

# **BAB I**

## **PENDAHULUAN**

### **1. Latar Belakang**

Dewasa ini penggunaan teknologi komunikasi semakin dibutuhkan disetiap bidang kehidupan, karena dengan diterapkannya teknologi komunikasi dalam melakukan pekerjaan akan didapatkan berbagai kemudahan. Sehingga lebih efisien dalam penggunaan waktu dan tenaga manusia, melihat kemajuan teknologi komunikasi saat ini yang menggunakan media gelombang radio atau frekuensi radio salah satunya yang akan penulis terapkan berupa suatu alat pendeteksi barang dengan menggunakan frekuensi radio. Mengirim dan menerima data menjadi sangat mudah dan cepat. Faktor murah , efisien tenaga dan menjadi salah satu faktor utama menggunakan fasilitas ini walau jarak yang ditentukan dekat. Namun menjadi mahal dan tidak efisien apabila fasilitas ini digunakan hanya untuk berkomunikasi jarak dekat.

Pengelola data menggunakan mikrokontroler ATMEL 89C51, selain murah dan tersedia dipasaran juga memiliki kemampuan yang tidak kalah dengan mikrokontroler lain dalam keluarga MCS-51. Untuk mengelola data dibutuhkan memori dan untuk mengirim dan menerima data dibutuhkan input dan output ke/dari mikrokontroler. Atas dasar itu dipilih mikrokontroler karena pada mikrokontroler dalam satu chip IC (Integrated Circuit) telah tersedia memori internal dan pin I/O sehingga tidak perlu menambahkannya seperti pada penggunaan mikroprosesor.

Lebih dari itu ada dua hal yang cukup penting yang melatarbelakangi pembuatan makalah tugas akhir ini, penulis mencoba memberikan gambaran sejauh mana penulis dapat menerima, mengerti, dan menerapkan ilmu yang penulis peroleh dibangku kuliah. Kedua, bahwa pembuatan alat ini merupakan langkah awal bagi penulis untuk dapat memanfaatkan alat ini bagi kepentingan masyarakat.

## **2. Tujuan Penulisan**

Maksud dan tujuan penulisan tugas akhir ini merancang alat sistem pendeteksi barang menggunakan pengendali atmel 89C51 dengan display terpisah menggunakan frekuensi radio dan merealisasikan alat sistem pendeteksi barang ini.

## **3. Batasan masalah**

Banyak sistem yang dipakai untuk mengontrol suatu unit peralatan, biasa menggunakan sistem analog (transistor, Op-Amp, dan lain-lain) maupun sistem digital (rangkaiannya gerbang logika, personal computer, mikrokontroler dan lain-lain).

Beranjak dari hal tersebut penulis membatasi masalah pada hal-hal sebagai berikut:

- Sistem control yang dipakai adalah mikrokontroler yaitu AT89C51.
- Dalam pembahasan, penulis memfokuskan pada hal-hal yang bersangkutan dengan alat yang penulis realisasikan.
- Penerima tidak dapat merespon balik dan hanya menerima informasi saja.

#### **4. Sistematika Penulisan**

Secara umum sistematika penulisan tugas akhir yang akan saya buat adalah sebagai berikut :

##### **Bab I Pendahuluan**

Pada bab ini diberikan penjelasan pentingnya topik yang sedang dikerjakan dan latar belakang mengapa penulis memilih topik sistem pendeteksi barang menggunakan pengendali atmel 89C51 dengan display terpisah menggunakan frekuensi radio. Juga dibahas tentang permasalahan seputar sistem pendeteksi barang menggunakan frekuensi radio tersebut.

##### **Bab II Dasar Teori**

Pada bab ini akan dijelaskan tentang sistem signaling antara pengirim dan penerima, mikrokontroler AT89C51 dan beberapa sub sistem yang berkaitan dengan perancangan sistem ini.

##### **Bab III Perancangan dan Realisasi**

Berisikan tentang blok diagram, prinsip kerja alat secara keseluruhan dari tiap-tiap rangkaian dan flowchart program yang digunakan pada pemrograman AT89C51.

#### **Bab IV Pengukuran dan Analisa**

Bab ini menerangkan metode cara pengukuran serta alat-alat yang digunakan dalam proses pengukuran secara ringkas, selain itu dicantumkan hasil pengukuran beserta analisisnya.

#### **Bab V Kesimpulan dan saran**

Pada bab ini akan disimpulkan hasil-hasil percobaan sesuai dengan tujuan yang hendak dicapai.

#### **Bab VI Daftar Pustaka**

Bab ini berisikan referensi tentang system pendeteksi barang menggunakan pengendali atmel 89C51 dengan display terpisah menggunakan frekuensi radio dan datasheet dari komponen yang digunakan.



## **BAB II**

### **TEORI DASAR**

Untuk memahami cara kerja dari alat yang dibuat pada tugas akhir ini, diperlukan pengertian dari beberapa teori dasar yang berhubungan dan mendukung alat ini. Berikut ini akan dijelaskan teori dasar dari komponen pendukung tersebut.

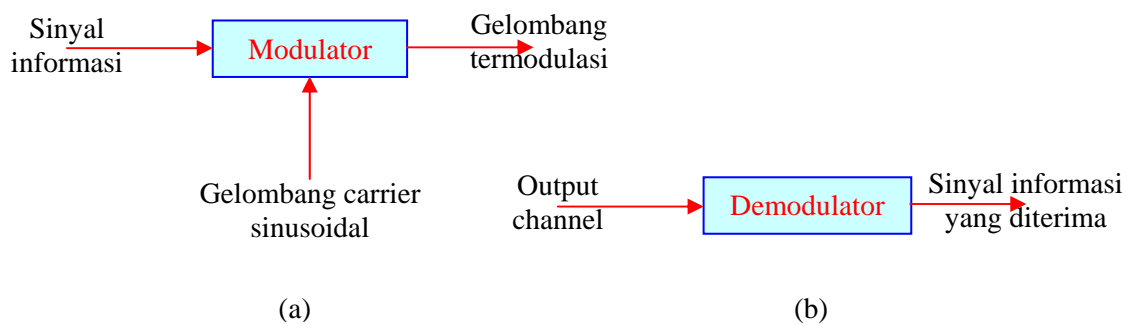
#### **2.1 Dasar Sistem Komunikasi Radio**

Tujuan dari suatu sistem komunikasi adalah mengirimkan sinyal informasi melalui sebuah *channel* komunikasi dimana posisi dari *transmitter* dengan *receiver* mempunyai tempat yang terpisah. Sinyal-sinyal informasi disini didalam ilmu komunikasi dinamakan sinyal-sinyal *baseband* (*baseband signals*). Sinyal *baseband* ini mempunyai lebar pita frekuensi yang mewakili sinyal-sinyal asli dan merupakan sumber dari pada informasi. Tujuan dari *channel* komunikasi adalah untuk menggeser *range* frekuensi *baseband* ke dalam frekuensi lain yang digunakan untuk pengiriman informasi, dan pada *receiver* sinyal yang telah digeser tersebut dikembalikan dengan cara menggeser kembali ke frekuensi asal seperti keadaan sebelum digeser. Sebagai contoh, suatu sistem radio bekerja pada frekuensi 30KHz dan di atasnya, dimana sinyal *baseband* berada pada *range* frekuensi *audio*, sehingga sinyal *baseband* tersebut bergeser pada frekuensi yang berada pada sistem radio tersebut. Pergeseran frekuensi ini dilakukan dengan suatu proses modulasi, dimana parameter dari sinyal *carrier* berubah-ubah menurut perubahan sinyal pemodulasi (sinyal informasi). Secara umum bentuk dari

sinyal *carrier* adalah gelombang *sinusoidal* sedangkan sinyal informasi yang berupa suara adalah gelombang kontinyu. Sinyal *baseband* ini disebut sebagai gelombang pemodulasi (*modulating wave*) sedangkan hasil dari proses modulasi disebut sebagai gelombang yang dimodulasi (*modulating wave*).

Modulasi merupakan proses terakhir dari pengiriman informasi yang digunakan pada sistem komunikasi. Sedangkan pada *receiver* proses akhirnya adalah mendapatkan kembali sinyal *baseband* (informasi) yang asli untuk diterima oleh *user*. Proses penerimaan ini yang pernah kita ketahui sebagai proses demodulasi (*demodulation*) yang mana merupakan kebalikan dari proses modulasi (*modulation*).

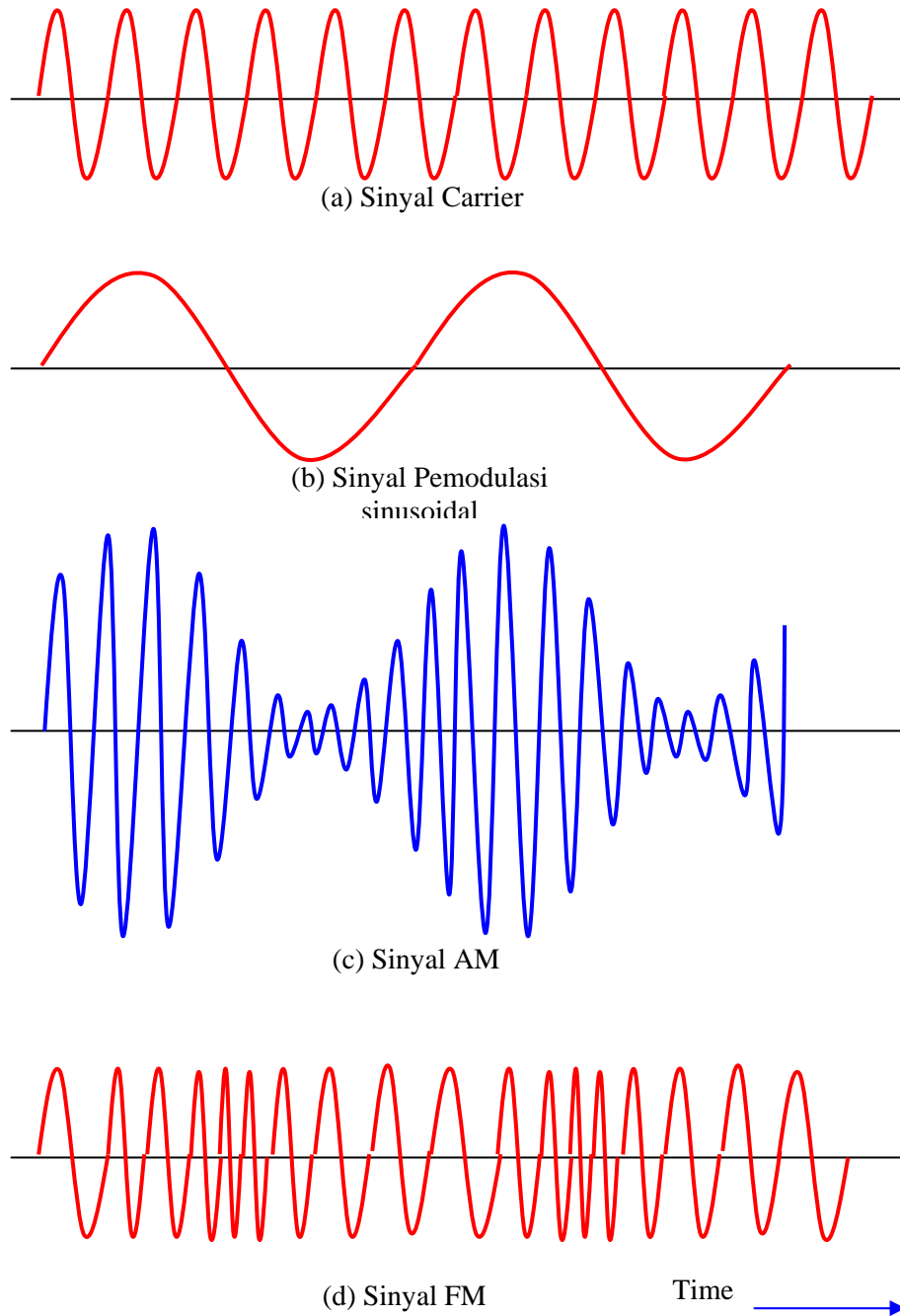
Gambaran dasar blok diagram dari pemrosesan sinyal, berupa *transmitter* pada sistem komunikasi *analog* yang berupa sebuah *modulator* dan sebuah *receiver* yang berupa *demodulator* seperti gambar 2.1 berikut:



(a) Transmitter dan (b) receiver

Gambar 2.1 Komponen Sistem Modulasi Gelombang Kontinyu

Sedangkan bentuk gelombang *output* pada sistem modulasi gelombang kontinyu yang dikenal dengan istilah modulasi amplitudo (*amplitude modulation / AM*) dan modulasi sudut (*angle modulation*) diperlihatkan pada gambar 2.2.



Pada Gambar 2.3 bagian (a) dan (b) menampilkan bentuk sinyal *carrier* dan sinyal informasi, sedangkan (c) dan (d) menampilkan perbedaan antara sinyal



termodulasi amplitudo (*AM*) dan sinyal termodulasi frekuensi (*FM*) dimana sinyal termodulasi frekuensi merupakan bentuk dari modulasi sudut (*angle modulated*).

