigurasi Pin IC CM8870PI

Penjelasan pin-pin CM8870PI adalah sebagai berikut :

- Pin *IN+* : input non-inverting.
- Pin *IN-* : input inverting.
- Pin *GS (Gain Select)* : penghubung keluaran dari *differential amplifier op-amp* untuk dihubungkan dengan resistor umpan balik.
- Pin *Vref (Reference Voltage)* : output tegangan referensi biasanya sebesar  $V_{cc}/2$  dan digunakan untuk bias input *op-amp*.
- Pin *INH* : pin yang berfungsi deteksi pencegahan masuknya *tones* yang dihasilkan oleh tombol A,B,C, dan D.
- Pin *PD* : pin *Power Down*.
- Pin *OSC1* : masukan *clock* yang didapat dari osilator kristal 3,58 MHz.
- Pin *OSC2* : keluaran *clock* yang diperoleh dari osilator kristal 3,58 MHz.
- Pin *VSS* : pin yang dihubungkan ke *Ground*.
- Pin *TOE (Three State Output Enable)* : pin *enable* yang menjadi keluaran untuk pin *Q1 – Q4*.
- Pin *Q1 – Q4* : output biner (4 bit) sebagai hasil dari respon masukan sinyal DTMF.
- Pin *StD (Delayed Steering)* : output dari logik tinggi ketika merespon adanya masukan sinyal DTMF.
- Pin *Est (Early Steering)* : output logik tinggi dari setiap masukan digital DTMF.

- Pin *St/GT (Steering Input/Guard Time)* : berfungsi sebagai pengatur waktu respons masukan sinyal DTMF.
- Pin *VDD* : pin masukan untuk IC yaitu sebesar +5V

## 2.2. Mikrokontroler ATMEGA 8

AVR merupakan salah satu jenis mikrokontroler yang di dalamnya terdapat berbagai macam fungsi. Perbedaannya pada mikro yang pada umumnya digunakan seperti MCS51 adalah pada AVR tidak perlu menggunakan *oscillator* eksternal karena di dalamnya sudah terdapat internal *oscillator*. Selain itu kelebihan dari AVR adalah memiliki *Power-On Reset*, yaitu tidak perlu ada tombol reset dari luar karena cukup hanya dengan mematikan *supply*, maka secara otomatis AVR akan melakukan *reset*. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 *byte* sampai dengan 512 *byte*.

AVR ATmega 8 adalah mikrokontroler CMOS *8-bit* berarsitektur AVR RISC yang memiliki 8K *byte in-System Programmable Flash*. Mikrokontroler dengan konsumsi daya rendah ini mampu mengeksekusi instruksi dengan kecepatan maksimum 16MIPS pada frekuensi 16 MHz. Jika dibandingkan dengan ATmega 8L perbedaannya hanya terletak pada besarnya tegangan yang diperlukan untuk bekerja. Untuk ATmega8 tipe L, mikrokontroler ini dapat bekerja dengan tegangan antara 2,7 - 5,5 V sedangkan untuk ATmega 8 hanya dapat bekerja pada tegangan antara 4,5 – 5,5 V. Mikrokontroler ATmega 8 memiliki beberapa fasilitas sebagai berikut:

- Saluran I/O sebanyak 23 buah, yaitu *Port B*, *Port C* dan *Port D*.
- ADC (*Analog to Digital Converter*) 10-bit sebanyak 8 *channel*.
- Dua 8-bit *Timer/Counter*.
- CPU yang terdiri dari 32 buah register.
- SRAM sebanyak 1 Kbyte.
- Memori *Flash* sebesar 8 Kb dengan kemampuan *Read While Write*.
- Unit interupsi *internal* dan *eksternal*.
- Port* antarmuka SPI.

- i. EPROM sebesar 512 byte yang dapat diprogram.
- j. *Watchdog Timer* dengan osilator *internal*.
- k. Port *USART* untuk komunikasi serial.

### 2.2.1. Konfigurasi Pin ATMEGA 8

**PDIP**

(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADCO)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

**Gambar 2.6.** Konfigurasi Pin ATMEGA 8

ATmega 8 memiliki 28 pin seperti ditunjukkan pada gambar 2.6, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki ATmega 8.

- a. VCC  
Merupakan *supply* tegangan digital.
- b. GND  
Merupakan *ground* untuk semua komponen yang membutuhkan *grounding*.
- c. Port B (PB7 - PB0)  
Didalam Port B terdapat XTAL1, XTAL2, TOSC1, TOSC2. Jumlah Port B adalah 8 buah pin, mulai dari pin B.0 sampai dengan B.7. Tiap pin dapat digunakan sebagai *input* maupun *output*. Port B merupakan sebuah 8-bit *bi-directional I/O* dengan internal pull-up resistor. Sebagai *input*, pin-pin yang terdapat pada port B yang secara eksternal diturunkan, maka akan mengeluarkan arus jika *pull-up* resistor

diaktifkan. Khusus PB6 dapat digunakan sebagai *input* kristal (*inverting oscillator amplifier*) dan *input* ke rangkaian *clock* internal, bergantung pada pengaturan *fuse bit* yang digunakan untuk memilih sumber *clock*. Sedangkan untuk PB7 dapat digunakan sebagai *output* kristal (*output oscillator amplifier*) bergantung pada pengaturan *Fuse bit* yang digunakan untuk memilih sumber *clock*. Jika sumber *clock* yang dipilih dari *oscillator internal*, PB7 dan PB6 dapat digunakan sebagai I/O atau jika menggunakan *Asynchronous Timer/Counter2* maka PB6 dan PB7 (TOSC2 dan TOSC1) digunakan untuk saluran *input timer*.

d. Port C (PC5 - PC0)

Port C merupakan sebuah *7-bit bi-directional I/O* port yang di dalam masing-masing pin terdapat *pull-up* resistor. Jumlah pin nya hanya 7 buah mulai dari *pin C.0* sampai dengan *pin C.6*. Sebagai keluaran/*output port* C memiliki karakteristik yang sama dalam hal menyerap arus (*sink*) ataupun mengeluarkan arus (*source*).

e. RESET / PC6

Jika RSTDISBL *Fuse* diprogram, maka PC6 akan berfungsi sebagai *pin I/O*. *Pin* ini memiliki karakteristik yang berbeda dengan *pin-pin* yang terdapat pada *port C* lainnya. Namun jika RSTDISBL *Fuse* tidak diprogram, maka *pin* ini akan berfungsi sebagai *input reset*. Dan jika *level* tegangan yang masuk ke *pin* ini rendah dan pulsa yang ada lebih pendek dari pulsa minimum, maka akan menghasilkan suatu kondisi reset meskipun *clock*-nya tidak bekerja.

f. Port D (PD7 - PD0)

Port D merupakan *8-bit bi-directional I/O* dengan internal *pull-up* resistor. Fungsi dari port ini sama dengan port-port yang lain. Hanya saja pada port ini tidak terdapat kegunaan-kegunaan yang lain. Pada *port* ini hanya berfungsi sebagai masukan dan keluaran saja atau biasa disebut dengan I/O.

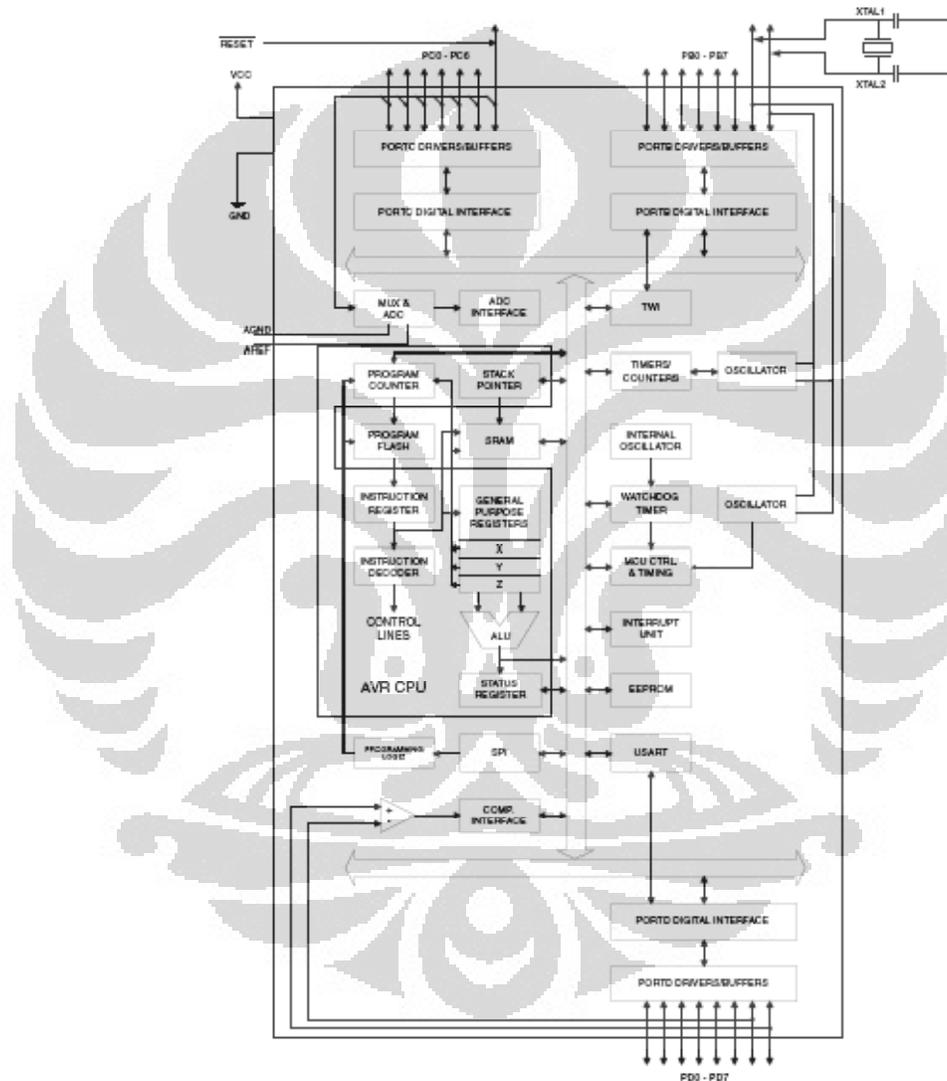
g. AVcc

*Pin* ini berfungsi sebagai *supply* tegangan untuk ADC. Untuk *pin* ini harus dihubungkan secara terpisah dengan VCC karena *pin* ini

digunakan untuk analog saja. Bahkan jika ADC pada AVR tidak digunakan tetap saja disarankan untuk menghubungkannya secara terpisah dengan VCC. Jika ADC digunakan, maka AVcc harus dihubungkan ke VCC melalui *low pass filter*.

h. AREF

Merupakan pin referensi jika menggunakan ADC.



