

3100097008571

# ALAT PENGUKUR KEKASARAN PERMUKAAN JALAN (NAASRA ROUGHNESS METER) YANG TERHUBUNG SERIAL KE IBM-PC

RSKC  
621.3981  
Sub  
A-1  

---

1995



PERPUSTAKAAN	
S 23	
Tgl. No. Inis.	24 NOV 1995
Terkas. F. 1	H
No. Agenda Psp.	5894

Oleh :

**LILIK SUBIYANTO**

**2882200996**

**JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI INDUSTRI  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1995**

**ALAT PENGUKUR KEKASARAN  
PERMUKAAN JALAN (NAASRA  
ROUGHNESS METER) YANG TERHUBUNG  
SERIAL KE IBM-PC**

**TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan  
Untuk Memperoleh Gelar Sarjana Teknik Elektro  
Pada**

**Bidang Studi Elektronika  
Jurusan Teknik Elektro  
Fakultas Teknologi Industri  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui / Menyetujui  
Dosen Pembimbing**



**(Ir. M. MOEFADOL ASYARI)**

**NIP. 130 422 814**

**SURABAYA**

**SEPTEMBER, 1995**

## ABSTRAK

Jalan adalah sarana transportasi darat, berfungsi menunjang pergerakan arus lalu-lintas, sekaligus meningkatkan arus komunikasi di semua bidang, dengan demikian maka perekonomian akan meningkat.

Untuk memenuhi terpeliharanya jalan yang baik, maka secara periodik kemunduran fungsional maupun kemunduran struktural jalan perlu dideteksi. Untuk itu diperlukan alat pengukur kekasaran jalan guna mengetahui kondisi jalan dan penyusunan rencana program lebih lanjut.

Alat pengukur kekasaran jalan yang ada saat ini seperti NAASRA ROUGHNESS METER yang didatangkan dari luar negeri bekerja secara mekanis, sedangkan alat pengukur kekasaran jalan yang dikembangkan pada tugas akhir ini bekerja secara elektronik, memungkinkan pendataan dan pencatatan lebih efektif dan akurat.

Prinsip kerja unit peralatan ini adalah gerak vertikal roda belakang kendaraan akibat tidak rata permukaan jalan, dipindahkan dari penggerak ke alat pencatat elektronik dengan menggunakan sensor infra merah. Alat ini dijalankan dan dikontrol menggunakan minimum sistem 8088 serta dilengkapi dengan komunikasi serial pada PC untuk menampilkan data, grafik dan penyimpanan data.

# KATA PENGANTAR

Puji syukur kami panjatkan kehadirat Allah swt atas ridlo dan rahmat yang dilimpahkanNya, sehingga penyusunan tugas akhir yang berjudul " **Alat Pengukur Kekasaran Permukaan Jalan (NAASRA Roughness Meter) yang Terhubung Serial Ke IBM PC** " ini dapat terselesaikan.

Tugas Akhir ini merupakan salah satu syarat untuk menyelesaikan program studi Strata satu (S1) pada jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Penyelesaian Tugas Akhir ini dilakukan dengan merencanakan dan membuat alat berdasarkan teori yang didapat dari bangku kuliah, literatur, bimbingan dari dosen pembimbing serta pihak lain yang telah memberikan bantuan dan dorongan.

Pada kesempatan ini penulis ingin mengucapkan terima kasih yang tulus atas kesempatan, bantuan serta bimbingan yang telah penulis terima, kepada :

1. Dr. Ir. Moch. Salehudin M.Eng.Sc selaku Ketua Jurusan Teknik Elektro FTI ITS.
2. Ir. Soetikno selaku Koordinator Bidang Studi Elektronika.
3. Ir. M. Moefadol Asyari, selaku dosen wali dan sekaligus sebagai dosen pembimbing .
4. Bp. Iksan, selaku Kepala Bagian Alat dan Perlengkapan Bina Marga TK. I Jawa Timur.
6. Bp. Gomful Dairi Msc., selaku Kepala Bagian Penelitian dan Pengembangan,

Puslitbang Jalan, Bandung.

7. Semua pihak yang banyak membantu sehingga terselesainya Tugas Akhir ini dengan baik.

Penulis menyadari adanya kekurangan dalam pembuatan Tugas Akhir ini oleh karena itu kritik dan saran dari berbagai pihak sangat diharapkan. Besar harapan penulis buku dan alat ini berguna bagi yang membutuhkan.