## **2.1 Modulasi dan Demodulasi**

Modulasi adalah suatu proses pencampuran sinyal informasi/gelombang yang dimodulasi dengan gelombang pembawa (gelombang yang memodulasi) sehingga di dapat gelombang yang termodulasi. Adapun maksud dari diadakannya modulasi, yaitu untuk memindahkan frekuensi, biasanya dari frekuensi rendah ke frekuensi tinggi. Pemindahan frekuensi ini dimaksudkan agar suatu sinyal dapat dikirimkan ke jarak yang jauh, dimana sinyal tersebut ditumpangkan pada suatu sinyal pembawa yang mempunyai frekuensi yang jauh lebih tinggi dari sinyal yang dibawa tersebut. Sedangkan demodulasi adalah kebalikan dari modulasi, yaitu suatu proses untuk mendapatkan atau mendeteksi kembali sinyal yang dikirimkan semula, atau dengan kata lain proses pemisahan sinyal informasi dengan sinyal pembawa (*carrier*).

## **2.3 Mikrokontroler**

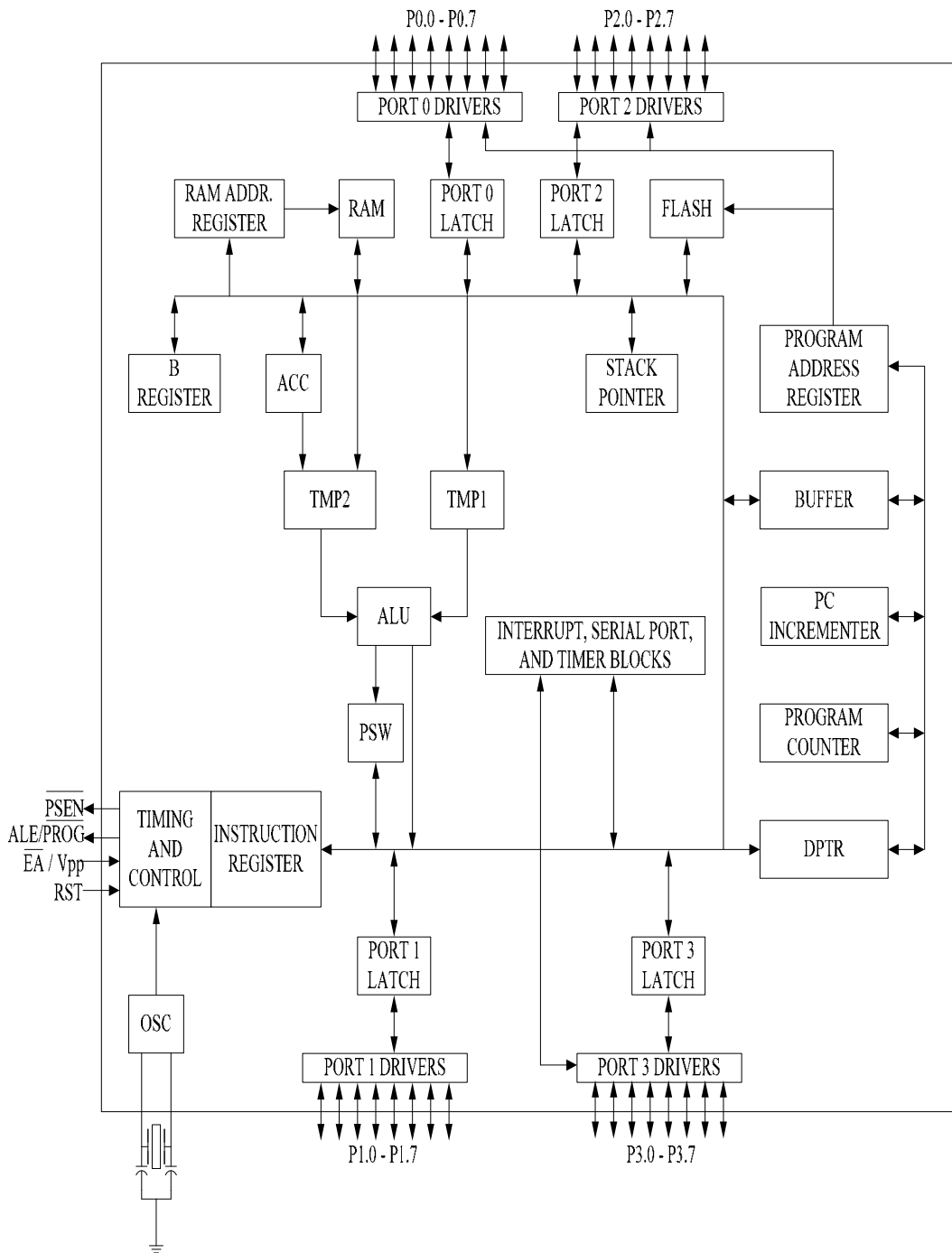
Perbedaan antara mikroprosesor, mikrokomputer serta mikrokontroler sangat penting, sebelum mempelajari ketiganya secara mendalam. Mikroprosesor adalah Central Processing Unit (CPU) dari sebuah komputer, tanpa memori, I/O unit, dan periferal yang dibutuhkan oleh suatu sistem lengkap. Sebagai contoh 8088 dan 80X86 adalah suatu mikroprosesor yang membutuhkan perangkat pendukung berupa RAM, ROM serta I/O unit.

Bila sebuah mikroprosesor dikombinasikan dengan I/O unit dan memori (RAM/ROM), akan dihasilkan sebuah mikrokomputer. Pada kenyataannya mengkombinasikan CPU dengan memori dan I/O unit dapat juga dilakukan dalam level chip, yang menghasilkan Single Chip Microcomputer (SCM) untuk membedakannya dengan mikrokomputer. Untuk selanjutnya karena fungsinya SCM dapat disebut mikrokontroler.

#### **2.4 Mikrokontroler AT89C51**

Mikrokontroler adalah mikroprosesor yang dikembangkan dengan menambahkan I/O, memori (RAM/ROM) yang dilakukan dengan level chip, yang menghasilkan single chip mikrokomputer atau mikrokontroler, salah satunya adalah AT89C51. Arsitektur AT89C51 ditunjukkan pada gambar 2.1. Unit pengolah pusat (Central Processing Unit) mengendalikan seluruh operasi pada mikrokontroler. Unit ini terbagi atas dua bagian, yaitu unit pengendali CU (Control Unit) serta unit aritmatika dan logika ALU (Arithmetic logic Unit). Didalam CPU terdapat berbagai register, misalnya register address counter atau instruction pointer, yang bertugas menyimpan alamat data atau instruksi yang akan diambil. Keuntungan yang di dapat dengan menggunakan mikrokontroler adalah :

- Harganya lebih ekonomis di banding kontroler tipe lainnya
- Ukuran sistemnya yang ringkas
- Pemakaian daya yang rendah
- Keamanan datanya terjamin
- Mudah digunakan untuk merancang sistem yang berbasis mikrokontroler



**Gambar 2.1** Arsitektur Mikrokontroler AT89C51

## **2.4.1 Struktur Memori**

AT89C51 mempunyai struktur memori yang terdiri atas :

- RAM Internal, memori sebesar 128 Byte yang biasanya digunakan untuk menyimpan variabel atau data yang bersifat sementara.
- Special Function Register, memori yang berisi register-register yang mempunyai fungsi-fungsi khusus yang disediakan oleh mikrokontroler tersebut seperti, timer, serial dan lain-lain.
- Flash PEROM, memori yang digunakan untuk menyimpan instruksi-instruksi MCS-51.

AT89C51 mempunyai struktur memori yang terpisah antara RAM Internal dan Flash PEROM-nya. RAM Internal di alamat oleh RAM Address Register, sedangkan Flash PEROM yang menyimpan perintah-perintah MCS-51 di alamat Program Address Register. Dengan adanya struktur memori yang terpisah tersebut walaupun RAM Internal dan Flash PEROM mempunyai alamat yang sama yaitu 00H, namun secara fisik kedua memori tersebut tidak saling berhubungan.

### **2.4.1.1 RAM Internal**

RAM Internal terdiri atas :

- Register Bank

AT89C51 mempunyai 8 buah register yang terdiri dari atas R0 hingga R7, kedelapan buah register ini selalu terletak pada alamat 00H hingga 07H

- Bit Addressable RAM

RAM pada alamat 20H hingga 2FH dapat di akses secara pengalamat bit (Bit Addressable) sehingga hanya dengan sebuah instruksi saja setiap bit dalam area ini dapat di set, clear, AND dan OR.

- General Purpose RAM

RAM keperluan umum di mulai dari alamat 30H hingga 7FH dan dapat di akses melalui pengalamatan langsung maupun tidak langsung.

### 2.4.1.2 Special Function Registers (SFR)

Gambar 2.2 menunjukkan pemetaan dari daerah memori yang disebut dengan Special Function Register (SFR). SFR yang ditandai dengan (...). AT89C51 mempunyai 21 Special Function Register yang terletak pada alamat 80H hingga FFH, beberapa dari register-register ini juga mampu di alamati dengan pengalamatan bit sehingga dapat dioperasikan seperti yang ada pada RAM yang lokasinya dapat di alamati dengan pengalamatan bit.

| 8Bytes |         |         |          |          |       |       |      |    |   |
|--------|---------|---------|----------|----------|-------|-------|------|----|---|
| F8     |         |         |          |          |       |       |      | FF | F |
| F0     | B       |         |          |          |       |       |      | F7 | 7 |
| E8     |         |         |          |          |       |       |      | EF | F |
| E0     | ACC     |         |          |          |       |       |      | E7 | 7 |
| D8     |         |         |          |          |       |       |      | DF | 7 |
| D0     | PSW     |         |          |          |       |       |      | D7 | F |
| C8     | (T2CON) | (T2MOD) | (RCAP2L) | (RCAP2H) | (TL2) | (TH2) |      | CF | 7 |
| C0     |         |         |          |          |       |       |      | C7 | F |
| B8     | IP      |         |          |          |       |       |      | BF | 7 |
| B0     | P3      |         |          |          |       |       |      | B7 | F |
| A8     | IE      |         |          |          |       |       |      | AF | F |
| A0     | P2      |         |          |          |       |       |      | A7 | 7 |
| 98     | SCON    | SBUF    |          |          |       |       |      | 9F | F |
| 90     | PI      |         |          |          |       |       |      | 97 | 7 |
| 88     | TCON    | TMOD    | TL0      | TL1      | TH0   | TH1   |      | 8F | F |
| 80     | P0      | SP      | DPL      | DPH      |       |       | PCON | 87 | 7 |

**Gambar 2.2** Pemetaan Spesial Function Register

Fungsi dari masing-masing register dijelaskan pada bagian berikut :

- **Accumulator**

Register ini terletak pada alamat E0H, hampir semua operasi logika selalu menggunakan register ini. Untuk proses pengambilan dan pengiriman data ke memori eksternal juga diperlukan register ini.

- **Register B**

Register B digunakan pada operasi perkalian dan pembagian. Pada instruksi-instruksi yang lain berfungsi seperti register umumnya.

- **Program Status Word (PSW)**

Register 8 bit berisi informasi status program pada mikrokontroler setelah suatu operasi dijalankan.

- **Stack Pointer (SP)**

Stack Pointer terdiri dari 8 bit, yang terletak pada alamat 81H., SP digunakan untuk menyimpan alamat data yang ditandai ketika pemanggilan sub program maupun interupsi.

- **Data Pointer (DPTR)**

DPTR yang berada pada alamat 83H untuk high byte (DPH) dan 82H untuk low byte (DPL), fungsi utamanya adalah sebagai tempat alamat 16 bit.

- **Register Control**

SFR IP, IE, TMOD, TCON, T2CON, SCON, dan PCON berisi bit kontrol dan status untuk sistem interupt, timer/counter, dan port serial.

- **Port 0 – 3**

AT89C51 mempunyai 4 buah port yaitu, Port 0 (P0), Port 1 (P1), Port 2 (P2), dan Port 3 (P3) yang terletak pada alamat 80H, 90H, A0H dan B0H. Semua port ini dapat diakses dengan pengalamatan secara bit sehingga dapat dilakukan perubahan output pada tiap-tiap pin pada port ini tanpa mempengaruhi pin-pin yang lainnya.

- **Register Timer**

Pasangan register (TH0 & TL0), (TH1 & TL1), serta (TH2 & TL2) adalah register 16 bit untuk proses perhitungan Timer/Counter 0, 1, dan 2.

- **Serial Data Buffer**

Serial Data Buffer sebenarnya merupakan 2 register yang terpisah, transmit buffer (untuk mengirim data serial) dan receive buffer (untuk menerima data serial).

- **Register Interupsi**

AT89C51 mempunyai 5 buah interupsi dengan dua level prioritas interupsi, Interupsi akan selalu non aktif setiap kali sistem di reset. Register-register yang berhubungan dengan interupsi adalah Interrupt Enable Register pada alamat A8H untuk keaktifan tiap-tiap interrupt dari Interrupt Priority Register pada alamat B8H.

#### **2.4.1.3 Flash PEROM**

AT89C51 mempunyai 4Kb Flash PEROM, yaitu ROM yang dapat di tulis ulang atau di hapus menggunakan sebuah perangkat programmer. Flash PEROM dalam AT89C51 menggunakan *Atmel's High Density Non Volatile Technology* yang mempunyai kemampuan untuk di tulis ulang hingga 1000 kali dan berisikan perintah standar MCS-51.