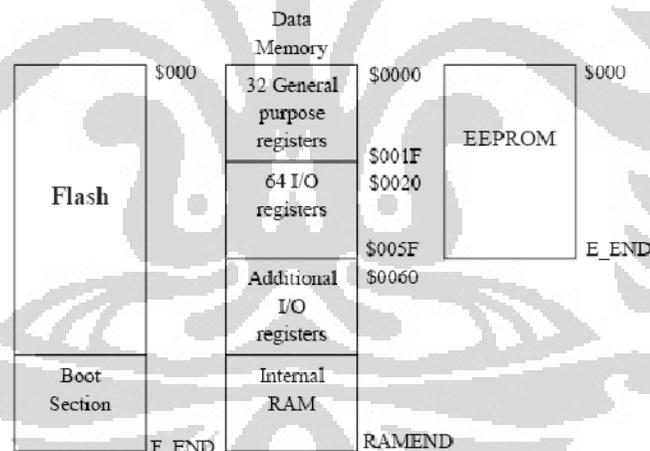
**Gambar 2.7.** Blok Diagram ATMEGA 8

Pada AVR status *register* mengandung beberapa informasi mengenai hasil dari kebanyakan hasil eksekusi instruksi aritmatik. Informasi ini digunakan untuk altering arus program sebagai kegunaan untuk

meningkatkan performa pengoperasian. Register ini di-update setelah operasi *Arithmetic Logic Unit* (ALU) hal tersebut seperti yang tertulis dalam *datasheet* khususnya pada bagian *Instruction Set Reference*. Dalam hal ini untuk beberapa kasus dapat membuang penggunaan kebutuhan instruksi perbandingan yang telah didedikasikan serta dapat menghasilkan peningkatan dalam hal kecepatan dan kode yang lebih sederhana dan singkat. *Register* ini tidak secara otomatis tersimpan ketika memasuki sebuah rutin interupsi dan juga ketika menjalankan sebuah perintah setelah kembali dari interupsi. Namun hal tersebut harus dilakukan melalui *software*. Berikut adalah gambar status *register*.

### 2.2.2 Peta Memori ATMEGA 8

Memori ATMEGA terbagi tiga bagian seperti ditunjukkan pada gambar 2.8. di bawah ini.



**Gambar 2.8.** Peta Memori ATMEGA 8

#### 1. Memori *Flash*

Memori flash adalah memori ROM tempat kode-kode program berada. Kata flash menunjukkan jenis ROM yang dapat ditulis dan dihapus secara elektrik. Memori flash terbagi menjadi dua bagian yaitu bagian aplikasi dan bagian *boot*. Bagian aplikasi adalah bagian kode-kode program aplikasi berada. Bagian *boot* adalah bagian yang digunakan khusus untuk *booting*

awal yang dapat diprogram untuk menulis bagian aplikasi tanpa melalui *programmer / downloader*, misalnya melalui USART.

## 2. Memori Data

Memori data adalah memori RAM yang digunakan untuk keperluan program. Memori data terbagi menjadi empat bagian yaitu : 32 *General Purpose Register* (GPR) adalah register khusus yang bertugas untuk membantu eksekusi program oleh *Arithmatich Logic Unit* (ALU) (*Arithmatich Logic Unit*), dalam instruksi assembler setiap instruksi harus melibatkan GPR. Dalam bahasa C biasanya digunakan untuk variabel global atau nilai balik fungsi dan nilai-nilai yang dapat memperingan kerja ALU. Dalam istilah processor komputer sehari-hari GPR dikenal sebagai "*chace memory*". I/O register dan Additional I/O register adalah *register* yang difungsikan khusus untuk mengendalikan berbagai pheripheral dalam mikrokontroler seperti *pin port, timer/counter, usart* dan lain-lain. Register ini dalam keluarga mikrokontrol MCS51 dikenal sebagai *Special Function Register* (SFR).

## 3. EEPROM

EEPROM adalah memori data yang dapat mengendap ketika chip mati (*off*), digunakan untuk keperluan penyimpanan data yang tahan terhadap gangguan catu daya.

### 2.3.Komunikasi Data Serial

Dalam komunikasi data serial, data dikirim satu bit per satu bit secara berurutan sesuai dengan kanal komunikasi yang ditentukan. Penerima juga akan merima data per bit sesuai dengan kanal komunikasi yang ditentukan. Ada tiga metode yang digunakan dalam komunikasi data serial, yaitu :

1. *Simplex*, merupakan komunikasi satu arah artinya informasi atau data hanya dari pengirim ke penerima.
2. *Half Duplex*, merupakan komunikasi dua arah yang tidak dapat dilakukan dalam waktu yang bersamaan artinya pengirim dan penerima

dapat melakukan komunikasi secara bergantian namun berkesinambungan.

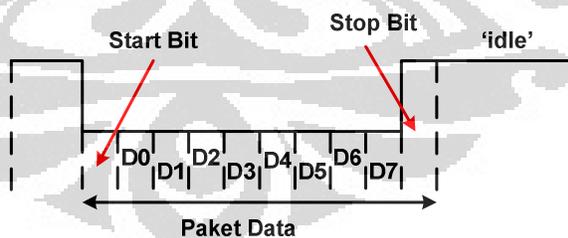
3. *Full Duplex*, merupakan komunikasi dua arah yang dapat dilakukan dalam waktu yang bersamaan artinya pengirim dan penerima dapat melakukan komunikasi dalam waktu yang bersamaan dan berkesinambungan.

Komunikasi Data Serial dibedakan menjadi dua, yaitu :

1. Komunikasi Data Serial Asinkron

Komunikasi serial asinkron merupakan komunikasi serial yang mengirimkan data berupa paket data. Paket data tersebut diawali dengan *start bit* untuk menunjukkan mulainya data, kemudian diikuti data yang akan dikirim berjumlah 8 bit, *parity bit* untuk mendeteksi *error bit* yang dikirim, dan terakhir diakhiri dengan *stop bit* untuk menunjukkan selesainya data.

Dalam teknik komunikasi serial asinkron, saat keadaan '*idle*' maka output logik selalu '1' kecuali saat mengirim data atau menerima data maka data selalu diawali dengan *start bit* yang nilainya selalu logik '0'. Start bit digunakan untuk sinkronisasi antara pengirim dan penerima. Gambar 2.9 menunjukkan bentuk umum format data komunikasi serial asinkron.



**Gambar 2.9.** Paket Data Komunikasi Serial Asinkron

Setelah start bit, selalu diikuti dengan data yang akan dikirim. Data tersebut data yang bernilai kecil (D0) atau *Least Significant Bit (LSB)* sampai dengan data dengan nilai yang paling besar (D7) atau *Most Significant Bit (MSB)*. Data berisi 5, 6, 7, 8 bit tergantung pada sistem

yang akan dibuat. Setelah MSB akan selalu diikuti dengan *parity bit*, *parity bit* dapat digunakan ataupun tidak digunakan tergantung pada sistem yang akan dibuat. Setelah itu paket data akan diakhiri dengan *stop bit* yang selalu bernilai logik '1'. Kecepatan transmisi data tergantung dengan *baudrate* yang digunakan.

## 2. Komunikasi Data Serial Sinkron

Komunikasi serial sinkron, data yang dikirim tidak secara terpisah dengan dibatasi oleh *start bit* dan *stop bit*, melainkan data dikirim dalam bentuk blok data yang dibatasi dengan data sinkronisasi. Jumlah dan karakter sinkronisasi tersebut dapat diprogram.

Dalam komunikasi data serial, pengirim dan penerima harus memenuhi protokol komunikasi, berikut ini protokol komunikasi serial yang harus dipenuhi, yaitu :

### 1. Baudrate

Kecepatan transmisi yang diukur dalam satuan bit. Clock dari pengirim dan penerima harus disinkronisasi dengan baudrate yang sama.

### 2. Start bit

Start bit selalu bernilai logik '0' yang menunjukkan data transmisi dimulai

### 3. Data bit

Lebar data bit berjumlah 5, 6, 7, 8 bit. Bit yang dikirim adalah bit dengan nilai terendah atau LSB.

### 4. Parity bit

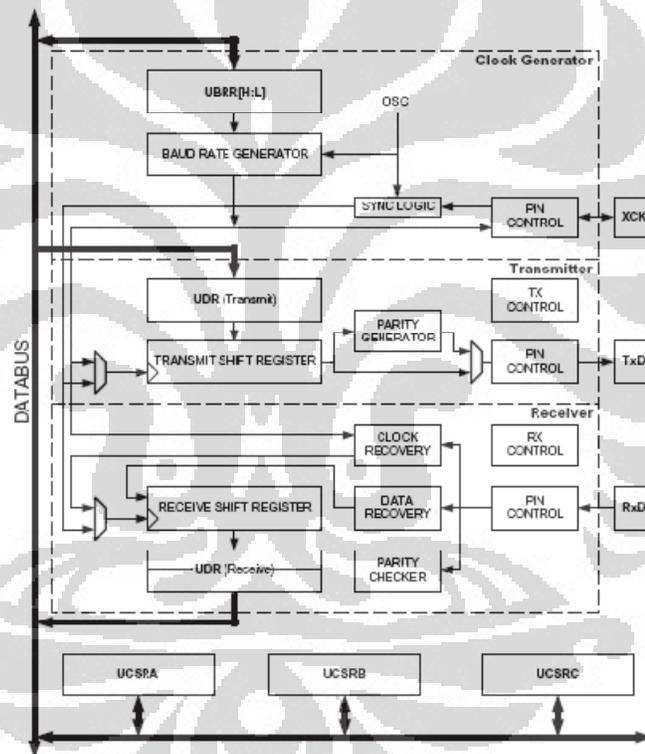
Parity bit digunakan untuk mendeteksi error data transmisi.

### 5. Stop bit

Stop bit selalu bernilai logik '1' yang menunjukkan data transmisi diakhiri.

### 2.3.1. Komunikasi Serial *Universal Synchronous Asynchronous Receiver Transmitter (USART)*

Mikrokontroler AVR Atmega 8 memiliki Port *USART* pada Pin 2 dan Pin 3 untuk melakukan komunikasi data antara mikrokontroler dengan mikrokontroler ataupun mikrokontroler dengan komputer. *USART* dapat difungsikan sebagai transmisi data sinkron, dan asinkron. Data Sinkron berarti *clock* yang digunakan antara *transmitter* dan *receiver* adalah satu sumber *clock*. Sedangkan asinkron berarti *transmitter* dan *receiver* mempunyai sumber *clock* sendiri-sendiri. *USART* terdiri dalam tiga blok yaitu *clock generator*, *transmitter*, dan *receiver*.



**Gambar 2.10.** Blok Diagram Universal Synchronous Asynchronous Receiver Transmitter (USART)

a. *Clock Generator*

*Clock generator* berhubungan dengan kecepatan transfer data (baud rate).

b. *USART Transmitter*

*USART transmitter* berhubungan dengan data pada Pin TX. Perangkat yang sering digunakan seperti register UDR sebagai tempat penampungan data yang akan ditransmisikan. *Flag TXC* sebagai akibat dari data yang ditransmisikan telah sukses (*complete*), dan *flag UDRE* sebagai indikator jika UDR kosong dan siap untuk diisi data yang akan ditransmisikan lagi.

c. *USART Receiver*

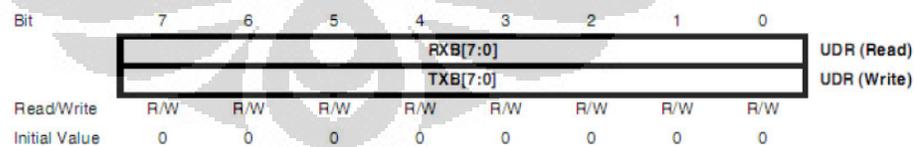
*USART receiver* berhubungan dengan penerimaan data dari Pin RX. Perangkat yang sering digunakan seperti register UDR sebagai tempat penampung data yang telah diterima, dan *flag RXC* sebagai indikator bahwa data telah sukses (*complete*) diterima.