Surabaya, Nopember 1995

Penyusun

—

STAMPED SIGNATURE

# DAFTAR ISI

<b>JUDUL</b> .....	i
<b>LEMBAR PENGESAHAN</b> .....	ii
<b>ABSTRAK</b> .....	iii
<b>KATA PENGANTAR</b> .....	iv
<b>DAFTAR ISI</b> .....	vi
<b>DAFTAR TABEL</b> .....	x
<b>DAFTAR GAMBAR</b> .....	xi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Permasalahan .....	2
1.3 Tujuan .....	2
1.4 Batasan Masalah .....	2
1.5 Metodologi .....	3
1.6 Sistematika Pembahasan .....	4
1.7 Relevansi .....	4
<b>BAB II TEORI PENUNJANG</b> .....	5
2.1 Perkerasan Jalan Raya .....	5

2.2 Pengukuran Kerataan Jalan Oleh Alat NAASRA Roughness Meter	6
2.3 Sensor Infra Merah	9
2.4 Pencacah (Counter)	10
2.4.1 Pencacah-10 BCD Tak-sinkron	11
2.5 Penggerak Peraga 7-Segment	12
2.6 Minimum Sistem 8088	13
2.6.1 Mikroprosesor 8088	13
2.6.1.1 Konfigurasi Pin 8088	13
2.6.1.2 Register Mikroprosesor 8088	19
2.6.2 Memori	21
2.6.2.1 ROM (Read Only Memory)	22
2.6.2.2 RAM (Random Access Memory)	23
2.7 Decoder	26
2.8 Programable Peripheral Interface 8255	27
2.8.1 Fungsi Pin Pada PPI 8255	29
2.9 Priority Interrupt Controller (PIC) 8259	32
2.9 Prinsip Komunikasi Data Serial	33
2.9.1 Standart Komunikasi RS-232	37
2.9.2 Hubungan RS-232	37
2.9.3 Port Komunikasi Asinkron Pada IBM PC	38
2.9.4 Inisialisasi Port Komunikasi	40
2.9.5 Membaca Data Dari Port Komunikasi	40
2.9.6 Mengirim Data Melalui Port Komunikasi	41

<b>BAB III PERENCANAAN HARDWARE DAN SOFTWARE .....</b>	<b>42</b>
3.1 Sensor Infra Merah .....	43
3.2 Decade Counter .....	46
3.3 Peraga (Display) Dan Penggeraknya .....	48
3.4 Perencanaan Pengukur Kecepatan .....	49
3.5 Central Processing Unit Minimum Sistem 8088 .....	54
3.6 Decoder .....	55
3.7 Programable Peripheral Interface 8255 .....	57
3.8 Rangkaian Interrupt (PIC 8259) .....	58
3.9 Serial Komunikasi Interface .....	59
3.10 Pengoperasian Tombol Dan Peraga .....	61
<b>BAB IV PERENCANAAN MEKANIK DAN INSTALASI .....</b>	<b>64</b>
4.1 Perencanaan Mekanik .....	64
4.2 Instalasi .....	65
<b>BAB V KALIBRASI DAN PENGUKURAN .....</b>	<b>66</b>
5.1 Kalibrasi .....	66
5.2 Pengukuran .....	67
<b>BAB VI PENUTUP .....</b>	<b>74</b>
6.1 Kesimpulan .....	74

6.2 Saran-saran ..... 74

**DAFTAR PUSTAKA** ..... 76

**LAMPIRAN**

## DAFTAR TABEL

Tabel 2.1 Operasi Bus Cycle Mode Minimum .....	18
Tabel 2.2 Pin-pin Pada RS-232 .....	38
Tabel 2.3 Alamat Interface Serial Komunikasi .....	38
Tabel 2.4 Line Control Register Port 3F8h (atau 2F8h, 3E8h, 2E8h) .....	39
Tabel 2.5 Line Status Register Port 3FDh (atau 2FDh, 3EDh, 2EDh) .....	40
Tabel 3.1 Logika Pengalamatan (Decoder) 000h - 003h dan 224h - 227h .....	56
Tabel 5.1 Hasil Pengukuran Simpangan .....	68
Tabel 5.2 Korelasi antara alat yang dibuat dengan alat NAASRA Roughness Meter .....	69
Tabel 5.3 Simpangan antara alat yang dibuat dengan alat NAASRA Roughness Meter.....	71