Program yang ada pada Flash PEROM akan dijalankan jika pada saat sistem di reset, pin EA/VP berlogika satu sehingga mikrokontroler aktif berdasarkan program yang ada pada Flash PEROM-nya. Namun, jika pin EA/VP berlogika nol, mikrokontroler aktif berdasarkan program yang ada pada memori eksternal.

Pada kondisi tertentu dapat juga terjadi dibutuhkan untuk memanggil program yang terletak pada memori internal walaupun saat pertama kali di reset, mikrokontroler memanggil program yang ada di eksternal memori atau pun sebaliknya. Hal ini dapat juga dilakukan dengan menggunakan instruksi LCALL (address) atau LJMP (address).

#### **2.4.2 Struktur dan Cara Kerja Port**

AT89C51 mempunyai 4 port bidirectional (Port 0 - Port 3), yang masing-masing terdiri dari 8 bit. Setiap port terdiri dari sebuah latch (Special Function Registers P0 sampai P3), sebuah output driver, dan sebuah input buffer. Penjelasan dari masing-masing port adalah sebagai berikut :

- Port 0

Port 0 adalah port paralel 8 bit dua arah yang tidak dilengkapi dengan rangkaian resistor pull up, port 0 dapat digunakan untuk low order address/data bus ketika mengakses memori eksternal dan memori data. Pada keadaan ini port 0 memiliki resistor pull up internal, port 0 terdapat pada kaki 32 sampai 39.

- Port 1, 2 dan 3

Port 1, 2 dan 3 adalah port paralel 8 bit dua arah yang telah dilengkapi dengan rangkaian resistor pull up. Port 1 terdapat pada kaki 1 sampai 8, port 2 dapat digunakan untuk high order address/data bus ketika mengakses program eksternal dan



memori data, port 2 terdapat pada kaki 21 sampai 28. Port 3 memiliki fungsi pengganti seperti pada Tabel 2.1, port 3 terdapat pada kaki 10 sampai 17.

**Tabel 2.1** Fungsi Pengganti Port 3

| <b>Pin</b> | <b>Fungsi Alternatif</b>                |
|------------|---|
| P3.0       | RXD (port input serial)                 |
| P3.1       | TXD (port input serial)                 |
| P3.2       | INT0 (interupt eksternal)               |
| P3.3       | INT1 (interupt eksternal)               |
| P3.4       | T0 (input timer / counter 0 eksternal ) |
| P3.5       | T1 (input timer / counter 1 eksternal ) |
| P3.6       | WR (sinyal tulis data memori eksternal) |
| P3.7       | RD (sinyal tulis data memori eksternal) |

- **Reset**

Kaki reset terdapat pada kaki 9. Reset merupakan aktif tinggi, yaitu mikrokontroler akan reset jika kaki reset menerima pulsa transisi dari rendah ke tinggi, reset dapat dilakukan secara manual maupun otomatis saat power diaktifkan (Reset On Power).

- **ALE/PROG**

ALE/PROG terdapat pada kaki 30. ALE (Address Latch Enable) digunakan untuk menahan alamat memori eksternal selama pelaksanaan instruksi, ALE hanya akan aktif pada saat mengakses memori eksternal.

- PSEN

PSEN (Program Store Enable) merupakan sinyal pengontrol yang memperbolehkan memori program eksternal masuk ke dalam bus selama proses pemberian/pengambilan instruksi (fetching), PSEN terdapat pada kaki 29.

- EA/VPP

EA/VPP terdapat pada kaki 31. Pada kondisi low, pin ini akan berfungsi sebagai EA, yaitu mikrokontroler akan menjalankan program yang ada pada memori eksternal setelah sistem di reset. Jika berkondisi high, pin ini akan berfungsi menjalankan program yang ada pada memori internal. Pada saat programming pin ini akan mendapat tegangan 12 volt (Vpp).

- XTAL 1 dan XTAL 2

Adalah kaki masukan ke rangkaian osilator internal.

- Ground

Ground terdapat pada kaki 20.

- Vcc

Vcc yang diberikan pada AT89C51 adalah sebesar 5 volt dan terdapat pada kaki 40.

### **2.4.3 Interupsi**

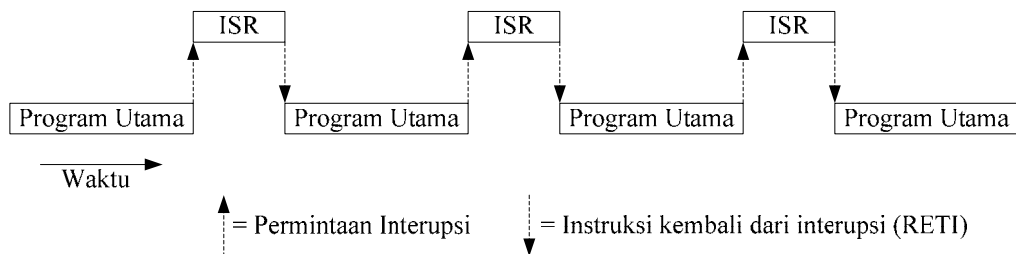
Meskipun memerlukan pengertian yang lebih mendalam, pengetahuan mengenai interupsi sangat membantu mengatasi masalah pemrograman mikroprosesor / mikrokontroler dalam hal menangani banyak peralatan input / output. Pengetahuan mengenai interupsi tidak cukup hanya di bahas secara teori saja, diperlukan contoh program yang konkrit untuk memahami.

Saat kaki reset pada IC mikroprosesor/mikrokontroler menerima sinyal reset (pada MCS-51 sinyal tersebut berupa sinyal '1' sesaat, pada prosesor lain umumnya merupakan sinyal '0' sesaat), Program Counter di isi dengan sebuah nilai. Nilai tersebut dinamakan sebagai vektor reset (reset vector), merupakan nomor awal memori-program yang menampung program yang harus dijalankan.

Pembahasan di atas memberi gambaran bahwa proses reset merupakan peristiwa perangkat keras (sinyal reset diumpankan ke kaki reset) yang di pakai untuk mengatur kerja dari perangkat lunak, yakni menentukan aliran program prosesor (mengisi Program Counter dengan vektor reset).

Program yang dijalankan dengan cara reset, merupakan program utama bagi prosesor. Peristiwa perangkat keras yang di pakai untuk mengatur kerja dari perangkat lunak, tidak hanya terjadi pada proses reset, tapi terjadi pula dalam proses interupsi.

Program yang dijalankan dengan cara interupsi, dinamakan sebagai program layanan interupsi (ISR - Interrupt Service Routine). Saat prosesor menjalankan ISR, pekerjaan yang sedang dikerjakan pada program utama sementara ditinggalkan, selesai menjalankan ISR prosesor kembali menjalankan program utama, seperti yang digambarkan dalam Gambar 2.3.



**Gambar 2.3** Bagan Kerja Mikrokontroler Menangani Interupsi

Sebuah prosesor bisa mempunyai beberapa perangkat keras yang merupakan sumber sinyal permintaan interupsi, masing-masing sumber interupsi di layani dengan ISR berlainan, dengan demikian prosesor mempunyai beberapa *vektor interupsi* untuk memilih ISR mana yang dipakai melayani permintaan interupsi dari berbagai sumber. Kadang kala sebuah *vektor interupsi* dipakai oleh lebih dari satu *sumber interupsi* yang sejenis, dalam hal semacam ini ISR bersangkutan harus menentukan sendiri *sumber interupsi* mana yang harus dilayani saat itu.

Jika pada saat yang sama terjadi lebih dari satu *permintaan interupsi*, prosesor akan melayani *permintaan interupsi* tersebut menurut prioritas yang sudah ditentukan, selesai melayani *permintaan interupsi* prioritas yang lebih tinggi, prosesor melayani *permintaan interupsi* berikutnya, baru setelah itu kembali mengerjakan *program utama*.

Saat prosesor sedang mengerjakan ISR, bisa terjadi *permintaan interupsi* lain, jika *permintaan interupsi* yang datang berikutnya ini mempunyai prioritas lebih tinggi, ISR yang sedang dikerjakan ditinggal dulu, prosesor melayani permintaan yang prioritas lebih tinggi, selesai melayani interupsi prioritas tinggi prosesor meneruskan ISR semula, baru setelah itu kembali mengerjakan *program utama*. Hal ini dikatakan sebagai *interupsi bertingkat (nested interrupt)*, tapi tidak semua prosesor mempunyai kemampuan melayani interupsi seperti ini.

#### **2.4.4 Bahasa Assembly**

Bahasa assembly adalah bahasa komputer tingkat rendah, merupakan bahasa yang bermain di angka-angka 0 dan 1 dan kumpulan 0 dan 1 tersebut akan membentuk suatu instruksi yang akan dijalankan oleh komputer dengan efek-efek tertentu.

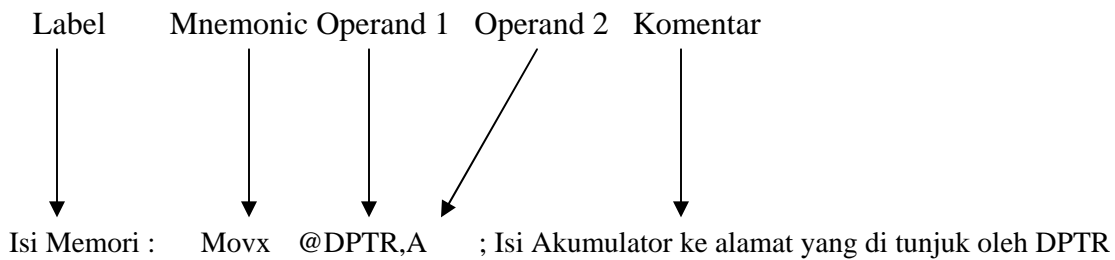
Keunggulan bahasa tingkat rendah ini adalah kecepatan proses program dan kecilnya ukuran program. Sedangkan hambatan pada pemrograman bahasa tingkat rendah ini adalah keruwetan yang ditimbulkan dalam penyusunan program, karena pada bahasa tingkat rendah ini tidak mudah melakukan sesuatu dengan 1 atau 2 perintah, tetapi diperlukan beberapa perintah.

#### **2.4.4.1 Program Sumber Assembly**

Program sumber assembly merupakan program yang di tulis oleh pembuat program berupa kumpulan baris-baris perintah dan biasanya di simpan dalam extension .ASM. Program ini di tulis menggunakan perangkat lunak-perangkat lunak teks editor seperti Notepad atau Editor DOS.

Program yang di tulis dalam bahasa assembly pada Notepad atau Editor DOS yang terdiri dari label kode mnemonic dan operand, biasa di sebut sebagai *source code* belum bisa di terima oleh prosesor untuk dijalankan sebagai program, tetapi harus diterjemahkan ke dalam bahasa mesin dalam bentuk kode biner dengan menggunakan program assembler. Hasil kerja program assembler adalah “program obyek” dan juga assembly listing. Program obyek berisikan kode-kode bahasa mesin inilah yang diumpankan ke memori program prosesor. Dalam dunia mikrokontroler biasanya program obyek ini diisikan ke UV EPROM, dan khusus mikrokontroler buatan Atmel, program ini diisikan ke dalam Flash PEROM yang ada di dalam chip AT89C51. Assembly listing merupakan naskah yang berasal dari program sumber, dalam naskah tersebut pada bagian sebelah setiap baris dari program sumber di beri tambahan hasil terjemahan program assembler. Tambahan tersebut berupa nomor memori program

berikut dengan kode yang akan diisikan pada memori program yang bersangkutan. Naskah ini sangat berguna untuk dokumentasi dan sarana untuk menelusuri program yang di tulis.



**Gambar 2.4** Bentuk Program Sumber Assembly

Seperti yang tampak pada Gambar 2.4, program sumber assembly terdiri dari beberapa bagian, yaitu :

- Label

Label berfungsi jika terjadi lompatan-lompatan atau pencabangan ke suatu alamat tertentu, pemberian nama label ini terserah programmer.

- Mnemonic

Mnemonic di sebut juga kode operasi adalah kode-kode yang akan dikerjakan oleh program assembler yang ada pada komputer atau pun mikrokontroler, kode operasi yang digunakan bergantung pada jenis mikrokontroler yang digunakan.

- Operand

Operand merupakan pelengkap mnemonic, jumlah operand yang dibutuhkan oleh mnemonic tidak sama, ada yang butuh satu, dua, tiga atau pun tidak sama sekali.

- Komentar

Bagian komentar tidak mutlak ada dalam sebuah program, namun bagian ini sering kali dibutuhkan untuk menjelaskan proses-proses kerja atau pun catatan-catatan tertentu pada bagian program.

#### **2.4.4.2 Assembler Directive (Pengaruh Assembler)**

Assembler directive (pengarah assembler) merupakan mnemonic yang akan di proses oleh Program Assembler, dan pengarah assembler ini terdiri dari berbagai macam bentuk. Berikut adalah pengarah assembler yang biasa digunakan untuk program-program assembler.

##### **2.4.4.2.1 Pengarah assembler kontrol kondisi**

- **ORG (Origin)**

ORG digunakan untuk menunjukkan lokasi memori tempat instruksi atau perintah yang ada dibawahnya di simpan.