### 2.3.2. Inisialisasi USART

Pada mikrokontroler AVR untuk mengaktifkan dan mengeset komunikasi USART dilakukan dengan cara mengaktifkan register-register yang digunakan untuk komunikasi USART. Register-register yang digunakan untuk komunikasi USART antara lain:

#### 1. *USART I/O Data Register (UDR)*

UDR merupakan register 8 bit yang terdiri dari 2 buah dengan alamat yang sama, yang digunakan sebagai tempat untuk menyimpan data yang akan dikirimkan (TXB) atau tempat data diterima (RXB) sebelum data tersebut dibaca.



**Gambar 2.11.** Register UDR

## 2. USART Control and Status Register A (UCSRA)

Bit	7	6	5	4	3	2	1	0	
	RXC	TXC	UDRE	FE	DOR	PE	U2X	MPCM	UCSRA
Read/Write	R	R/W	R	R	R	R	R/W	R/W	
Initial Value	0	0	1	0	0	0	0	0	

**Gambar 2.12.** Register UCSRA

Penjelasan bit penyusun UCSRA:

a. RXC (*USART Receive Complete*)

Bit ini akan set ketika data yang masuk ke dalam UDR belum dibaca dan akan berlogika nol ketika sudah dibaca. Flag ini dapat digunakan untuk membangkitkan interupsi RX jika diaktifkan dan akan berlogika nol secara otomatis bersamaan dengan eksekusi vektor interupsi yang bersangkutan.

b. TXC (*USART Transmit Complete*)

Bit ini akan set ketika data yang dikirim telah keluar. Flag ini akan membangkitkan interupsi TX jika diaktifkan dan akan clear secara otomatis bersamaan dengan eksekusi vektor interupsi yang bersangkutan.

c. UDRE (*USART Data Register Empty*)

Flag ini sebagai indikator isi UDR. Jika bernilai satu maka UDR dalam keadaan kosong dan siap menerima data berikutnya, jika flag bernilai nol berarti sebaliknya.

d. FE (*Frame Error*)

Bit ini sebagai indikator ketika data yang diterima error, misalnya ketika *stop* bit pertama data dibaca berlogika nol maka bit FE bernilai satu. Bit akan bernilai 0 ketika *stop* bit data yang diterima berlogika nol.

e. DOR (*Data OverRun*)

Bit ini berfungsi untuk mendeteksi jika ada data yang tumpang tindih. Flag akan bernilai satu ketika terjadi tumpang tindih data.

f. PE (*Parity Error*) : bit yang menentukan apakah terjadi kesalahan paritas.

Bit ini berfungsi jika ada kesalahan paritas. Bit akan berlogika satu ketika terjadi bit *parity error* apabila bit paritas digunakan.

g. U2X (*Double the USART Transmission Speed*) :

Bit yang berfungsi untuk menggandakan laju data menjadi dua kalinya. Hanya berlaku untuk modus asinkron, untuk mode sinkron bit ini diset nol.

h. MPCM (*Multi Processor Communication Mode*) :

Bit untuk mengaktifkan modus multi prosesor, dimana ketika data yang diterima oleh USART tidak mengandung informasi alamat akan diabaikan.

### 3. USART CONTROL AND STATUS REGISTER B (UCSRB)

Bit	7	6	5	4	3	2	1	0	
	RXCIE	TXCIE	UDRIE	RXEN	TXEN	UCS22	RXB8	TXB8	UCSRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R	R/W	
Initial Value	0	0	0	0	0	0	0	0	

**Gambar 2.13.** Register UCSRB

Penjelasan bit penyusun UCSRB:

a. RXCIE (*RX Complete Interrupt Enable*)

Bit pengatur aktivasi interupsi penerimaan data serial, akan berlogika satu jika diaktifkan dan berlogika nol jika dinonaktifkan.

b. TXCIE (*TX Complete Interrupt Enable*)

Bit pengatur aktivasi pengiriman data serial, akan berlogika satu jika diaktifkan dan berlogika nol jika dinonaktifkan.

c. UDRIE (*USART Data Register Empty Interrupt Enable*) :

Bit ini berfungsi untuk mengaktifkan interupsi data register kosong, berlogika satu jika diaktifkan dan sebaliknya.

d. RXEN (*Receiver Enable*)

Bit ini berfungsi untuk mengaktifkan pin RX saluran USART. Ketika pin diaktifkan maka pin tersebut tidak dapat digunakan untuk fungsi pin I/O karena sudah digunakan sebagai saluran penerima USART.

e. TXEN (*Transmitter Enable*)

Bit ini berfungsi untuk mengaktifkan pin TX saluran USART. Ketika pin diaktifkan maka pin tersebut tidak dapat digunakan untuk fungsi pin I/O karena sudah digunakan sebagai saluran pengirim USART.

f. UCSZ2 (*Character Size*)

Bit ini bersama dengan UCSZ1 dan UCSZ0 dalam register UCSRC digunakan untuk memilih tipe lebar data bit yang digunakan.

**Tabel 2.1.** Penentuan Ukuran Karakter

UCSZ[2..0]	Ukuran Karakter dalam bit
000	5
001	6
010	7
011	8
100-110	Tidak dipergunakan
111	9

g. RXB8 (*Receive Data Bit 8*)

Bit ini digunakan sebagai bit ke-8 ketika menggunakan format data 9-10 bit, dan bit ini harus dibaca dahulu sebelum membaca UDR

h. TXB8 (*Transmit Data Bit 8*)

Bit ini digunakan sebagai bit ke-8 ketika menggunakan format data 9-10 bit, dan bit ini harus ditulis dahulu sebelum membaca UDR

#### 4. USART CONTROL AND STATUS REGISTER C (UCSRC)

Bit	7	6	5	4	3	2	1	0	
	URSEL	UMSEL	UPM1	UPM0	USBS	UCSZ1	UCSZ0	UCPOL	UCSRC
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	1	0	0	0	0	1	1	0	

**Gambar 2.14.** Register UCSRC

Penjelasan bit penyusun UCSRC:

a. URSEL (*Register Select*):

Bit ini berfungsi untuk memilih register UCSRC dengan UBBRH, dimana untuk menulis atau membaca register UCSRC maka bit harus berlogika satu.

b. *UMSEL (USART Mode Select)*

Bit pemilih mode komunikasi serial antara sinkron dan asinkron.

c. *UPM[1...0] (Parity Mode)*

Bit ini berfungsi untuk memilih mode paritas bit yang akan digunakan. Transmitter USART akan membuat paritas yang akan digunakan secara otomatis.

d. *USBS (Stop Bit Select)*

Bit yang berfungsi untuk memilih jumlah *stop* bit yang akan digunakan.

e. *UCSZ1 dan UCSZ0* : merupakan bit pengatur jumlah karakter serial

Bit yang berfungsi untuk memilih lebar data yang digunakan dikombinasikan dengan bit *UCSZ2* dalam register *UCSRB*

f. *UCPOL (Clock Parity)* :

Bit yang berguna hanya untuk modus sinkron. Bit ini berhubungan dengan perubahan data keluaran dan sampel masukan, dan *clock* sinkron (*XCK*)

#### 2.4. *Transmitter TLP-433*

*Transmitter* adalah serangkaian elektronika yang mengubah energi listrik dari sebuah baterai ke dalam bentuk gelombang radio. Energi tersebut merupakan arus yang bergerak memutar dengan sangat cepat sehingga dapat memancar pada sebuah konduktor (antenna) sebagai gelombang elektromagnet.

*Transmitter* yang digunakan adalah TLP-434. *Transmitter* ini digunakan untuk pengiriman data yang beroperasi pada frekuensi 433,92 MHz. Struktur dan konfigurasi pin dapat dilihat pada gambar 2.15.



**Gambar 2.15.** *Transmitter TLP-434*

#### 2.4.1. Konfigurasi Pin *Transmitter*

*Transmitter TLP-433* memiliki 4 Pin, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki transmitter TLP-433.

a. GND

Pin ini berfungsi sebagai ground dari sebuah komponen. Pin GND terletak pada kaki pertama komponen *transmitter TLP-433*.

b. Data Input

Pin ini berfungsi sebagai sebuah masukan data akan dikirim oleh transmitter. Data tersebut berupa data serial yang dikirim oleh mikrokontroler ke transmitter melalui pin TXD atmega 8. pin Data Input terletak pada kaki kedua *transmitter TLP-433*.

c. VCC

Pin berfungsi sebagai *supply* tegangan digital. Pin VCC terletak pada kaki ketiga *transmitter TLP-433*.

d. ANT

Merupakan sebuah pin untuk antena / pemancar. Pin ANT terletak pada kaki keempat *transmitter TLP-433*.

#### 2.4.2. Karakteristik *Transmitter*

a. Frekuensi Kerja

*Transmitter TLP-433* ini bekerja pada frekuensi 433,92 MHz.

b. Teknik Modulasi

*Transmitter* TLP-433 ini bekerja menggunakan modulasi digital *Amplitude Shift Keying* (ASK). Modulasi ASK merupakan bentuk modulasi yang mewakili data digital sebagai variasi dalam amplitude gelombang karier. Amplitudo dari sebuah sinyal karier analog mengubah dengan aliran bit, menjaga frekuensi dan fase konstan. *Level* amplitudo dapat digunakan untuk mewakili logika binary 0 dan 1.

c. Tegangan Kerja

*Transmitter* TLP-433 ini dapat bekerja level tegangan 1,5 – 12 Vdc.

d. Jangkauan

*Transmitter* TLP-433 ini memiliki radius kerja yang cukup jauh, yaitu : untuk keadaan diluar atau area terbuka, transmitter ini dapat memancarkan hingga jarak 400 kaki (121,92m). sedangkan untuk keadaan didalam dapat memancarkan hingga jarak 200 kaki (60,92m).

## 2.5. Receiver RLP-433

*Receiver* adalah serangkaian elektronika yang menerima input dari sebuah antenna, menggunakan filter elektronik dari sejumlah sinyal radio yang diharapkan dari semua sinyal radio yang diterima oleh antenna. Sinyal yang telah diterima kemudian dikuatkan oleh *amplifier* dan kemudian masuk kedalam *demodulation* dan *decoder* sehingga menghasilkan sinyal output yang berguna, seperti suara, gambar, data digital, hasil pengukuran.

Untuk menerima data yang telah dikirim *transmitter* diperlukan receiver dengan frekuensi yang sama, maka digunakan *receiver* RLP-434 yang juga beroperasi pada frekuensi 433,92MHz. Bentuk fisik dari RLP-434 ditunjukkan pada gambar 2.16. dibawah ini.