## DAFTAR GAMBAR

Gambar 2.1	Bentuk 'Kekasaran Permukaan'.....	6
Gambar 2.2	Grafik Korelasi Antara RCI dan IRI di Indonesia .....	9
Gambar 2.3	Sensor Infra Merah .....	10
Gambar 2.4	Rangkaian Pencacah-10 BCD Tak-sinkron .....	11
Gambar 2.5	Logika IC 74ls47 .....	12
Gambar 2.6	Blok Diagram Sistem Minimum .....	13
Gambar 2.7	Konfigurasi Pin Mikroprosesor 8088 .....	14
Gambar 2.8	Timing Diagram dari 8088 .....	15
Gambar 2.9	Register Mikroprosesor 8088 .....	20
Gambar 2.10	Posisi Tiap Bit pada Flag Register .....	21
Gambar 2.11	Timing Diagram Pembacaan Data dari ROM .....	23
Gambar 2.12	Timing Diagram Proses Baca Data dari RAM .....	25
Gambar 2.13	Timing Diagram Penulisan Data ke RAM .....	26
Gambar 2.14	Diagram Blok dari PPI 8255 .....	28
Gambar 2.15	Definisi Dasar Mode dan Interface bus PPI 8255 .....	29
Gambar 2.16	Format Control Word PPI 8255 .....	30
Gambar 2.17	Konfigurasi Pin PPI 8255 .....	31
Gambar 2.18	Internal Blok Diagram PIC 8259 .....	33
Gambar 2.19	Format Initialization Command Word .....	34
Gambar 2.20	Format Operational Command Word .....	35

Gambar 2.21 Format Data Karakter pada Komunikasi Asinkron .....	36
Gambar 3.1 Blok Diagram Perencanaan Hardware .....	42
Gambar 3.2 Sensor Infra Merah dan Komparator LM324 .....	44
Gambar 3.3 Respon Tegangan pada Komparator .....	44
Gambar 3.4 Rangkaian Decade Counter Data Kekasaran .....	47
Gambar 3.5 Rangkaian Decade Counter Pengukuran Jarak Tempuh .....	47
Gambar 3.6 Rangkaian Peraga 7-Segment Untuk kekasaran dan Pengukuran Jarak .....	49
Gambar 3.7 Rangkaian Peraga Untuk Set Jarak .....	50
Gambar 3.8 Astable MV NE 555 sebagai Osilator .....	53
Gambar 3.9 Sinyal pada Rangkaian Pengukur Kecepatan .....	54
Gambar 3.10 Memory Map Minimum Sistem 8088 .....	55
Gambar 3.11 Rangkaian Decoder Alamat 000h-003h .....	56
Gambar 3.12 Rangkaian Decoder Alamat 224h-227h .....	57
Gambar 4.1 Mekanik Alat Ukur .....	64
Gambar 4.2 Instalasi Alat Ukur .....	65

# BAB I

## PENDAHULUAN

### 1.1 LATAR BELAKANG

Jalan merupakan sarana transportasi darat yang sangat penting menunjang pergerakan arus lalu-lintas, sekaligus meningkatkan arus komunikasi disemua bidang dengan demikian maka laju perekonomian akan meningkat.

Pemakaian jalan terus-menerus, padatnya arus lalu-lintas dan faktor alam menyebabkan kemunduran fungsional maupun kemunduran struktural jalan sehingga kenyamanan berkendara menjadi berkurang, lalu-lintas tidak teratur, perjalanan menjadi lambat, biaya operasional kendaraan meningkat yang akan mengakibatkan pula terhambatnya laju perekonomian. Untuk itu instansi yang terkait dalam hal ini mempunyai wewenang perawatan jalan agar jalan tetap terpelihara dengan baik, maka secara periodik dilakukan survai untuk mengetahui kondisi jalan dan melengkapi laporan rencana program yang akan dilaksanakan dan perkiraan rencana untuk tahun-tahun mendatang.