Contoh : `ORG 2000H`

`MOV DPTR,#4000H`

Disini, perintah `MOV DPTR,#4000H` yang berada di bawah `ORG 2000H` akan terletak di alamat 2000H.

- **END**

END biasanya diletakkan di akhir baris dari file program assembly sebagai tandaakhir pernyataan (*statement*) bagi program assembler dalam melakukan proses assembly.

- **USING**

USING digunakan oleh program assembler (ASM51) untuk menentukan register bank yang aktif sehingga lambang-lambang yang didefinisikan terlebih dahulu oleh

program assembler, seperti AR0 hingga AR7 dapat menunjuk sesuai dengan alamat yang ditentukan oleh register bank yang aktif.

```
Contoh : USING      3
          Push      AR7
```

Perintah Push AR7 yang diawali dengan using 3 menunjukkan bahwa R7 yang terletak di Register Bank 3, yaitu 1FH sehingga setelah dilakukan proses assembly, perintah tersebut akan menghasilkan kode operasi yang sama dengan Push 1FH yaitu C0 1F.

#### **2.4.4.2.2 Pengarah assembler definisi lambang**

- EQU (Equate)

EQU digunakan untuk mendefinisikan sebuah simbol atau lambang assembler secara bebas.

```
Contoh : Bilangan  EQU  50H
          MOV  A,#Bilangan
```

Pada perintah di atas, akumulator di isi dengan konstanta 'bilangan'. Konstanta ini telah didefinisikan sebelumnya dengan nilai 50H dengan menggunakan assembler directive EQU.

- SET

Sama dengan pengarah EQU, pengarah SET digunakan untuk mendefinisikan lambang assembler secara bebas. Perbedaan dengan pengarah EQU adalah pengarah ini dapat di definisi ulang.



Contoh : POINTER SET R0 ; Lambang pointer didefinisikan sama dengan R0

POINTER SET R1 ; Lambang pointer kembali didefinisikan sebagai R1

- BIT

Pengarah bit digunakan untuk mendefinisikan suatu lambang yang menunjuk ke suatu lokasi bit pada memori yang dapat dialamati secara pengalamatan bit.

Contoh :

Flag Bit 0 ; Lambang flag menunjuk ke lokasi 0 secara pengalamatan bit.

- CODE

Pengarah CODE digunakan untuk menyatakan alamat yang dilokasikan dalam area memori program dengan sebuah lambang. Nilai dari alamat tersebut tidak boleh melebihi 65535 atau FFFFH.

Contoh : RESET CODE 0

EXT10 CODE RESET + (1024/16)

- DATA

Pengarah DATA digunakan untuk menyatakan secara langsung alamat pada memori internal dengan sebuah lambang. Alamat yang di mulai dari 00 hingga 7FH menunjukkan RAM Internal dan alamat 80H hingga FFH menunjukkan alamat dari Register Fungsi Khusus.

Contoh :

BUFFER DATA 32H ; Lambang buffer menunjukkan alamat 32H

- IDATA

Pengarah IDATA digunakan untuk menyatakan alamat RAM Internal dalam sebuah lambang dengan pengalamatan tidak langsung. Alamat tersebut dapat di mulai dari 00 hingga FFH.

Contoh : BUFFER IDATA 30H  
          BUFFER2 IDATA BUFFER+1

- XDATA

XDATA digunakan menyatakan alamat yang ada pada memori eksternal dalam sebuah lambang. Nilai dari alamat ini tidak boleh melebihi 65535 atau FFFFH.

Contoh : SektorEksternal1 XDATA 2000H  
          SektorEksternal2 XDATA SektorEksternal1+2000H

- RET

Instruksi ini biasa digunakan pada saat kembali dari subroutine yang di panggil dengan instruksi seperti ACALL atau LCALL.

- RETI

Instruksi ini digunakan untuk kembali ke program utama setelah selesai melaksanakan interupsi.

#### **2.4.4.2.3 Pengarah assembler penyedia memori dan pengarah penyimpanan**

- DB (Define Byte)

DB digunakan untuk memberi nilai tertentu pada memori di lokasi tersebut.

Contoh : ORG 3000H

DB 50H, 51H

Pengarah assembler DB terletak di bawah ORG 3000H. Oleh karena itu, data 50H dan 51H akan menempati lokasi di alamat 3000H dan 3001H.

- DW (Define Word)

DW biasanya diikuti dengan label dan berfungsi untuk memberi nilai 2 byte pada lokasi memori tempat DW dituliskan. Nilai 2 byte adalah nilai yang diberikan berasal dari alamat tempat label diberikan.

Contoh :

```
ORG 2100H
```

Lokasi1 :

```
ORG 3000H
```

```
DW Lokasi1 ; Alamat 3000H dan 3001H diisi dengan data 21H dan 00H
```

Pengarah assembler DW pada listing ini diikuti dengan label lokasi1. Lokasi1 terletak di alamat 2100H. Setelah Proses Assembly dilakukan, data 21H dan 00H akan diletakkan di alamat 3000H dan 3001H pada hasil assembly.

- DS (Define String)

Pengarah assembler ini selalu diikuti dengan angka dan berfungsi untuk menyediakan tempat sebesar angka tersebut, mulai dari alamat assembler directive ini berada.

Tempat yang disediakan selalu terletak pada RAM Internal.

Contoh :

```
ORG 50H
```

```
Memori1 DS 2 ; Disediakan tempat sebesar 2 byte  
; mulai dari alamat 50H dari RAM
```

Pada contoh di atas, pengarah assembler ini terletak di alamat 50H dari RAM Internal. Oleh karena itu, mulai dari alamat 50H tersedia tempat sebesar 2 byte yang menempati alamat 50H dan 51H.

- DBIT

Pengarah assembler ini digunakan untuk menyediakan tempat pada segmen bit dan dapat juga digunakan jika segmen yang aktif adalah BSEG.

Contoh :

BSEG                               ; Pilih bit segmen

PetaIO : DBIT     32     ; Sediakan 32 bit buffer untuk I/O

#### **2.4.4.3 Operasi Logika**

Ekspresi ini digunakan untuk memprmudah pembuat program dalam pemberian nilai pada operand yang memerlukan proses operasi logika terlebih dahulu. Operator-operator tersebut terdiri atas :

ORL    untuk operasi logika OR

ANL    untuk operasi logika AND

XRL    untuk operasi logika EXOR

CPL    untuk operasi logika komplement

Contoh :     ANL   A,P1   ; AND kan A dengan P1 simpan hasilnya di A

#### 2.4.4.4 Operasi Khusus

- SHR untuk menggeser ke kanan

Contoh :

MOV A,#00001000b SHR 1 adalah sama dengan

MOV A,#00000100b

- SHL untuk menggeser ke kiri

Contoh :

MOV A,#00001000b SHL 1 adalah sama dengan

MOV A,#00010000b

- HIGH untuk mengambil nilai byte tinggi

Contoh :

MOV A,#HIGH 1234H adalah sama dengan

MOV A,#12H

- LOW untuk mengambil nilai byte rendah

Contoh :

MOV A,#LOW 1234H adalah sama dengan

MOV A,#34H

- 0 untuk operasi yang harus didahulukan

Contoh :

MOV A,#(10+4)\*3, bilangan10 desimal terlebih dahulu dijumlahkan dengan 4 sebelum di kali dengan 3 dengan adanya operator 0.

#### 2.4.4.5 Operasi Aritmatika

- INC A

Instruksi ini menambahkan nilai akumulator A dengan 1 dan hasilnya di simpan di akumulator A.

- INC direct

Instruksi ini menambahkan data yang di RAM Internal yang alamatnya di tunjuk secara langsung dengan 1 dan hasilnya di simpan di alamat tersebut.

Contoh : INC 00H

Data di alamat 00H di tambah dengan 1 dan hasilnya di simpan di alamat 00H.

- INC @R1

Instruksi ini menambahkan data yang alamatnya di tunjuk oleh RI (Register Index) dengan 1 dan hasilnya di simpan di alamat tersebut.

Contoh : INC @R1

Data di alamat yang di tunjuk oleh R1 ditambahkan 1 dan hasilnya di simpan di alamat tersebut, jika R1 berisi 10H maka data di alamat 10H di tambah 1 dan di simpan kembali di alamat 1.

- INC DPTR

Instruksi ini menambahkan nilai DPTR dengan 1 dan hasilnya di simpan di DPTR.

- ADD A,@R1

Instruksi ini menambahkan akumulator dengan data yang berada di alamat RI (ditunjuk RI) dan hasilnya di simpan di akumulator. RI adalah Register Index di mana pada MCS-51 adalah berupa R0 dan R1.

Contoh : ADD A,@R0

Isi akumulator A akan di tambah dengan isi memori RAM Internal yangdi tunjuk oleh R0, jika R0 berisi 05H maka isi alamat 05H pada RAM Internal akan diumlahkan pada akumulator A dan hasilnya di simpan di akumulator.

- DEC A

Instruksi ini melakukan pengurangan pada nilai akumulator A dengan 1 dan hasilnya di simpan di akumulator A.

- DEC direct

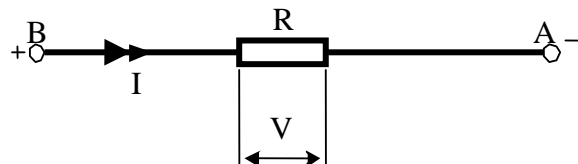
Instruksi ini melakukan penguranga pada data di RAM Internal yang alamatnya di tunjuk secara langsung dengan 1 dan hasilnay di simpan di alamat tersebut.

Contoh : DEC 00H

Data di alamat 00H dikurangi dengan 1 dan hasilnya di simpan di alamat 00H.

## 2.5. Resistor

Resistor akan menghambat aliran arus listrik. Besarnya arus yang mengalir dalam sebuah resistor tergantung pada nilai resistansinya dan beda potensial yang dipasang pada resistor tersebut seperti terlihat pada gambar 2.5. Semakin besar resistor semakin kecil arus yang mengalir. Sebaliknya, semakin besar beda potensial semakin besar arus yang mengalir. Hubungan tersebut dikenal sebagai hukum Ohm :



Gambar 2.5. Resistor pada suatu rangkaian

$$\text{Arus (ampere)} = \frac{\text{Potensial (volt)}}{\text{Resistor (ohm)}} \quad (2 - 1)$$

atau

$$I = \frac{V}{R} \quad , \quad R = \frac{V}{I} \quad , \quad V = R \times I \quad (2 - 2)$$

Pada saat arus mengalir melalui sebuah resistor, maka terdapat sejumlah energi yang terhambur atau terdisipasi menjadi panas. Dan inilah prinsip yang digunakan dalam alat pemanas listrik, yang mengubah energi listrik menjadi energi panas.

Satuan daya listrik adalah watt, yang setara dengan joule setiap detik. Daya (P) memiliki persamaan:

$$P = V \times I \quad (2 - 3)$$

dengan hukum ohm persamaan diatas dapat ditulis sebagai

$$P = R \times I^2 \quad (2 - 4)$$

## **2.6. Kapasitor (kondensator)**

Sebuah komponen elektronik yang tidak mengijinkan arus DC (arus searah) mengalir melaluinya dan mengijinkan arus bolak-balik (AC) mengalir melalui rangkaian sesuai dengan besarnya frekuensi. Reaksi yang ditimbulkan oleh kapasitor terhadap aliran AC disebut reaktansi kapasitif. Reaktansi berbanding terbalik dengan kapasitansi kapasitor.

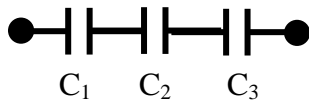
Kapasitor tidak dapat mengubah bentuk gelombang dan frekuensi dari suatu arus yang mengalir melaluinya, tetapi dapat mengubah fase tegangan dan arus.





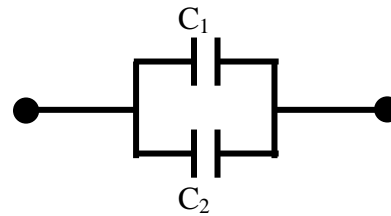
Gambar 2.8. Simbol Kapasitor

1. Seri



$$C = \frac{1}{\frac{1}{C_1} + \frac{1}{C_2} + \frac{1}{C_3}}$$

2. Paralel



$$C = C_1 + C_2$$

## BAB III

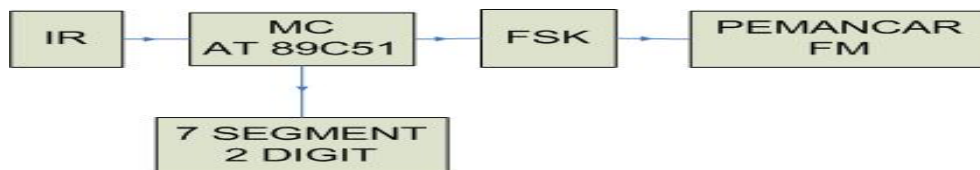
### PERANCANGAN ALAT

Sistem lengkap dari rancangan alat ini terdiri dari bagian pengendali., sensor infra merah, rangkaian mikrokontroler, transmitter, receiver. Pada bagian sensor menggunakan infra merah yang apabila terhalang oleh sebuah benda maka sensor akan memberikan input “0”. Dan untuk pengiriman datanya menggunakan gelombang radio FM. Dalam bab ini akan dibahas mengenai perancangan alat mulai dari blok diagram alat secara keseluruhan serta cara kerja dari setiap blok yang ada.