**Gambar 2.16.** Receiver RLP-434

### 2.5.1. Konfigurasi Pin Receiver RLP - 434

*Receiver* RLP-434 memiliki 8 Pin, yang masing-masing pin nya memiliki fungsi yang berbeda-beda baik sebagai port maupun fungsi yang lainnya. Berikut akan dijelaskan fungsi dari masing-masing kaki *receiver* TWS-DS.

a. DGND

Pin ini berfungsi sebagai ground digital. Pin GND terletak pada kaki kedelapan komponen *receiver* RLP-434.

b. GND

Pin ini berfungsi sebagai ground dari sebuah antenna. Pin ini terletak pada kaki kedua dan ketiga komponen *receiver* RLP-434.

c. VCC

Pin berfungsi sebagai *supply* tegangan digital. Pin VCC terletak pada kaki ketiga *receiver* RLP-434.

d. DATA

Pin ini berfungsi sebagai sebuah data keluaran akan dikirim oleh transmitter. Data tersebut berupa data serial yang akan dibaca oleh mikrokontroler melalui pin RXD atmega 8. pin DATA terletak pada kaki ketujuh *receiver* RLP-434.

e. ANT

Merupakan sebuah pin untuk antena / pemancar. Pin ANT terletak pada kaki pertama *receiver* RLP-434.

f. *Not Connected* (NC)

Merupakan sebuah pin yang tidak digunakan karena tidak memiliki fungsi. Pin NC terletak pada kaki keenam *receiver* RLP-434.

### 2.5.2. Karakteristik Receiver RLP - 434

a. Frekuensi Kerja

*Receiver* RLP-434 ini bekerja pada frekuensi 433,92 MHz.

b. Tegangan Kerja

*Receiver* RLP-434 ini dapat bekerja level tegangan 3,5 – 5,5 Vdc.

c. Temperatur Kerja

*Receiver* RLP-434 bekerja baik pada suhu  $-20 - 70$  °C.

## 2.6. Aktuator

Aktuator merupakan penggerak dalam sebuah system dan merupakan komponen yang penting dalam sebuah system yang membutuhkan gerakan atau perpindahan.

### 2.6.1. Motor DC

Motor DC adalah alat yang mengubah energi listrik DC menjadi energi mekanik putaran. Sebuah motor DC sederhana dibangun dengan menempatkan kawat yang dialiri arus di dalam medan magnet. Kawat yang membentuk loop ditempatkan sedemikian rupa diantara dua buah magnet permanen. Bila arus mengalir pada kawat, arus akan menghasilkan medan magnet sendiri yang arahnya berubah-ubah terhadap arah medan magnet permanen sehingga menimbulkan putaran. Besarnya nilai arus yang mengalir pada lilitan kawat akan mempengaruhi besarnya nilai torsi dan kecepatan motor. Bentuk fisik motor DC ditunjukkan pada gambar 2.17. di bawah ini.



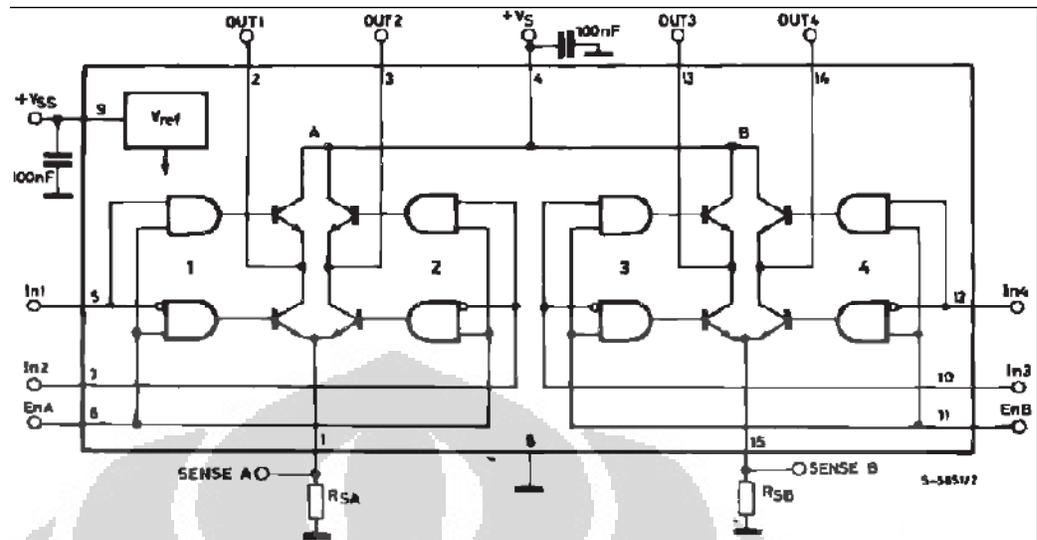
**Gambar 2.17.** Motor DC

Motor DC dibuat dalam tiga tipe dasar. Tiga tipe dasar motor dc itu adalah motor seri, motor paralel, dan motor gabungan. Motor seri banyak

dipakai pada beban awal yang berat dengan momen gaya yang tinggi sehingga putaran motor akan menjadi rendah, pada aplikasinya yaitu pada motor derek atau kerek pengangkat. Motor paralel dibuat untuk pengaplikasian pada pemompaan cairan, dimana kecepatan yang konstan sangat diperlukan.

### **2.6.2. Driver Motor**

Rangkaian pengendali merupakan rangkaian antarmuka yang digunakan untuk mengoperasikan output yang memiliki daya besar dengan input berdaya rendah. Pada sistem katub ini, input dari rangkaian pengendali ini berupa output dari microcontroller yang keduanya memiliki nilai arus dan tegangan yang rendah, sedangkan rangkaian pengendali merupakan rangkaian yang membutuhkan arus dan tegangan yang besar. Untuk mengatasi perbedaan nilai arus dan tegangan pada input dan output digunakan rangkaian pengendali sekaligus untuk melindungi input dari besarnya daya pada output. Dalam sistem katub ini, rangkaian pengendali motor dc 12 V yang digunakan adalah rangkaian IC L 298. L 298 adalah jenis IC driver motor yang dapat mengendalikan arah putaran dan kecepatan motor DC ataupun motor stepper. Mampu mengeluarkan output tegangan untuk motor DC dan motor stepper sebesar 50 volt. IC L298 terdiri dari transistor-transistor logik (TTL) dengan gerbang NAND yang memudahkan dalam menentukan arah putaran suatu motor DC dan motor stepper. Pin enable A dan B untuk mengendalikan jalan atau kecepatan motor, pin input 1 sampai 4 untuk mengendalikan arah putaran motor. Pin enable diberi tegangan 5V untuk kecepatan penuh dan PWM (Pulse Widht Modulation) untuk kecepatan rotasi yang bervariasi tergantung dari level high nya seperti yang terlihat pada gambar 2.18. di bawah ini.



Gambar 2.18. Diagram Blok IC L298

## BAB 3

### PERANCANGAN ALAT

#### 3.1. Deskripsi Sistem

Sistem ini adalah modul beroda dengan kendali *remote* kontrol nirkabel, yang terdiri dari modul *remote* dan modul beroda. Pada *remote* terdiri dari empat buah *push-button* sebagai *input* untuk mengendalikan modul beroda. Pada modul *remote* terdapat system elektronik yang menerapkan DTMF. Hasil pengolahan DTMF menjadi *input* untuk mikrokontroler pada modul *remote*. Kemudian mikrokontroler pada modul *remote* akan mentransmisikan data melalui *transmitter* TLP434 secara *wireless*. Data yang ditransmisikan oleh modul *remote*, akan diterima modul *receiver* RLP434 pada modul beroda. Mikrokontroler pada modul beroda berfungsi mengolah data yang diterima dari *receiver* dan kemudian mengatur gerak actuator, yaitu motor DC.

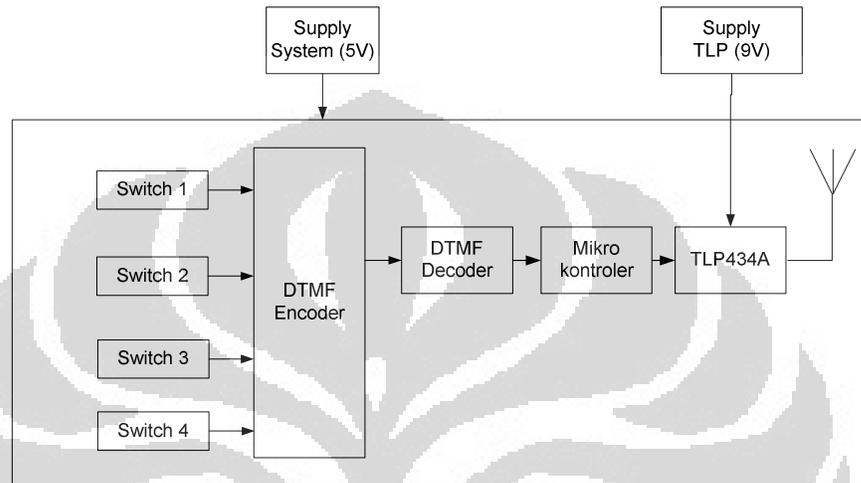
Sistem ini mempunyai kemampuan mengendalikan modul beroda dengan empat keadaan, yaitu modul beroda bergerak maju, modul beroda bergerak mundur, modul beroda bergerak belok kiri, dan modul beroda bergerak belok kanan.

Tujuan sistem :

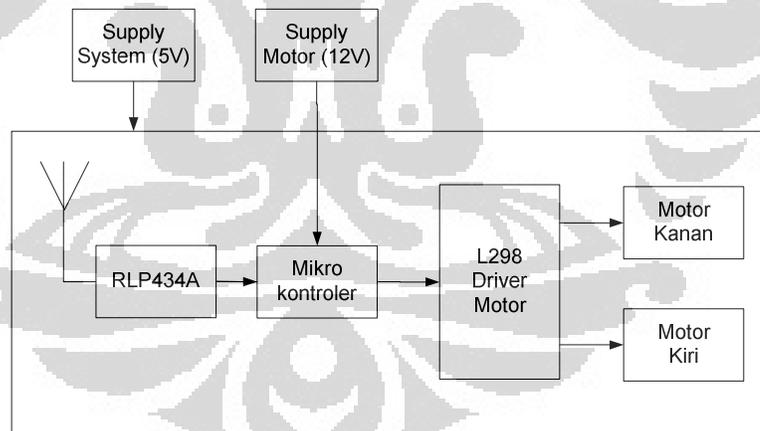
Mampu mengirimkan data dari *transmitter* ke *receiver*, mampu mengaktifkan atau menon-aktifkan bagian yang dikendalikan seperti peralatan elektronika yang ada dirumah.

Diagram Blok Sistem :

Secara umum system dapat dideskripsikan dalam dua buah diagram blok, yaitu diagram blok modul *remote* pada gambar dan diagram blok modul modul beroda pada gambar 3.1 dan gambar 3.2. di bawah ini.



**Gambar 3.1.** Diagram Blok Modul *Remote*



**Gambar 3.2.** Diagram Blok Modul Modul beroda

Berdasarkan gambar di atas, setiap bagian sistem memiliki sumber baterai yang terpisah antara bagian yang satu dengan yang lainnya.

Spesifikasi Sistem :

1. Modul *Remote*

*Input* : modul ini memiliki *input* empat buah *push-button* yang terhubung dengan *decoder* DTMF

*Output* : modul ini memiliki *output* data melalui pin TXD pada mikrokontroler yang ditransmisikan oleh TLP434.

2. Modul Modul beroda

*Input* : modul modul beroda memiliki *input* data RLP434 dari transmisi TLP434.

*Output* : modul modul beroda memiliki *output* berupa data yang dikeluarkan mikrokontroler untuk mengaktifkan motor DC melalui perantara *motor driver*

3. Komunikasi yang terjadi antara *transmitter* TLP434 dengan *receiver* RLP434 adalah komunikasi serial asinkron, dan berlangsung satu arah (*simplex*).