Untuk membantu survai tersebut digunakan suatu alat yang dapat mendeteksi dan mencatat data kekasaran permukaan jalan. Alat pengukur kekasaran jalan yang ada saat ini yaitu NAASRA ROUGHNESS METER yang didatangkan dari luar negeri bekerja secara mekanis dan pencatatan datanya dilakukan secara manual yaitu dicatat oleh seseorang, sehingga memungkinkan kesalahannya pencatatan lebih besar dan survai berjalan lambat karena adanya beberapa kelemahan dari surveyor seperti

kelelahan dan kelengahan. Adanya kekurangan dari alat yang telah ada tersebut dalam tugas akhir ini dibuat alat pengukur kekasaran jalan yang bekerja secara elektronik dan otomatis, sehingga diharapkan hasil pencatatannya lebih akurat dan pelaksanaan survey menjadi lebih cepat dan mudah.

## **1.2 PERMASALAHAN**

Dalam tugas Akhir ini saya mencoba membuat alat Pengukur Kekasaran Jalan yang merupakan perbaikan dari alat yang ada sebelumnya. Pendataan secara elektronik dengan menggunakan minimum sistem 8088 dan diinterfacekan pada PC. Dengan melihat data-data dan grafik yang telah dicatat oleh komputer dapat diketahui kondisi jalan yang disurvei dan sebagai pedoman penyusunan rencana program oleh instansi terkait langkah yang akan diambil terhadap jalan tersebut, metode dan waktu pelaksanaannya. Alat ini diharapkan pula dapat membantu para peneliti dalam penelitian jalan secara umum.

## **1.3 TUJUAN**

Tujuan pembuatan Tugas Akhir ini adalah pembuatan alat untuk mengukur kekasaran permukaan jalan yang diinterfacekan secara serial ke Personal Computer sebagai media penampil dan pencatat data dan grafik pengukuran.

## **1.4 BATASAN MASALAH**

Alat Pengukur Kekasaran Permukaan Jalan yang dibuat pada Tugas Akhir ini hanya digunakan untuk mendeteksi tingkat kekasaran dari permukaan jalan, yaitu kekasaran jalan berupa penonjolan batuan kerikil pada permukaan akibat penyusutan

aspal atau berkurangnya keelastisan aspal, atau kekasaran jalan akibat lendutan aspal dengan kerapatan tinggi (*bleeding*).

Pemakaian alat ini dimaksudkan untuk menghindari kerusakan yang lebih parah dari jalan (tindakan preventif).

## 1.5 METODOLOGI

Langkah-langkah penyusunan tugas akhir ini adalah sebagai berikut:

Mengumpulkan dan mempelajari literatur yang berhubungan dengan masalah yang dihadapi. Literatur ini adalah mengenai Manual Alat Pengukur Kekasaran Permukaan Jalan yang sudah ada NAASRA ROUGHNESS METER, Manual Operasi dan Perawatan Jalan yang diterbitkan oleh Bina Marga, Counter Elektronik, Mikroprosesor, komunikasi data serial, bahasa pemrograman dan perangkat pendukung interface.

Merencanakan perangkat keras yang meliputi: merencanakan rangkaian counter untuk pengukur kekasaran jalan dan pengukur jarak tempuh (*Odometer*), maupun kecepatan (*Speedometer*), Rangkaian pengolah sinyal digital serta penggunaan minimum sistem 8088, serial interfacing RS 232 untuk komunikasi serial ke IBM PC.

Membuat mekanik untuk memindahkan gerak vertikal ke alat pencatat elektronik.

Membuat perangkat keras serta perangkat lunak sesuai dengan hasil pengujian. Memberikan saran-saran pengembangan alat lebih lanjut. Dari langkah-langkah diatas disusun buku laporan Tugas Akhir ini.

## 1.6 SISTEMATIKA PEMBAHASAN

Sistematika pembahasan dari laporan tugas akhir ini disusun sebagai berikut:

Bab I adalah pendahuluan yang berisi tentang latar belakang, permasalahan, pembatasan masalah, tujuan, metodologi, sistematika, dan relevansi.

Bab II adalah teori penunjang membahas mengenai perkerasan jalan raya, pengukuran kekasaran permukaan jalan dengan alat NAASRA Roughness Meter, sensor infra merah, pencacah (*counters*), decoder BCD ke seven segment, minimum sistem 8088, PPI 8255, PIC 8259 dan komunikasi serial RS232.

Bab III adalah perencanaan perangkat keras dan perangkat lunak yang meliputi sensor pada alat ukur dengan infra merah, pencacah sepuluh (*decade counter*) untuk pengukuran kekasaran, pengukuran jarak tempuh dan kecepatan kendaraan, minimum sistem 8088 serta PPI 8255, serta perangkat lunak yang sesuai dengan perangkat keras tersebut.