#### 3.1 Blok Diagram Rangkain

Untuk dapat menganalisa alat yang dibuat, terlebih dahulu dibuat blok diagram rangkaian. Blok diagram rangkain ini dimaksudkan untuk mendapatkan gambaran umum mengenai cara kerja alat. Secara keseluruhan blok diagram alat ini dapat dilihat pada gambar 3.1.1 dan 3.2.

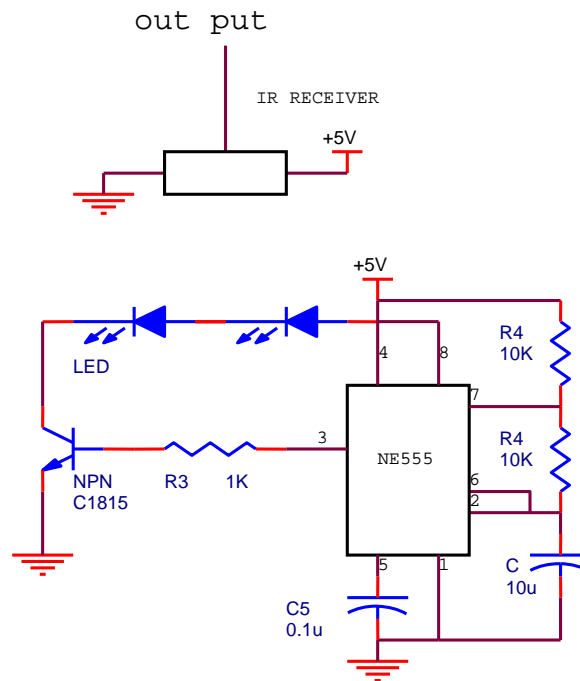
##### 3.1.1 Blok diagram pada display I



**Gambar 3.1** Blok diagram rangkaian pemancar FM

- Sensor infra merah

Fungsi rangkaian sensor ini adalah sebagai pendeteksi apabila ada barang yang masuk, dan pada rangkain infra merah ini menggunakan photodioda sebagai pemancar infra merah dan phototransistor sebagai penerimanya.

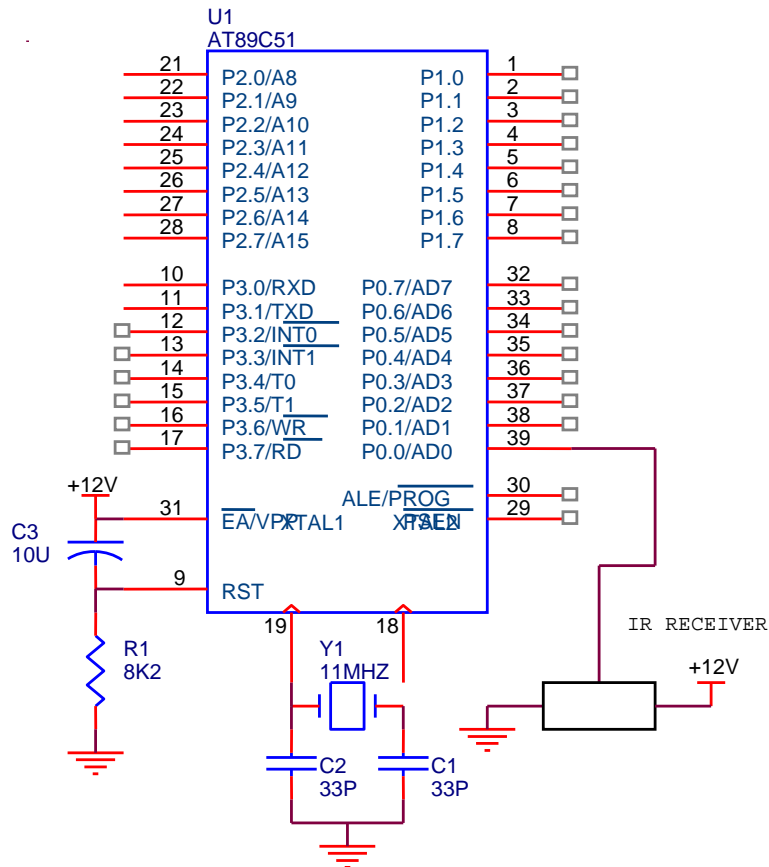


**Gambar 3.2** Rangkaian sensor infra merah

- Rangkaian mikrokontroler AT89C51

Rangkaian mikrokontroler ini merupakan pusat pengolahan data dari pusat pengendali alat. Didalam rangkaian mikrokontroler terdapat empat buah port yang akan digunakan untuk menampung input atau output data dan terhubung langsung dengan rangkaian-rangkaian pengendali. Rangkaian ini terdiri dari: osilator kristal 12 MHz untuk membangkitkan pulsa internal dan dua buah kapasitor sebesar 33

pF yang berfungsi untuk menstabilkan frekuensi. Kapasitor sebesar 10 uF dan resistor 8k2 berfungsi sebagai rangkaian reset.



**Gambar 3.3** Rangkaian mikrokontroler AT89C51

- seven segment

Fungsi dari rangkaian ini adalah untuk menampilkan label atau nomor urut dari barang yang terdeteksi oleh sensor. Proses kerjanya hampir bersamaan waktunya dengan rangkaian pengambilan data pengukuran. Sesaat data barang pertama yang masuk maka seven segment harus menampilkan angka satu (1). Begitu pula untuk barang kedua dan seterusnya hingga barang ke 99.

Dari gambar 3.6 dibawah, terlihat bahwa untuk menggerakan penampilan seven segment digunakan port 1 dari AT 89C51. Komponen seven segment yang digunakan disini memakai konfigurasi Common Anode, yang berarti untuk menghidupkan atau menyalakan LED didalamnya diperlukan logika 0 pada port 1.

Agar penampilan seven segment menampilkan angka 0 maka LED a,b,c,d,e dan f harus dinyalakan pada bit port 1 yang terkait yaitu P1.0, P1.1, P1.2, P1.3, P1.4, dan P1.5 sehingga data yang terkirim adalah 1100.0000 = C0H.

Untuk data selengkapnya dapat dilihat pada table 3.1 dibawah ini.

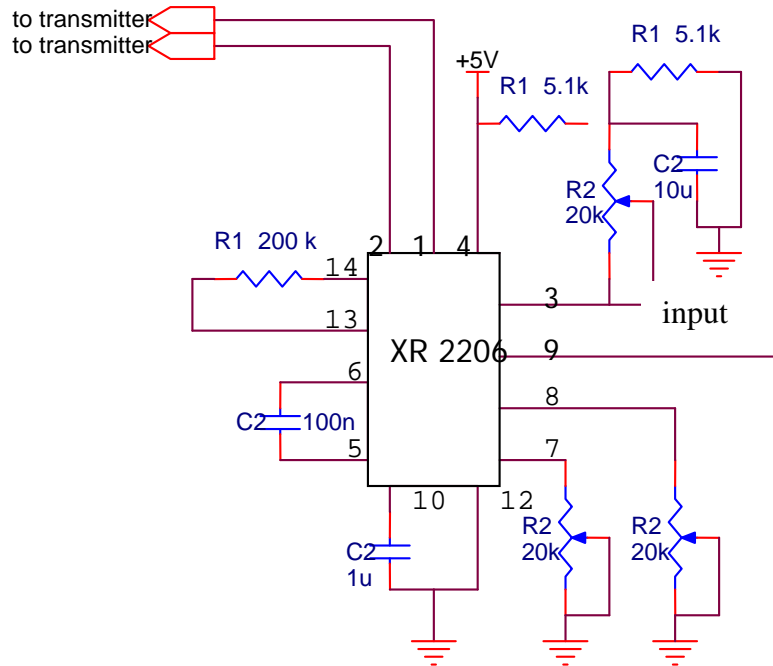
| Tampilan Angka | P1.7 dp | P1.6 g | P1.5 f | P1.4 e | P1.3 d | P1.2 c | P1.1 b | P1.0 a | Heksa |
|----------------|---------|--------|--------|--------|--------|--------|--------|--------|-------|
| 0              | 1       | 1      | 0      | 0      | 0      | 0      | 0      | 0      | C0    |
| 1              | 1       | 1      | 1      | 1      | 1      | 0      | 0      | 1      | F9    |
| 2              | 1       | 0      | 1      | 0      | 0      | 1      | 0      | 0      | A4    |
| 3              | 1       | 0      | 1      | 1      | 0      | 0      | 0      | 0      | B0    |
| 4              | 1       | 0      | 0      | 1      | 1      | 0      | 0      | 1      | 99    |
| 5              | 1       | 0      | 0      | 1      | 0      | 0      | 1      | 0      | 92    |
| 6              | 1       | 0      | 0      | 0      | 0      | 0      | 1      | 0      | 82    |
| 7              | 1       | 1      | 1      | 1      | 1      | 0      | 0      | 0      | F8    |
| 8              | 1       | 0      | 0      | 0      | 0      | 0      | 0      | 0      | 80    |
| 9              | 1       | 0      | 0      | 1      | 0      | 0      | 0      | 0      | 90    |

Tabel 3.1  
daft  
ar heksa

dari tampilan angka pada seven segment

- Modulator FSK

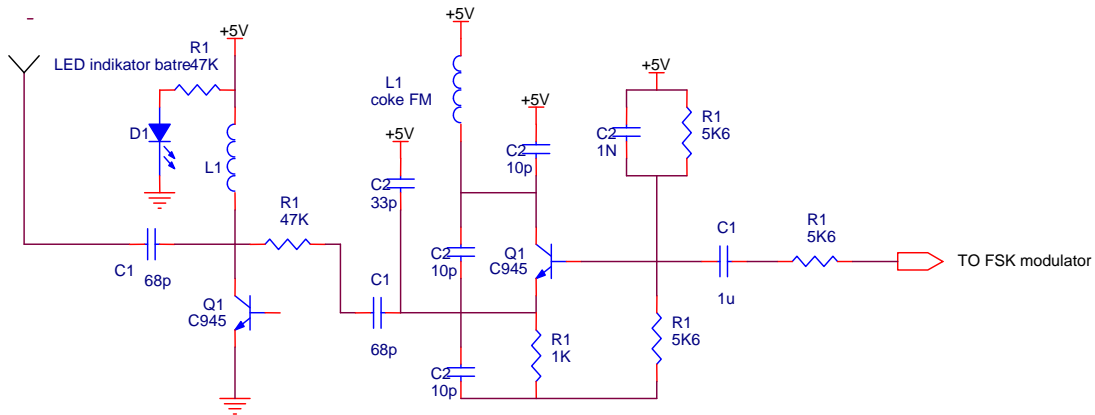
Pada rangkaian modulator ini berfungsi untuk untuk mengubah sinyal digital yang merupakan output dari mikrokontroler menjadi sinyal analog yang akan diteruskan ke pemancar FM.



**Gambar 3.4** Rangkaian modulator FSK

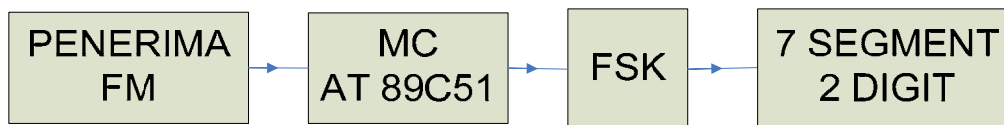
- Pemancar FM

Pada rangkaian pemancar ini berfungsi untuk memancarkan sinyal suara, dimana pada rangkaian ini menggunakan pemancar Frekuensi Modulator (FM).



**Gambar 3.5** Rangkaian pemancar FM

### 3.1.2 Blok diagram pada display II



**Gambar 3.6** Blok diagram rangkaian penerima FM

- Rangkaian penerima FM

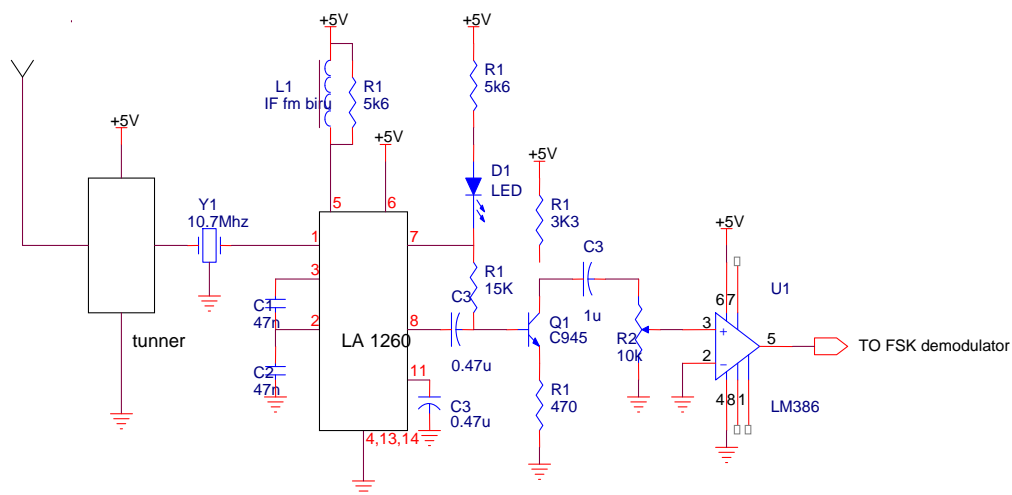
Pada rangkaian penerima menggunakan yang sudah tersedia dipasaran. Rangkaian penerima ini bekerja pada frekuensi broadcast yaitu 85 MHz sampai 108 MHz. Output rangkaian ini ada dua sisi, sisi kanan dan sisi kiri. Tetapi output yang digunakan hanya satu yaitu sisi kanan, mengingat spesifikasinya sudah memenuhi/dapat langsung dikirim ke blok demodulator FSK tanpa harus melalui rangkaian penguat.