4. Frekuensi yang digunakan *transmitter* TLP434 dengan *receiver* RLP434 adalah 433 MHz.

5. Sumber Tegangan

Modul *remote* memiliki sumber tegangan baterai 9 volt dan modul modul beroda memiliki sumber tegangan baterai 9 volt

6. Pengolah Data

Pengolah data yang digunakan system adalah mikrokontroler ATMega 8

7. Aktuator

Aktuator yang digunakan adalah dua buah motor DC

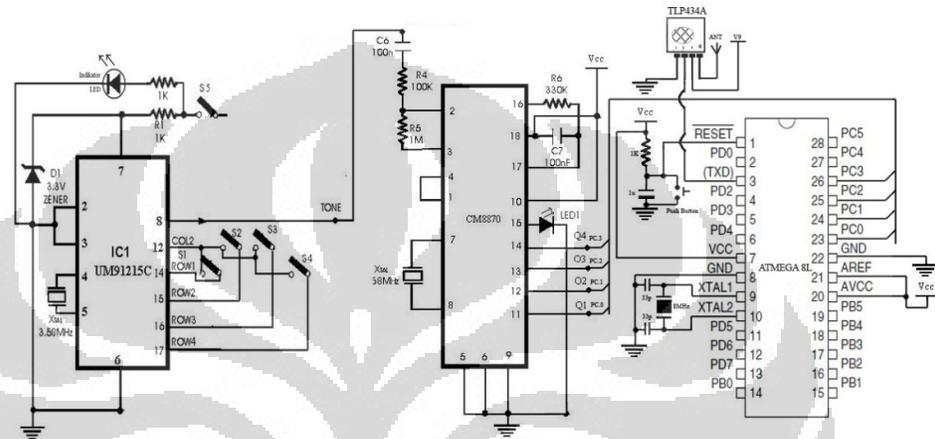
8. Rangkaian *Driver*

Rangkaian *driver* yang digunakan untuk mengendalikan Motor DC yaitu IC L298.

### 3.2. Perancangan Modul *Remote*

Fungsi : Mengirimkan data dari mikrokontroler berdasarkan *input push-button* melalui *transmitter RLP434*, ke *receiver RLP434* pada modul modul beroda.

Rangkaian modul *remote* :



Gambar 3.3. Rangkaian Modul *Remote*

Spesifikasi modul *remote* :

1. Mikrokontroler ATmega 8
2. *Encoder DTMF*, yaitu IC UM91215
3. *Decoder DTMF*, yaitu IC M8870
4. Empat *push-button*
5. LED indikator
6. *Transmitter TLP434*
7. Baterai 9 Volt

Perancangan modul *remote* dimulai dengan menentukan empat *push-button* terhubung dengan *encoder DTMF* (IC UM 91215) kolom 2 dan baris 1,2,3, dan 4 sehingga didapatkan konfigurasi pada table di bawah ini.

Tabel 3.1 Konfigurasi *Push-Button* Modul *Remote*

Kombinasi	Angka ( <i>Keyboard</i> )
Kolom 2 dan Baris 1	2
Kolom 2 dan Baris 2	5
Kolom 2 dan Baris 3	8
Kolom 2 dan Baris 4	0

Berdasarkan tabel di atas maka didapatkan konfigurasi pin yang terhubung antara *push-button* dengan *encoder* DTMF (IC UM 91215) sebagai berikut.

Tabel 3.2 Konfigurasi Pin UM 91215 dengan *Push-Button*

Tombol	Kombinasi	Pin UM 91215
<i>Push-Button</i> 1 (S1)	Kolom 2 dan Baris 1	Pin 12 dan 14
<i>Push-Button</i> 2 (S2)	Kolom 2 dan Baris 2	Pin 12 dan 15
<i>Push-Button</i> 3 (S3)	Kolom 2 dan Baris 3	Pin 12 dan 16
<i>Push-Button</i> 4 (S4)	Kolom 2 dan Baris 4	Pin 12 dan 17

Setelah menentukan konfigurasi pin yang digunakan pada *encoder* DTMF (IC UM 91215) maka dapat diketahui frekuensi yang dihasilkan pada pin *Tone* (8) UM 91215, pada table di bawah ini.

Tabel 3.3. Frekuensi *Output Encoder* UM 91215

Angka ( <i>Keyboard</i> )	Frekuensi Baris/ <i>Low</i> (Hz)	Frekuensi Kolom/ <i>High</i> (Hz)
2	697	1336
5	770	1336
8	852	1336
0	941	1336

Proses selanjutnya merancang *decoder* DTMF untuk tombol-tombol yang digunakan pada *encoder* UM91215, yaitu tombol 2, tombol 5, tombol 8, dan tombol 0.

Tabel 3.4. Hasil Pengkodean *Decoder* M8870

Angka ( <i>Keyboard</i> )	Decode Output ( <i>logic</i> )
2	0 0 1 0
5	0 1 0 1
8	1 0 0 0
0	1 0 1 0

Setelah merancang rangkaian elektronika, merancang PCB, menentukan baterai, proses selanjutnya yaitu membentuk tempat atau kotak untuk modul *remote*, seperti yang ditunjukkan gambar 3.4. di bawah ini.

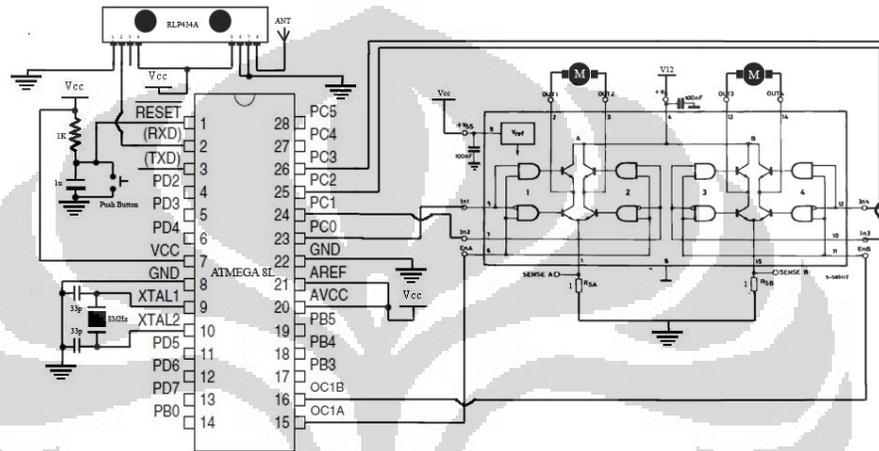


Gambar 3.4. Modul Remote

### 3.3. Perancangan Modul Modul beroda

Fungsi : Menerima data dari *transmitter* RLP434, mengolah data pada mikrokontroler, dan menggerakkan actuator modul beroda.

Rangkaian elektronika modul beroda :



Gambar 3.5. Rangkaian Modul Beroda

Spesifikasi system :

1. Receiver RLP434
2. Mikrokontroler ATmega 8
3. Driver motor L298
4. Dua buah motor DC
5. Baterai 9 V
6. LED indikator

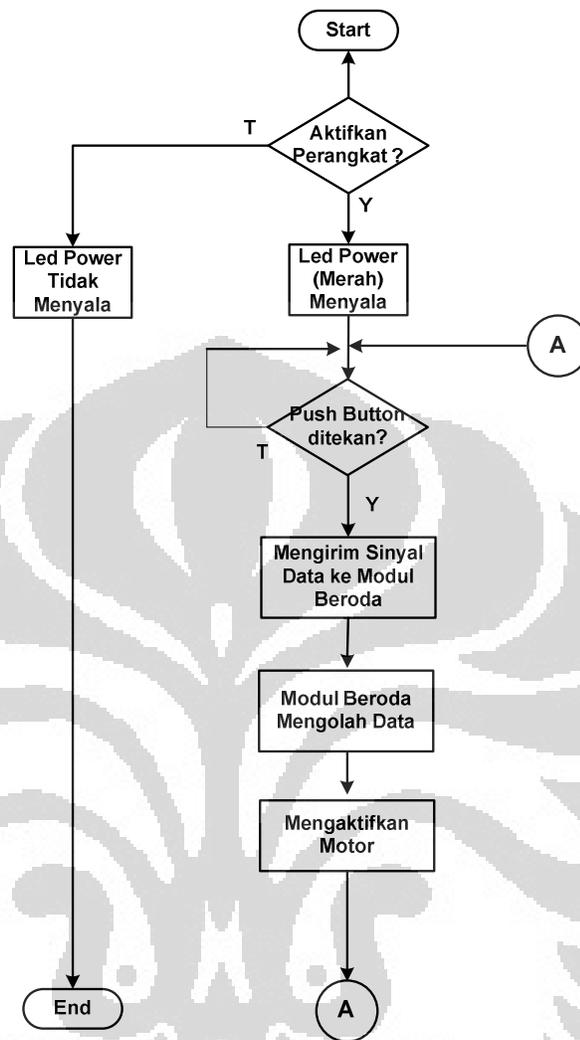
Setelah merancang rangkaian elektronika, menentukan motor DC dan baterai, maka selanjutnya merancang mekanik dari modul beroda, seperti pada gambar di bawah ini.



**Gambar 3.6.** Modul Beroda

### **3.4. Perancangan Algoritma Sistem**

Untuk menciptakan system yang efektif dan efisien maka perlu perancangan algoritma yang mudah dan tidak berbelit. System ini menggunakan komunikasi satu arah antara *transmitter* dan *receiver*-nya, sehingga pembuatan algoritma tidak terlalu rumit, seperti pada gambar di bawah ini.



**Gambar 3.7.** Diagram Alir Algoritma Sistem

Pada modul modul beroda, sistem aktif jika tombol atau *switch* berada pada kondisi ON. Selanjutnya modul modul beroda menunggu sampai ada proses pengiriman data dari *receiver* melalui penekanan *push-button* di modul *remote*. Jika modul modul beroda sudah menerima data dari *receiver*, maka mikrokontroler melakukan proses pengolahan data. Selanjutnya mikrokontroler akan mengaktifkan modul beroda sesuai dengan instruksi yang diterima dari *receiver*, yaitu dengan menggerakkan motor DC. Mikrokontroler pada modul modul beroda, menunggu sampai ada pengiriman data selanjutnya namun tetap mengaktifkan instruksi pada kondisi terakhir atau sebelumnya.