Untuk keperluan proyek akhir ini di bagian penerima, frekuensi di tune pada frekuensi 85 MHz agar tidak mengganggu frekuensi yang sudah digunakan oleh stasiun-stasiun radio yang ada.

Komponen penerima FM terdiri dari Tuner FM, keramik filter 10,7 MHz, IC LA1260, transistor dan beberapa komponen pendukung lain seperti resistor dan kapasitor.

Di dalam tuner FM terdiri dari rangkaian penguat RF, osilator lokal, mixer, penguat IF, filter serta rangkaian terpadu yang berfungsi menyeleksi frekuensi IF dengan bantuan lilitan trafo IF FM.

IC LA 1260 berfungsi sebagai detektor yang akan mengembalikan sinyal asli dari masukan IF termodulasi 10,7MHz yang ditala melalui trafo IF, bila hasil tala tersebut sudah sama dengan sinyal masukan IF maka kapasitor 1 mikrofarat yang sama dengan isyarat yang dikirimkan. Output dari kaki 8 IC LA 1260 sudah berbentuk frekuensi audio sehingga tinggal menguatkan sinyal tersebut dengan transistor C 945.

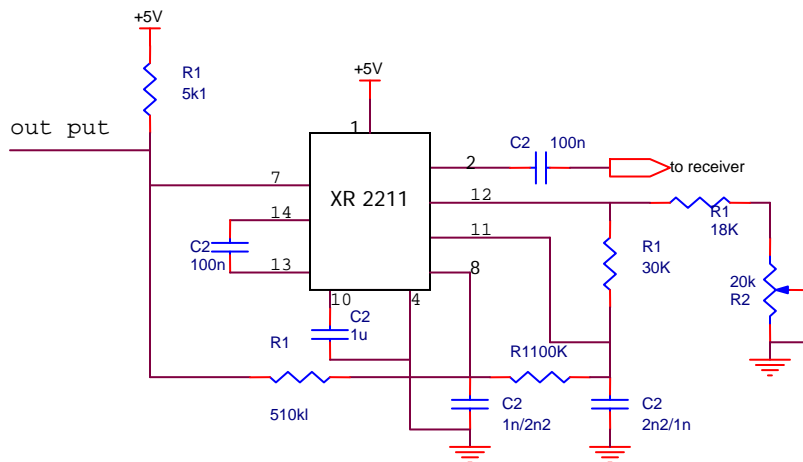




**Gambar 3.7** Rangkaian penerima FM

- Demodulator FSK

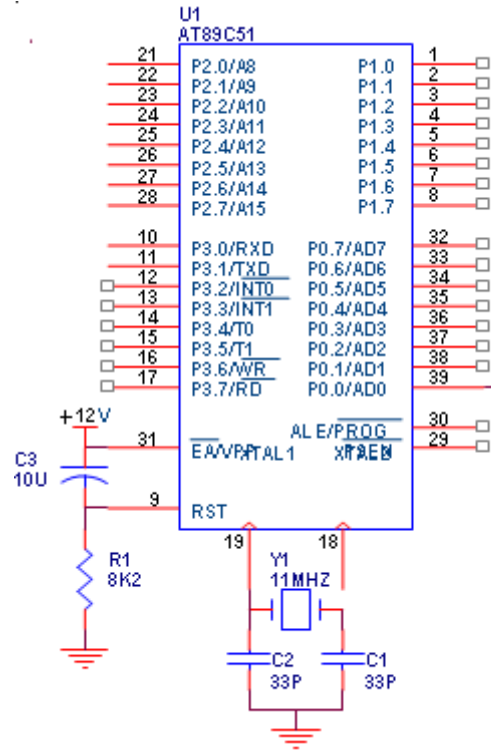
Pada rangkaian demodulator ini berfungsi sebagai perubah sinyal, yaitu dari sinyal analog yang merupakan sinyal keluaran pemancar dan menghasilkan sinyal digital karena proses dari demodulator.



**Gambar 3.8** Rangkaian demodulator FSK

- Rangkaian mikrokontroler AT89C51

Pada rangkain mikrokontroler di bagian penerima mempunyai fungsi yang sama pada mikrokontroler di bagian pemancar, yaitu sebagai pusat pengolahan data dari pusat pengendali alat.

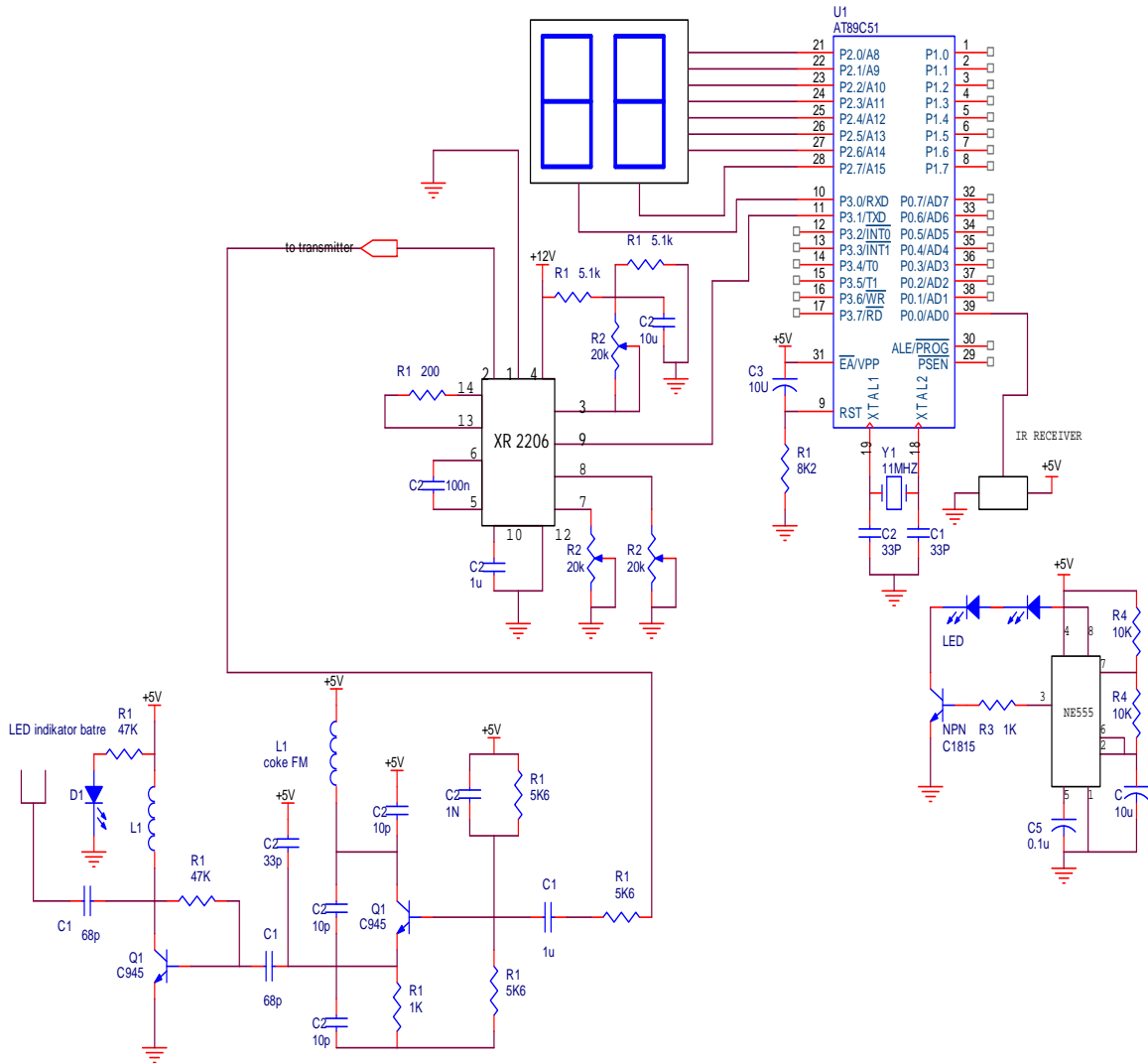


**Gambar 3.9** Rangkaian mikrokontroler AT89C51

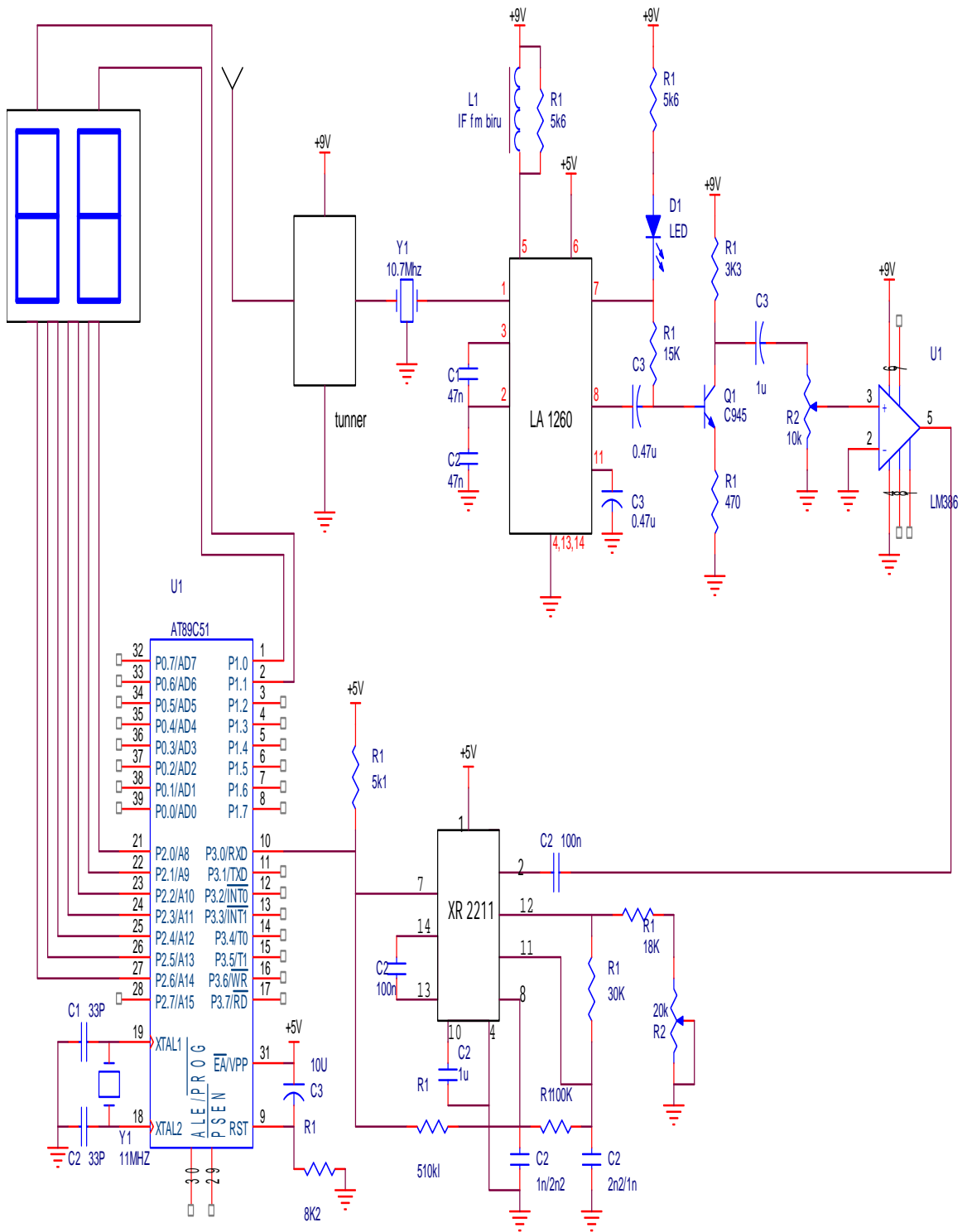
- Rangkaian 7 segment

Pada rangkaian seven segment di penerima juga mempunyai fungsi yang sama pada rangkaian seven segment di pemancar.

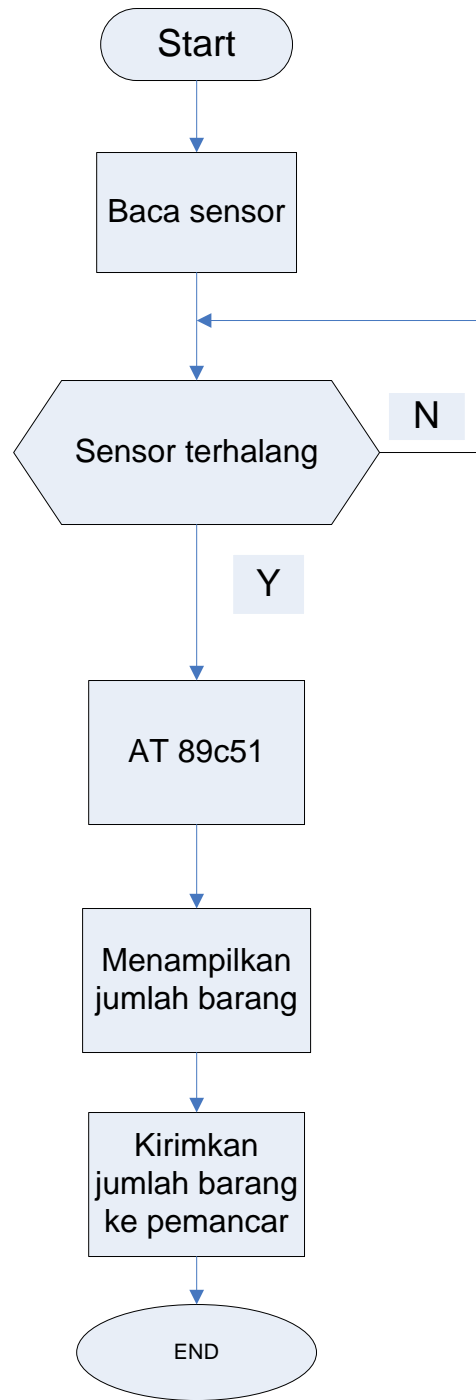
### 3.2 Rangkaian keseluruhan alat



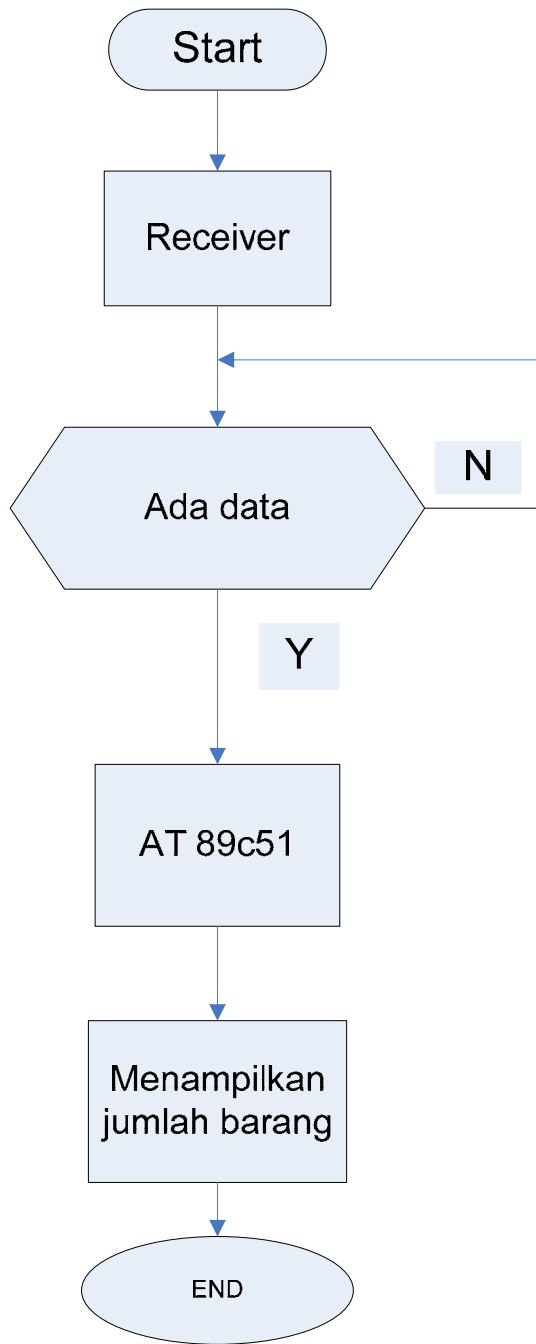
Gambar 3.10 rangkaian pada display I



Gambar 3.11 rangkaian pada display II



**Gambar 3.12** digram flowchat pemancar



**Gambar 3.13** digram flowchat penerima

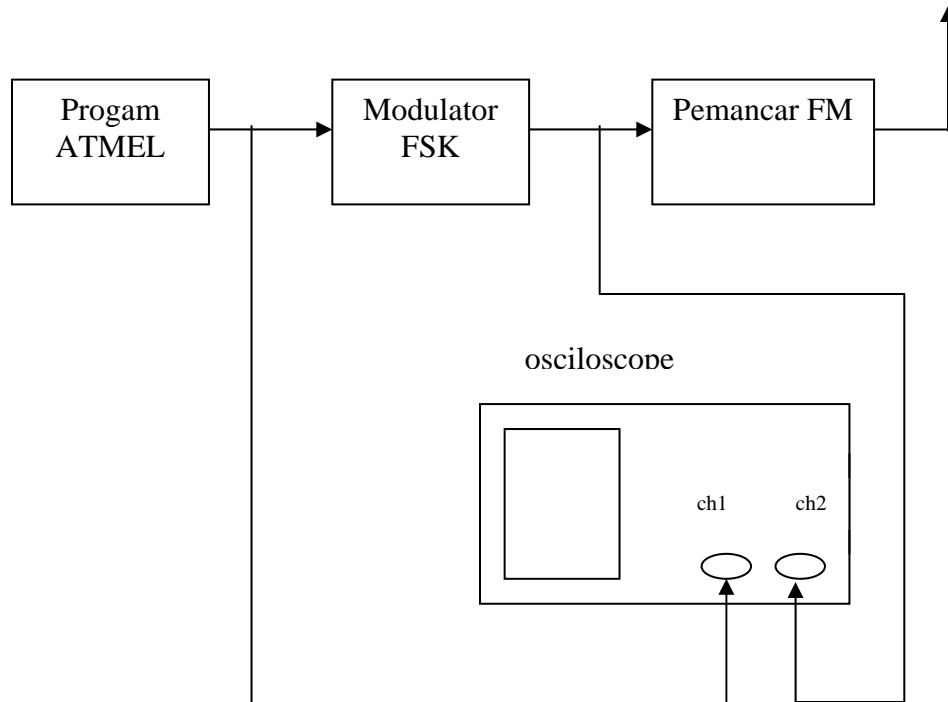
## BAB IV

### PENGUKURAN DAN ANALISA DATA

#### 4.1 Pengukuran modulator FSK

##### 4.1.1 Langkah kerja

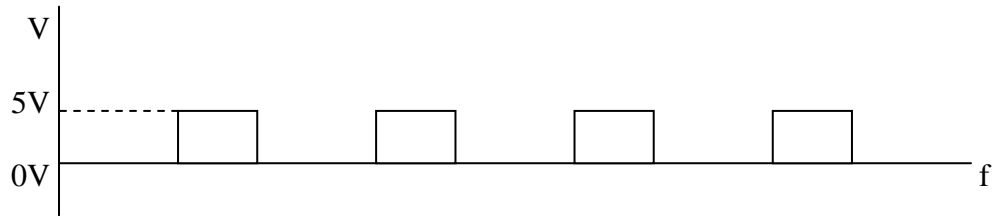
1. Membuat setup pengukuran seperti yang ditunjukkan pada gambar dibawah ini :



Gambar 4.1 Setup pengukuran modulator FSK

2. Mengatur trimpot di bagian modulator FSK sehingga didapatkan sinyal output sama dengan sinyal input dari data di dalam ATMEL. Amplitudo sinyal output dan fasa sinyal modulator FSK dengan data ATMEL harus sama.

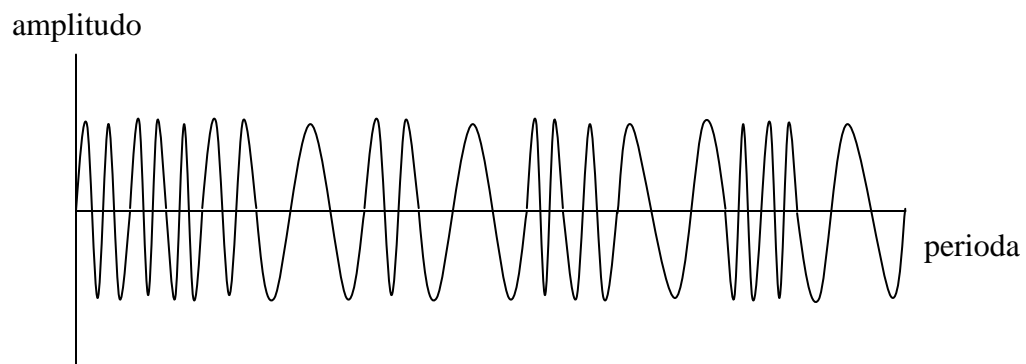
#### 4.1.2 Hasil pengukuran



Gambar 4.2 Input modulator FSK

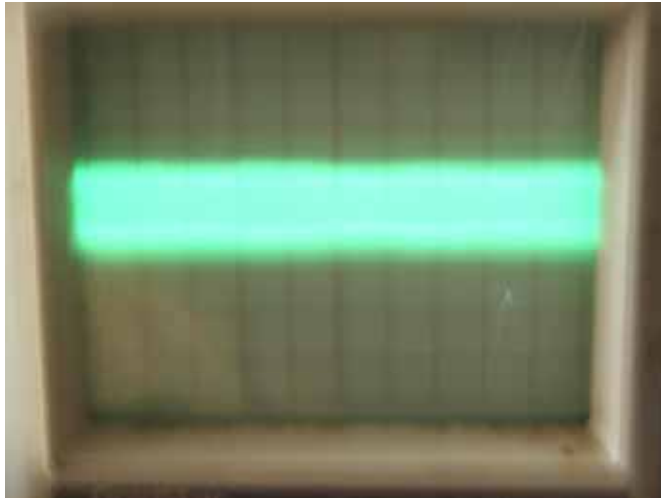


Gambar 4.3 Input modulator FSK dari hasil pengukuran osiloskop



Gambar 4.4 output modulator FSK





Gambar 4.5 output modulator FSK dari hasil pengukuran osiloskop

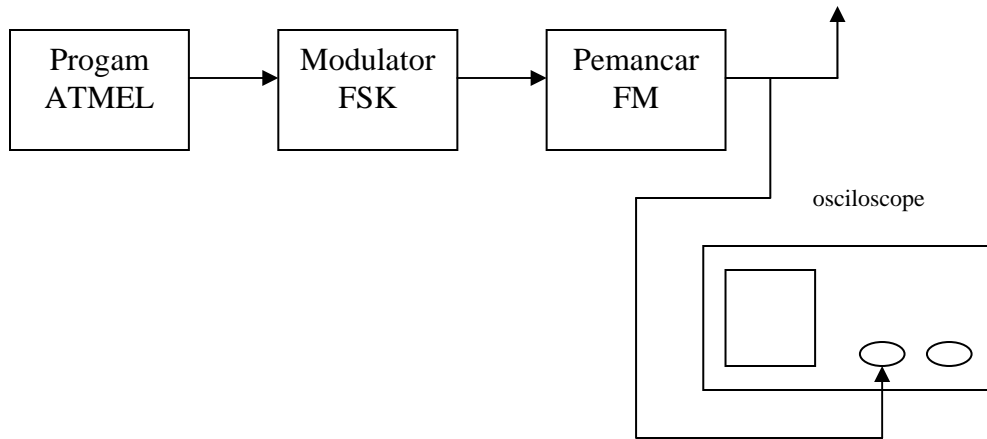
#### 4.1.3 Analisa hasil pengukuran

Dari hasil pengukuran terlihat bahwa sinyal yang masuk kerangkaian modulasi berupa sinyal digital dan terlihat pula dari hasil pengukuran bahwa sinyal yang dihasilkan dari pemodulasian berupa sinyal analog. Ini membuktikan bahwa hasil kerja dari rangkaian modulasi bekerja dengan baik. Pada pengujian osiloskop sinyal terlihat seperti garis tebal, ini dikarenakan pada osiloskop hanya mampu bekerja pada frekuensi maksimum 40 Mhz sedangkan alat yang akan di uji mempunyai frekuensi 85 Mhz.