### 3.5. Perancangan *Driver Motor*

Sistem ini menggunakan *driver* motor L298 untuk *interface* antara mikrokontroler dengan motor DC. Berikut perancangan konfigurasi dari *driver* motor.

a. Sumber tegangan motor DC 12 Volt

b. *Input* L298

*Input* 1 terhubung dengan mikrokontroler *port* PC.0

*Input* 2 terhubung dengan mikrokontroler *port* PC.1

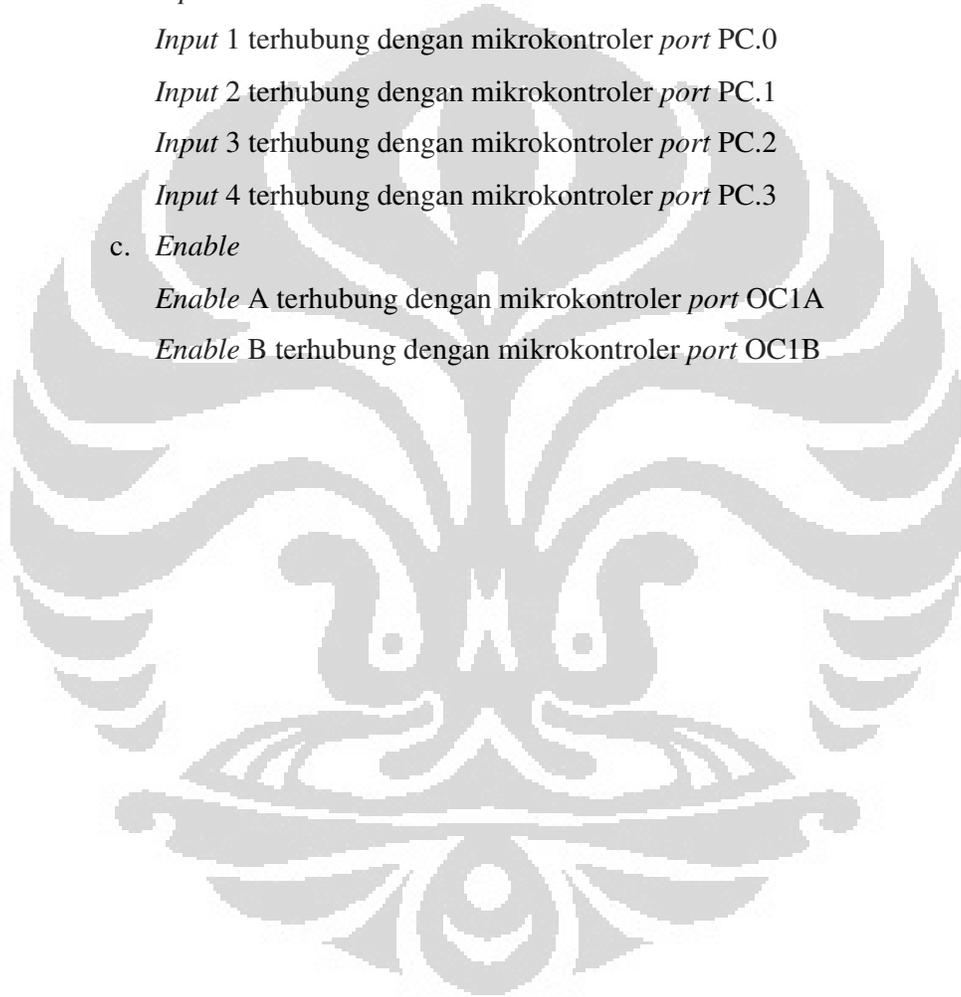
*Input* 3 terhubung dengan mikrokontroler *port* PC.2

*Input* 4 terhubung dengan mikrokontroler *port* PC.3

c. *Enable*

*Enable* A terhubung dengan mikrokontroler *port* OC1A

*Enable* B terhubung dengan mikrokontroler *port* OC1B



## BAB 4

### PENGUJIAN DAN ANALISA DATA

#### 4.1. Deskripsi Pengujian

Nama Pengujian : Pengujian rancang bangun kendali modul beroda nirkabel dengan teknik DTMF.

Tujuan Pengujian: Mengetahui apakah *transmitter*, *receiver*, dan rangkaian *driver*, pada sistem telah bekerja sesuai dengan yang telah direncanakan.

Target Pengujian :

- a. Menguji apakah fungsi *transmitter*, *receiver*, dan rangkain *driver*, dan sesuai dengan perencanaan.
- b. Mengetahui apakah protokol komunikasi yang digunakan telah sesuai dengan perencanaan.
- c. Membaca data serial yang dikirimkan oleh *transmitter* ke *receiver*.

Data Lingkungan Pengujian :

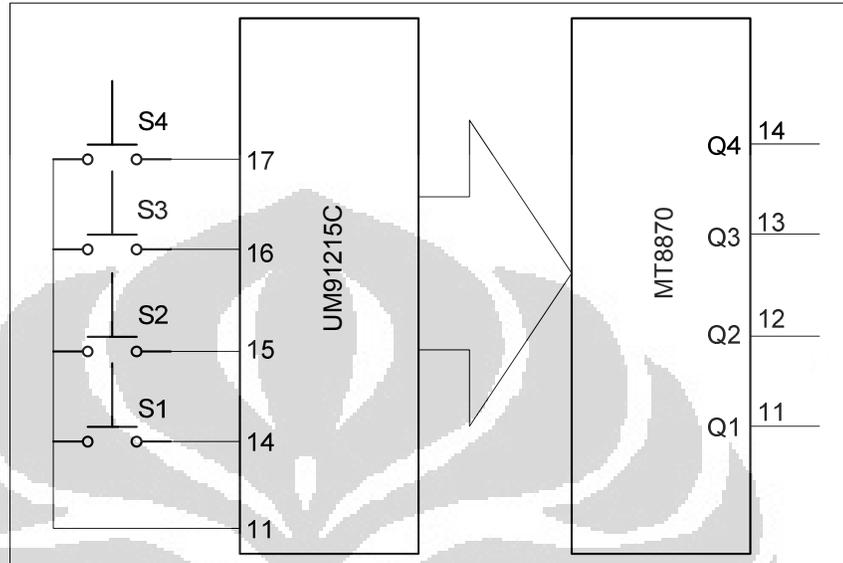
- a. Lokasi Pengujian : Laboratorium Kendali Teknik Elektro UI
- b. Tanggal Pelaksanaan : 12 Juni 2012
- c. Pelaksana : Muhammad Fachri

#### 4.2. Pengujian Penekanan Tombol Modul *Remote* Bersamaan

- a. Prosedur Pengujian *Remote*
  1. Mengaktifkan modul *remote* dan modul modul beroda
  2. Menekan dua buah tombol secara bersamaan
  3. Mengamati pergerakan pada modul modul beroda

b. Konfigurasi pengujian

Pengujian dilakukan dengan mengambil bagian DTMF saja pada modul *remote*.



**Gambar 4.1.** Konfigurasi Pengujian Penekan Tombol

c. Hasil Pengujian

Tabel 4.1 Hasil Pengujian Penekanan Tombol

Penekanan Tombol	Output MT8870			
	Q4	Q3	Q2	Q1
S1 dan S2	0	0	1	0
S2 dan S3	0	1	0	1
S3 dan S4	1	0	0	0
S4 dan S1	1	0	1	0

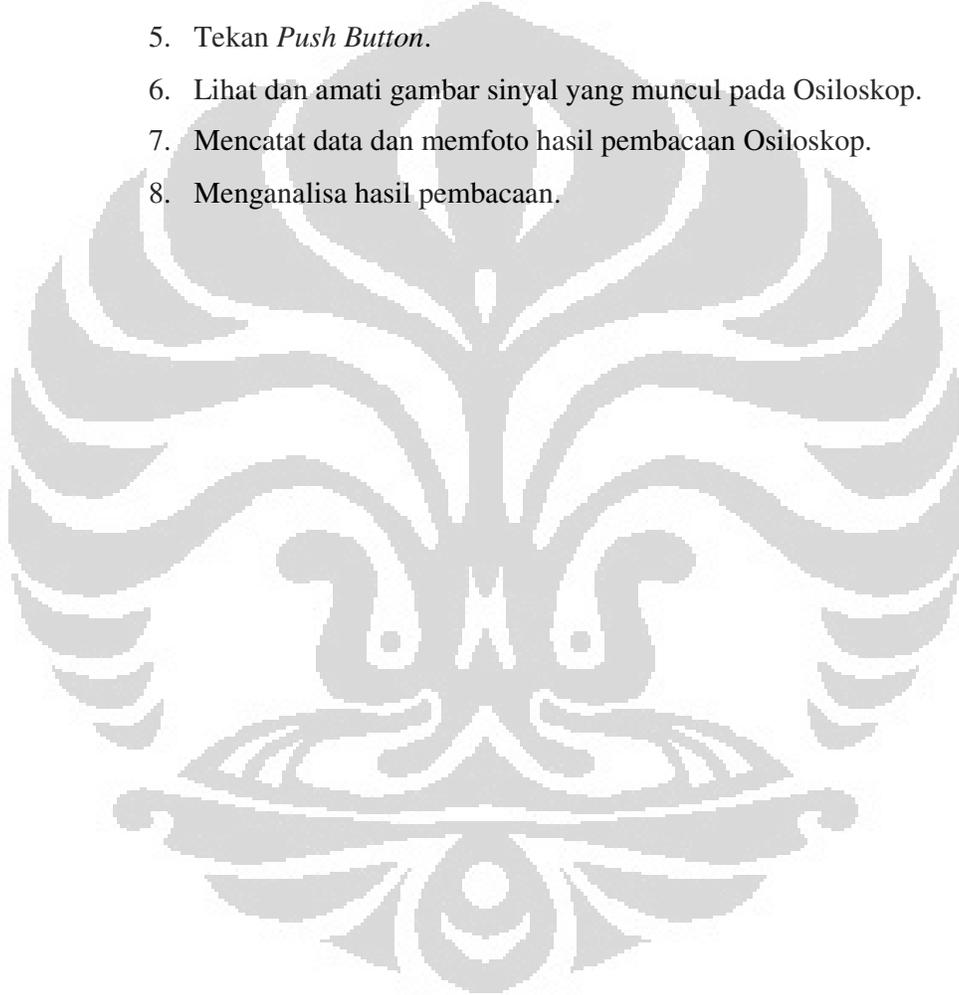
d. Analisa Data Pengujian

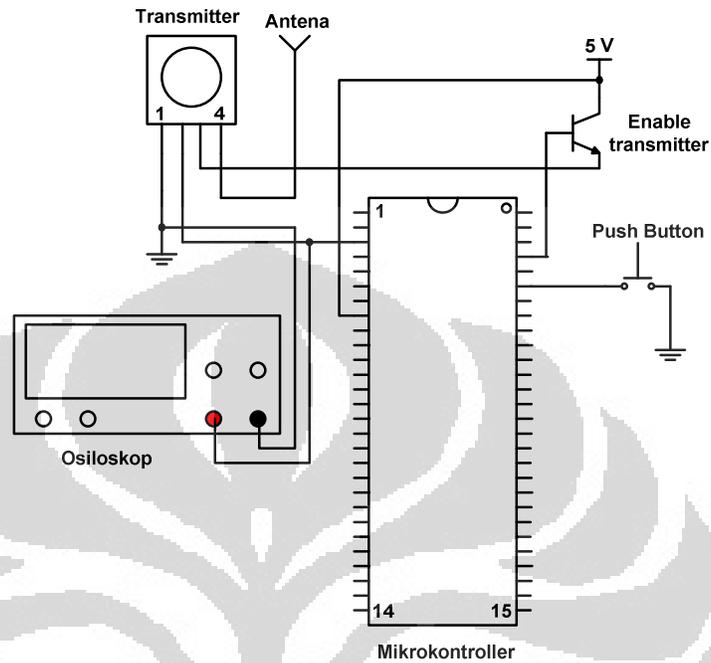
Berdasarkan hasil pengujian yang ditampilkan pada tabel di atas, maka dapat disimpulkan bahwa jika terjadi penekanan dua tombol secara bersamaan maka nilai yang terbaca pada *encoder* adalah tombol yang terlebih dahulu tertekan. Sehingga hasil keluaran dari *decoder* mengikuti tombol yang paling awal tertekan pada *encoder*.

### 4.3. Pengukuran Data yang Dikirim *Transmitter*

#### a. Prosedur Pengujian *Transmitter*

1. Membuat konfigurasi pengujian sesuai dengan gambar 4.2.
2. Hidupkan dan mengatur Osiloskop pada skala ukur yang tepat.
3. Hubungkan kabel BNC to Probe ke Osiloskop.
4. Aktifkan *switch* modul remote ke mode ON.
5. Tekan *Push Button*.
6. Lihat dan amati gambar sinyal yang muncul pada Osiloskop.
7. Mencatat data dan memfoto hasil pembacaan Osiloskop.
8. Menganalisa hasil pembacaan.



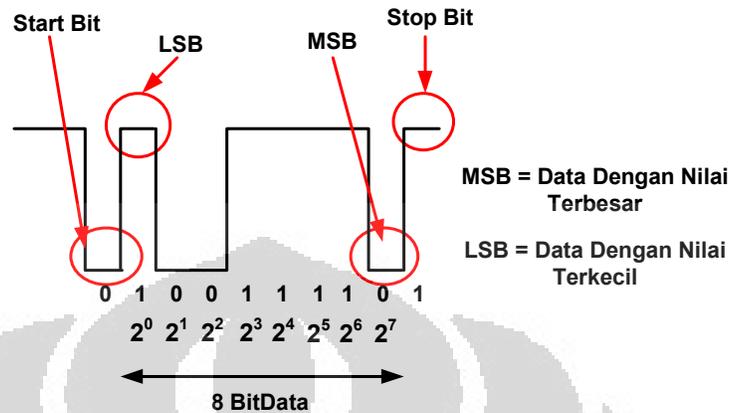
b. Konfigurasi Pengujian *Transmitter*

**Gambar 4.2.** Konfigurasi Pengujian *Transmitter*

c. Peralatan Pendukung Pengujian *Transmitter*

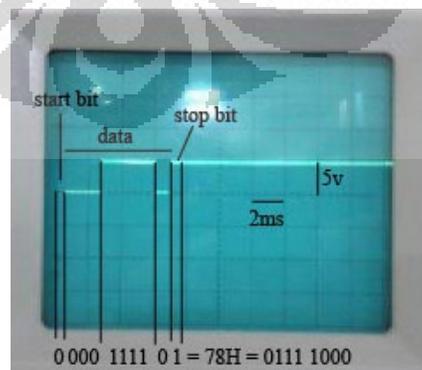
**Tabel 4.2.** Peralatan Pendukung Pengujian *Transmitter*

Komponen	Merk	Tipe	Simbol	Fungsi
Osiloskop Digital	Tektronix	TDS 3052B	-	Menampilkan sinyal
Osiloskop Analaog	-	-	-	Menampilkan sinyal
Modul <i>remote</i>	-	-	-	Mengirim data

d. Data dan Analisis Hasil Pengujian *Transmitter*

Gambar 4.3. Paket Data Serial

Data serial asinkron dikirim dengan paket data berupa bit start, 8 bit data, parity, stop bit. Paket data ini selalu diawali dengan *start bit* berjumlah satu bit yang nilainya selalu logik '0' kemudian diikuti dengan data yang bernilai terendah (D0) atau LSB hingga data yang bernilai terbesar (D7) atau MSB. Setelah MSB diikuti dengan *parity bit*, karena pada sistem ini tidak digunakan parity bit maka setelah MSB diikuti dengan *stop bit* berjumlah satu bit yang nilainya selalu logic '1'. Dalam keadaan *idle* atau keadaan dimana transmitter maupun receiver tidak sedang mengirim atau menerima data, keadaan tersebut selalu berlogik '1'.



Gambar 4.4. Data Serial pada Modul Remote

Pada gambar 4.4, data serial yang dikirim modul *remote* 1 berupa karakter 'x' yang merupakan kode ASCII dari 78H atau 0111 1000B. data dikirim berupa paket data, yaitu 1 start bit, 8 bit data, dan 1 stop bit dengan *baudrate* 1200bps. logik '1' bernilai 5V dan logik '0' bernilai 0V sesuai dengan skala volt/div yang terbaca pada osiloskop. Osiloskop diatur dengan skala 5 volt/div dan 2 ms/div sehingga data tersebut dapat diamati dan dianalisa.

Dalam pembacaan data serial pada osiloskop perlu dilihat juga *baudrate* yang digunakan oleh mikrkontroller dengan pengaturan time/div pada osiloskop. Agar data serial dapat dibaca, maka harus dihitung lebar data setiap 1 bit.

$$\text{Baudrate} = \frac{1200}{1s} = \frac{1200}{1000ms} = 1,2\text{bit} / ms$$

$$1 \text{ bit} = \frac{1\text{bit}}{1,2\text{bit} / ms} = 0,833ms$$

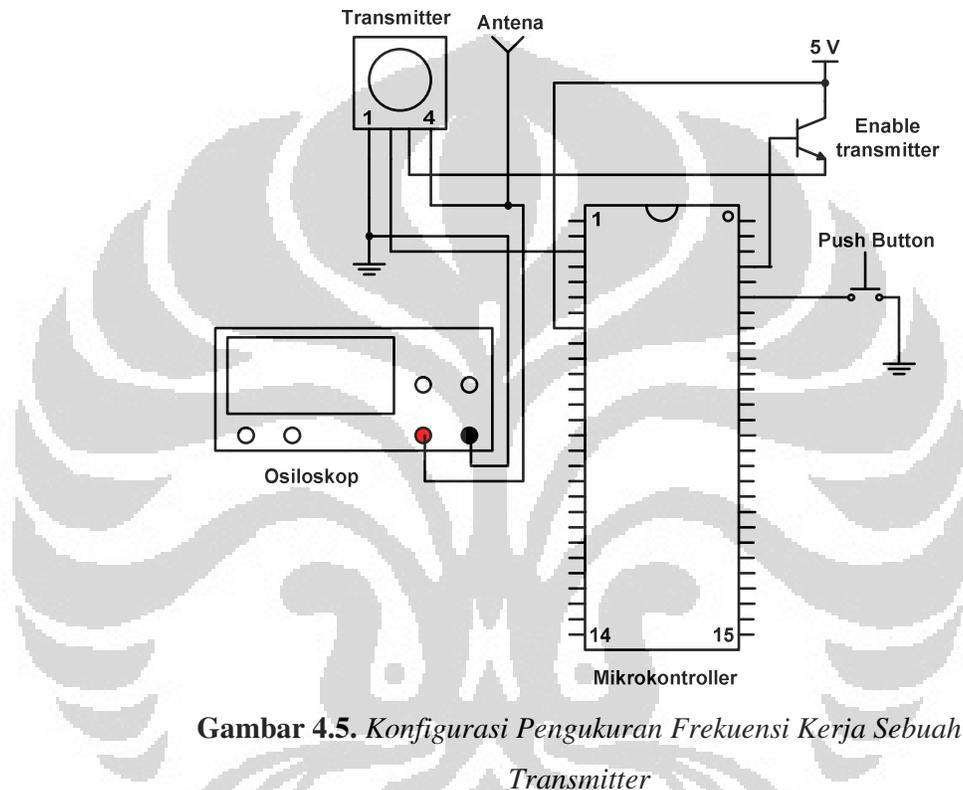
Dari hasil perhitungan, didapat bahwa lebar data untuk 1 bit adalah 0,83ms. Kemudian menghitung lebar data 1 bit dengan time/div sebesar 2ms pada osiloskop. Setiap divisi atau kotak besar terdapat 5 skala atau kotak kecil maka artinya setiap kotak kecil besarnya adalah 0,4 ms. Berdasarkan *baudrate* yang digunakan, lebar data 1 bit sebesar 0,833ms sehingga membutuhkan dua kotak kecil atau 0,8ms untuk setiap 1 bit. Terlihat pada gambar 4.4, bahwa lebar data untuk 1 bit adalah 0,8ms sehingga data yang dikirim telah sesuai dengan baudratanya.

#### 4.4. Pengukuran Frekuensi Kerja Sebuah *Transmitter*

- a. Prosedur Pengukuran Frekuensi Kerja Sebuah *Transmitter*
  1. Membuat konfigurasi pengujian sesuai dengan gambar 4.6.
  2. Hidupkan dan Mengatur Osiloskop pada skala ukur yang tepat.
  3. Hubungkan kabel BNC to Probe ke Osiloskop.
  4. Aktifkan swtich modul *remote* ke mode ON.
  5. Tekan Push Button pada modul *remote*.

6. Lihat dan Amati gambar sinyal yang muncul pada Osiloskop.
7. Mencatat dan Memfoto hasil pembacaan Osiloskop.
8. Menganalisa hasil pembacaan.

b. Konfigurasi Pengukuran Frekuensi Kerja Sebuah *Transmitter*



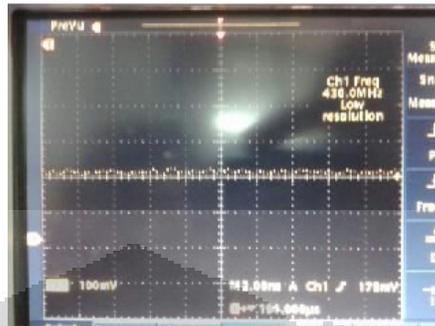
**Gambar 4.5.** Konfigurasi Pengukuran Frekuensi Kerja Sebuah *Transmitter*

c. Peralatan Pendukung Pengukuran Frekuensi Kerja Sebuah *Transmitter*

**Tabel 4.3.** Peralatan Pengukuran Frekuensi Kerja Sebuah *Transmitter*

Komponen	Merk	Tipe	Simbol	Fungsi
Osiloskop Digital	Tektronix	3052B	-	Menampilkan sinyal
Osiloskop Analag	-	-	-	Menampilkan sinyal
Modul <i>remote</i>	-	-	-	Mengirim data

- d. Data Hasil Pengujian pengukuran frekuensi kerja sebuah transmitter



**Gambar 4.6.** Data Hasil Frekuensi Kerja Transmitter pada Osiloskop

- e. Analisis Data Hasil Pengukuran Frekuensi Kerja Transmitter

Besarnya frekuensi pada *transmitter* data diukur pada sinyal keluaran dari antenna terhadap *ground* pada rangkaian. Pada pengukuran didapatkan bahwa frekuensi dari *transmitter* mempunyai nilai 430 MHz seperti pada gambar 4.6. nilai tersebut sedikit memiliki perbedaan dengan nilai perancangan, karena besarnya frekuensi modulasi mendekati frekuensi *carrier*, dimana frekuensi *carrier* yang digunakan adalah 433,92 Mhz. Hal tersebut terjadi karena adanya rugi-rugi daya pada tiap-tiap komponen pendukung dalam modul *remote*. Walaupun terdapat sedikit perbedaan pada frekuensi yang dirancang dengan frekuensi yang pada hasil pengukuran, namun sudah cukup untuk kebutuhan transmisi frekuensi *transmitter* tersebut.

Selain itu alasan digunakannya transmitter dengan frekuensi 433,92 MHz untuk perangkat keras pada skripsi ini karena komponen tersebut tidak sulit untuk ditemukan dan harganya yang terjangkau.

#### 4.5. Pengukuran Jangkauan Maksimum Sebuah *Transmitter*

- a. Prosedur pengukuran jangkauan maksimum sebuah transmitter
1. Aktifkan swtich modul *remote* ke mode ON.
  2. Aktifkan swtich modul *plant* ke mode ON.
  3. Tekan Push Button pada modul *remote*.
  4. Lakukan pengukuran dengan melangkah.