## 4.2 Pengukuran pemancar FM dengan output dari modulstor FSK

### 4.2.1 Langkah kerja

1. Membuat setup pengukuran seperti yang ditunjukkan pada gambar dibawah ini :

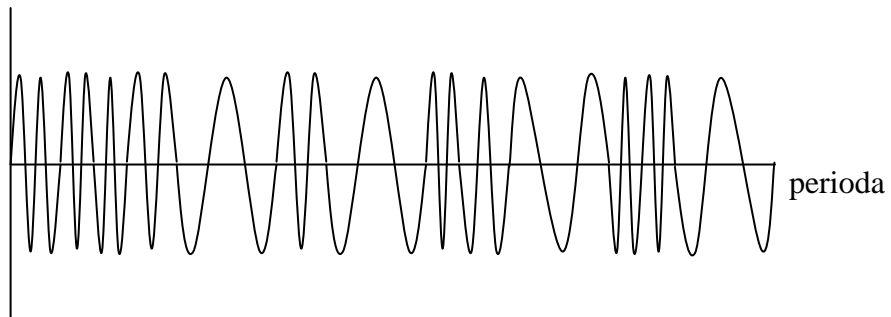


Gambar 4.6 Setup pengukuran pemancar FM dengan output modulator FSK

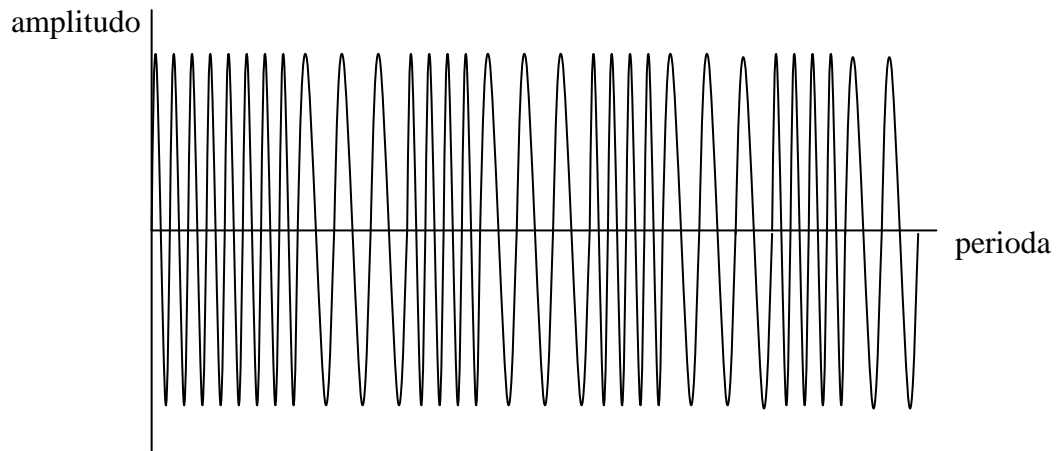
2. Mengamati sinyal output dari pemancar FM.

### 4.2.2 Hasil pengukuran

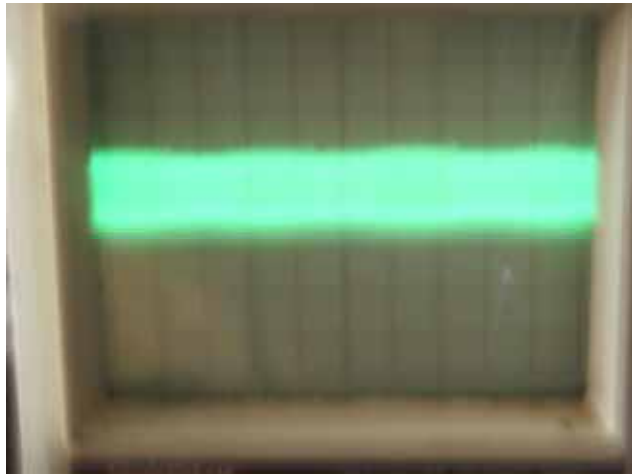
amplitudo



Gambar 4.7 output modulator FSK



Gambar 4.8 Output pemancar FM



Gambar 4.9 Output pemancar FM dari hasil pengukuran osiloskop

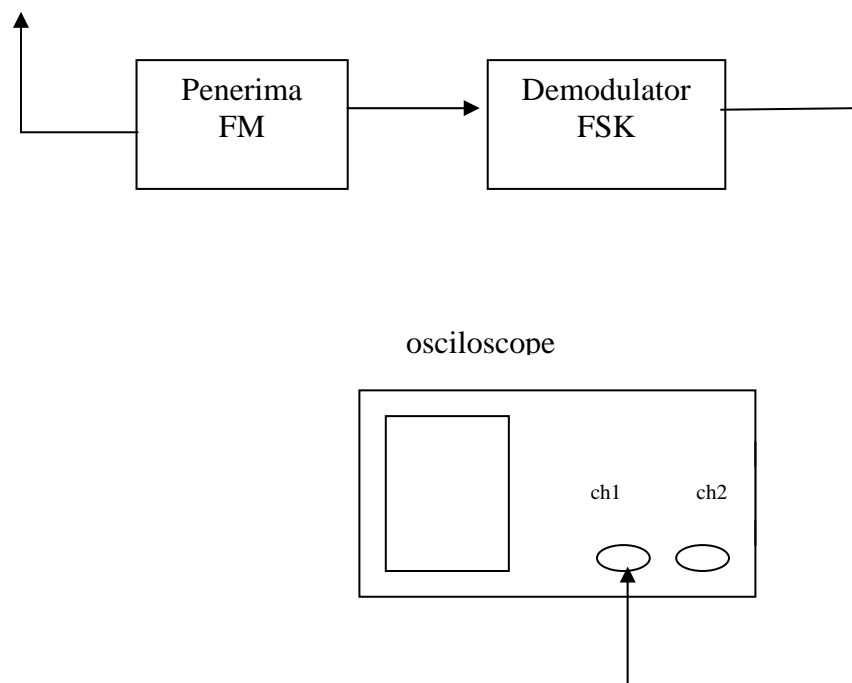
### 4.2.3 Analisa hasil pengukuran

Dari gambar 4.12 diatas dapat disimpulkan bahwa sinyal yang akan dipancarkan telah dimodulasi dan diperkuat agar sinyal dapat diterima di rangkaian penerima dengan baik.

## 4.3 Pengukuran Demodulator FSK

### 4.3.1 Langkah kerja

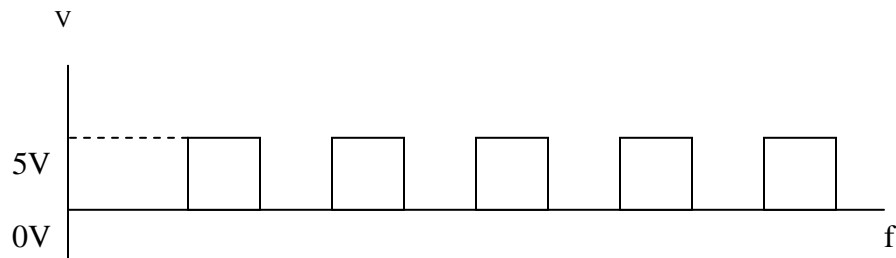
1. Membuat setup pengukuran seperti yang ditunjukkan pada gambar dibawah ini :



Gambar 4.10 Setup pengukuran demodulator FSK

2. Mengatur trimpot di bagian demodulator FSK sehingga didapatkan sinyal output sama dengan sinyal input. Amplitudo sinyal output dan fasa sinyal demodulator FSK dengan input harus sama.

#### 4.3.2 Hasil pengukuran



Gambar 4.11 Output demodulator FSK



Gambar 4.12 Output demodulator FSK dari hasil pengukuran osiloskop

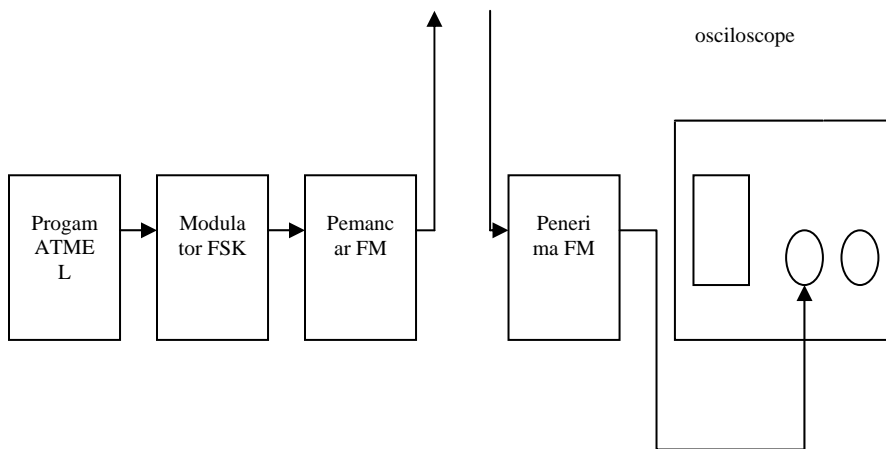
#### 4.3.3 Analisa hasil pengukuran

Pada demodulator sinyal informasi yang ditumpangkan ke sinyal pembawa (carrier) akan dipisahkan kembali sehingga menjadi sinyal digital. Dan sinyal input FSK adalah sinyal TTL begitu pula output demodulator FSK pun adalah sinyal TTL dengan amplitude yang sama.

#### 4.4 Pengukuran penerima FM dengan input dari modulstor FSK

##### 4.4.1 Langkah kerja

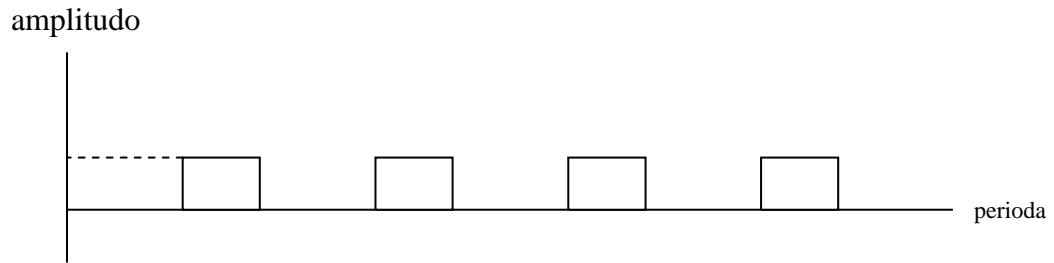
1. Membuat setup pengukuran seperti yang ditunjukkan pada gambar dibawah ini :



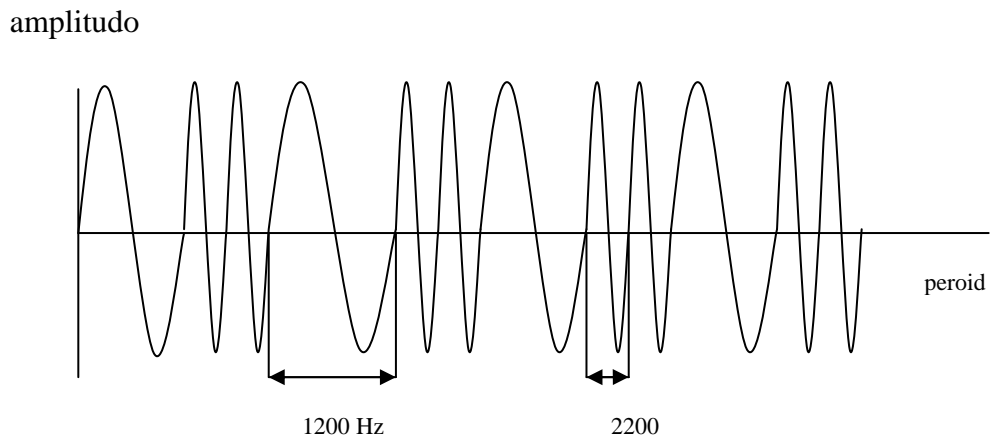
Gambar 4.13 Setup pengukuran penerima Fmdan Modulator FSK

2. Mengamati sinyal output dari penerima FM.

#### 4.4.2 Hasil pengukuran



Gambar 4.14 input modulator FSK



Gambar 4.15 Output penerima FM dengan input dari modulator FSK

#### 4.4.3 Analisa pengukuran

Pengesetan modulator FSK pada frekuensi mark 1200 Hz dan space 1200 Hz di buktikan pula oleh output penerima FM. Perubahan amplitudo pada komunikasi tidak terlalu mempengaruhi, yang penting dalam hal ini yaitu kecepatan informasi. Dari hasil

pengukuran sinyal dengan frekuensi space menunjukkan logik 0 dan sinyal frekuensi mark menunjukkan logik 1.



## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Setelah mengamati mulai dari proses perancangan sampai pengujian akhir dapat diambil kesimpulan sebagai berikut :

1. Alat ini dapat bekerja sesuai dengan frekuensinya yaitu menerima pesan Yang dikirimkan dari bagian pengirim dengan keterbatasan jarak terima sehingga pesan itu masih dapat dimengerti, ini dimungkinkan karena daya pancar pengirim relatif kecil.
2. Saat komunikasi dating berlangsung, output di display bukan seluruhnya berasal dari pengirim karena terdapat sisipan data-data yang tidak dapat dimengerti, dan hal ini dimungkinkan karena sinyal yang berasal dari pemancar lain yang menggunakan frekuensi yang sama.
3. Alat ini bisa bekerja dengan baik apabila pada frekuensi 85 Mhz yang tidak ada pemancar lain yang menggunakan frekuensi tersebut.

#### **5.2 Saran**

Untuk mengembangkan proyek akhir ini, penulis memberikan beberapa saran sebagai berikut :

1. Alat ini bekerja secara simplex (satu arah) maka bias dikembangkan satu tingkat diatasnya yaitu bekerja secara half duflex sehingga bisa saling membalas / mengirim pesan (komunikasi lebih aktif).

2. Frekuensi yang digunakan adalah frekuensi FM sehingga memungkinkan banyak noise yang mengganggu sehingga jarak pun jadi lebih dekat namun jika menggunakan transceiver jenis HT jarak komunikasi bisa lebih jauh.