5. Kalibrasi langkah tersebut dengan alat ukur meter.
6. Catat hasil pengukuran ke tabel.

b. Data hasil Pengukuran

**Tabel 4.4.** *Pengukuran Transmisi Modul Remote dengan Modul Beroda*

Jarak	Pengujian 1	Pengujian 2
1 meter	Baik	Baik
5 meter	Baik	Baik
10 meter	Baik	Baik
15 meter	Baik	Baik
18 meter	Baik	Baik
19 meter	Baik	Tidak Baik
20 meter	Tidak Baik	Tidak Baik
25 meter	Tidak Baik	Tidak Baik
30 meter	Tidak Baik	Tidak Baik

Dengan kondisi baterai modul beroda 11.6 Volt dan kondisi baterai modul *remote* 9 Volt

c. Analisis data hasil pengukuran

Berdasarkan data hasil pengukuran pada tabel 4.4 diketahui bahwa, setelah dilakukan dua kali pengujian jangkauan maksimum sistem antara modul beroda dengan modul *remote* adalah 18 meter. Lebih dari jarak tersebut, maka proses transmisi akan mengalami gangguan dengan ditandainya sinyal yang ditransmisikan terkadang dapat ditangkap oleh *receiver* atau terkadang modul penerima tidak dapat menangkap sinyal yang ditransmisikan oleh *transmitter* tersebut. Hal tersebut terjadi karena adanya rugi-rugi daya pada tiap-tiap komponen pendukung dalam modul *remote*.

Selain itu, hal itu terjadi karena sumber daya dari baterai yang digunakan adalah 9V dan antena yang digunakan juga mempengaruhi daya pancar *transmitter* tersebut, sehingga menyebabkan kurang mendukung terciptanya daya pancar yang lebih besar.

## BAB 5

### KESIMPULAN

#### 5.1. Kesimpulan

Setelah merancang sistem, merealisasikan alat, dan mendapatkan hasil pengukuran pada alat skripsi ini, didapatkan beberapa kesimpulan, yaitu :

1. Alat telah berhasil dibuat dan telah berkerja dengan baik.
2. Proses konversi DTMF pada modul *remote* telah berjalan dengan baik.
3. Berdasarkan pengujian penekanan dua tombol pada modul *remote* secara bersamaan, bahwa tombol yang lebih awal tertekan yang akan mengaktifkan sistem.
4. *Transmitter* TLP434 dan *receiver* RLP434 memiliki perbedaan antara spesifikasi dengan hasil pengukuran sebesar 1%.
5. Metode komunikasi yang digunakan antara *transmitter* dengan *receiver* adalah komunikasi *simplex*. Komunikasi ini telah bekerja dengan baik pada sistem.
6. Jangkauan maksimum untuk transmisi modul *remote* dengan modul beroda mampu berfungsi optimal sampai dengan jarak 18 meter. Namun jangkauan optimal transmisi data juga dipengaruhi dengan *noise* dari lingkungan dan kondisi baterai.

## DAFTAR ACUAN

- [1] Hewitt, P.G. Conceptual Physics. California: Addison Wesley Publishing Company, Inc, 1987
- [2] ATMEL. "8-bit AVR microcontroller with 8K bytes in-System Programmable Flash", datasheet, <http://www.atmel.com>
- [3] Sumanto, Mesin Arus Searah. Jogjakarta: ANDI OFFSET, 1994
- [4] Setiawan, Sulhan. Mudah dan Menyenangkan Belajar Mikrokontroler. Yogyakarta: ANDI OFFSET, 2006
- [5] Budiharto, Widodo. Membuat Modul beroda Cerdas. Jakarta: PT Elex Media Komputindo, 2006
- [6] "Radio Remote Control using DTMF" dari internet [www.electronic-circuits-diagrams.com/sd/sd/htm](http://www.electronic-circuits-diagrams.com/sd/sd/htm)
- [7] Lovine, John. Modul berodas, Androids, and Animatrons. New York: McGraw Hill Companies, 2001

## LAMPIRAN

### LAYOUT PCB

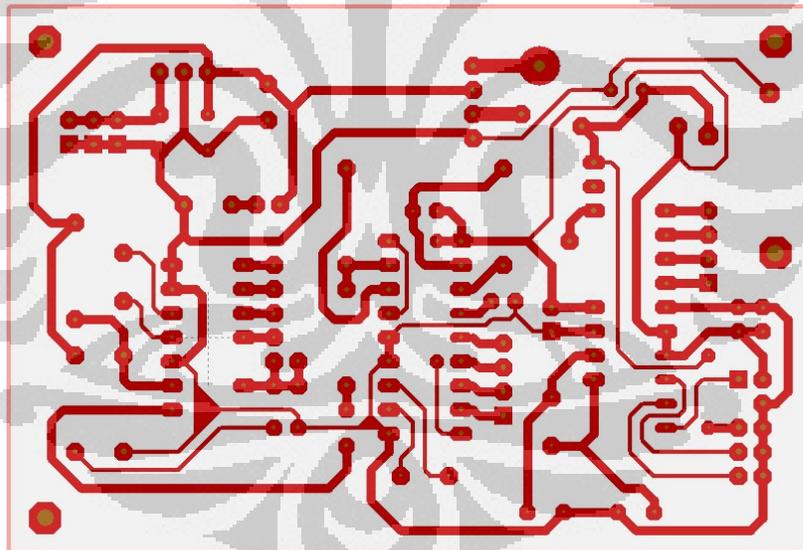
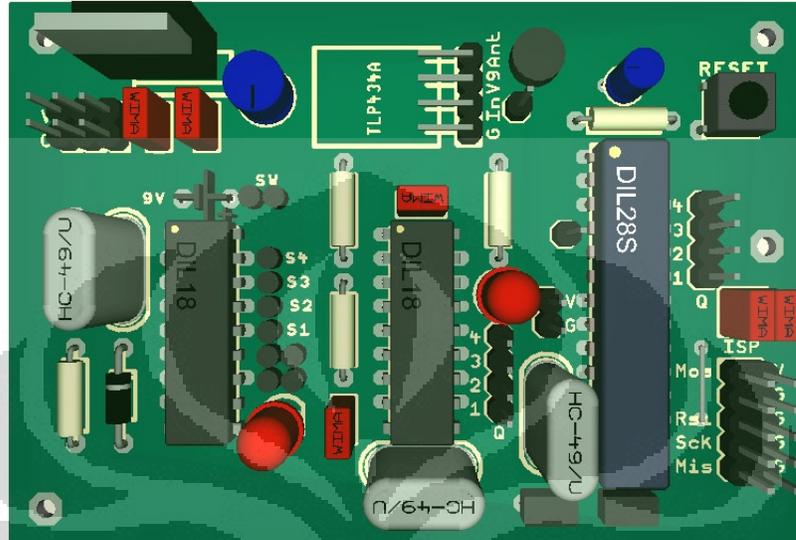


FOTO MODUL REMOTE



FOTO MODUL BERODA



## DAFTAR PROGRAM

=====Modul Beroda (*Receiver*)=====

```
#include <mega8.h>
#include <stdio.h>
#include <delay.h>

/* Defined Variable */
#define PWMA OCR1A
#define PWMB OCR1B
#define ENKA_M1 PORTC.0
#define ENKI_M1 PORTC.1
#define ENKA_M2 PORTC.2
#define ENKI_M2 PORTC.3
#define ON 1
#define OFF 0
#define LED PORTD.1

/* Enumerations */
enum {
    MAJU,
    KANAN,
    KIRI,
    MUNDUR,
    STOP
};

/* Motor Function */
void move(char TYPE, char PWM)
{
    PWMA = PWM;
    PWMB = PWM;

    switch(TYPE) {
        case MAJU :
            ENKA_M1 = 1;
            ENKI_M1 = 0;
            ENKA_M2 = 1;
            ENKI_M2 = 0;
            break;
        case KANAN :
            ENKA_M1 = 1;
            ENKI_M1 = 0;
            ENKA_M2 = 0;
            ENKI_M2 = 1;
            break;
        case KIRI :
            ENKA_M1 = 0;
```

```

        ENKI_M1 = 1;
        ENKA_M2 = 1;
        ENKI_M2 = 0;
        break;
    case MUNDUR :
        ENKA_M1 = 0;
        ENKI_M1 = 1;
        ENKA_M2 = 0;
        ENKI_M2 = 1;
        break;
    case STOP :
        ENKA_M1 = 0;
        ENKI_M1 = 0;
        ENKA_M2 = 0;
        ENKI_M2 = 0;
        break;
    default :
        ENKA_M1 = 1;
        ENKI_M1 = 0;
        ENKA_M2 = 1;
        ENKI_M2 = 0;
        break;
}
}

void main(void)
{
    /* local variable */
    char command;

    /* inialisasi port untuk IO */
    PORTB=0x00;
    DDRB=0x06;
    PORTC=0x00;
    DDRC=0x0F;

    /* setting PWM untuk kecepatan motor */
    TCCR1A=0xA1;
    TCCR1B=0x0B;
    TCNT1H=0x00;
    TCNT1L=0x00;

    /* setting data direction untuk port D bit 1 menjadi Output */
    DDRD.1 = 1;
    PORTD.1 = 1;

    while (1)
    {

```

```

/* Command Parsing */
command = getchar();
switch(command) {
    /* panggil function maju ketika dapet masukan 'w' */
    case 'w' :
        move(MAJU, 250);
        break;
    case 'd' :
        /* panggil function kanan ketika dapet masukan 'd' */
        move(KANAN, 250);
        break;
    case 'a' :
        /* panggil function kiri ketika dapet masukan 'a' */
        move(KIRI, 250);
        break;
    case 's' :
        /* panggil function mundur ketika dapet masukan 's' */
        move(MUNDUR, 250);
        break;
    case 'x' :
        /* panggil function berhenti ketika dapet masukan 'x' */
        move(STOP, 250);
        break;
    default :
        move(STOP, 250);
        break;
}

/* LED Act */
LED = ON;
delay_ms(100);
LED = OFF;
delay_ms(100);
};
}

```

=====Modul Remote (Transmitter)=====

```

/* Library */
#include <mega8.h>
#include <stdio.h>
#include <delay.h>

/* Defined Variable */
#define A    PINC.0
#define B    PINC.1
#define C    PINC.2
#define D    PINC.3

```

```

void main(void)
{
/* local variable */
char command;

/* 4 bits for inputs */
PORTC=0x0F;
DDRC=0x00;

while (1)
{
/* sets command - ini input dari DTMF module */
command = A;
command = (command << 1) | B;
command = (command << 1) | C;
command = (command << 1) | D;

// printf("|command = %x \n\r", command);
// printf("|bit 1 %x bit 2 %x bit 3 %x bit 4 %x \n\r", A,B,C,D);
// command = 0;

#if 1
/* parsing */
switch(command) {

case 0x04 :
putchar('w');
delay_ms(10);
break;

case 0x0A :
putchar('d');
delay_ms(10);
break;

case 0x01 :
putchar('a');
delay_ms(10);
break;

case 0x05 :
putchar('s');
delay_ms(10);
break;

default :
putchar('x');

```

```
        delay_ms(10);  
        break;  
    }  
    command = 0x00;  
#endif  
};  
}
```