Bab IV pembuatan mekanik serta perencanaan instalasi pada kendaraan.

Bab V kalibrasi dan pengujian alat terhadap alat pengukur kekasaran permukaan jalan yang telah ada yaitu NAASRA Roughness Meter.

Bab VI adalah penutup yang berisi kesimpulan dan saran pengembangan.

## 1.7 RELEVANSI

Hasil tugas akhir ini diharapkan dapat berguna bagi instansi terkait dalam hal ini Bina Marga sebagai alat bantu survai perawatan dan pemeliharaan jalan atau sebagai penunjang penelitian pengembangan metode pembuatan jalan sebagai sarana transportasi darat.

## BAB II

# TEORI PENUNJANG

### 2.1 PERKERASAN JALAN RAYA

Guna dapat memberikan rasa aman dan nyaman kepada sipemakai jalan, maka konstruksi perkerasan jalan harus memenuhi syarat-syarat tertentu antara lain :

- permukaan yang rata, tidak bergelombang, tidak melendut dan tidak berlubang.
- permukaan cukup kaku, sehingga tidak mudah berubah bentuk akibat beban yang bekerja di atasnya.
- permukaan cukup kasar, memberikan gesekan yang baik antara ban dan permukaan jalan sehingga tak mudah selip.
- permukaan tidak mengkilap, tidak silau jika kena sinar matahari.

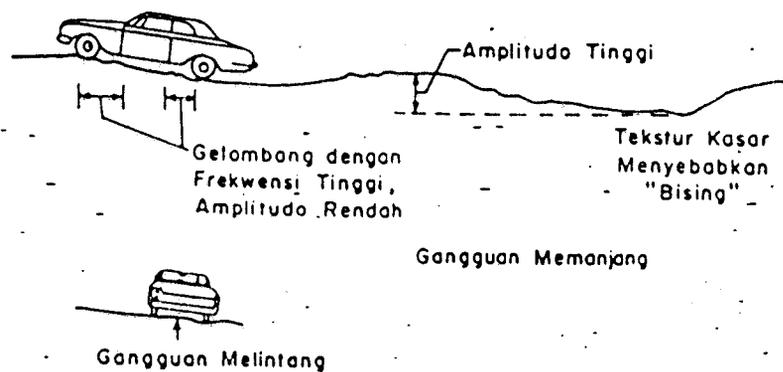
Untuk memenuhi kondisi diatas maka harus didukung pula dengan kekuatan/struktural konstruksi perkerasan jalan diantaranya : ketebalan yang cukup, kedap terhadap air, permukaan mudah mengalirkan air, dan kekuatan memikul beban. Disamping itu tak dapat dilupakan sistem pemeliharaan yang terencana dan tepat selama umur pelayanan.

Penanganan konstruksi perkerasan dapat bersifat pemeliharaan, penunjang, peningkatan, maupun rehabilitasi. Penanganan yang paling sesuai dapat dilakukan dengan baik setelah dilakukan evaluasi terhadap tingkat kekasaran permukaan jalan.. Untuk mendukung dan membantu evaluasi tersebut digunakan data-data dari pengukuran kekasaran jalan dengan menggunakan alat pengukur kekasaran jalan. Untuk keperluan tersebut alat pengukur kekasaran jalan yang sangat akurat dan lebih obyektif dalam mendeteksi tingkat kekasaran

jalan sangat diperlukan.

## 2.2 PENGUKURAN KEKASARAN JALAN OLEH ALAT NAASRA ROUGHNESS METER

Nilai pengukuran kekasaran jalan yang dilakukan oleh alat NAASRA diperoleh dari gerak vertikal antara badan kendaraan dan poros belakang (gardan) yang dikendalikan diatas permukaan jalan yang diukur. Sehingga hasil pengukuran merupakan jumlahan dari getaran yang terjadi dalam satuan Count/Km atau Count/m. Bentuk dari kekasaran permukaan jalan ini dapat berupa gelombang jalan (kekasaran jalan dengan amplitudo tinggi) atau ketidakrataan jalan yang merata (gelombang jalan dengan frekuensi getaran tinggi amplitudo rendah) seperti pada gambar 2.1



Gambar 2.1  
Bentuk 'kekasaran permukaan'

Parameter yang dipakai secara internasional untuk menyatakan kekasaran jalan adalah IRI (*International Roughness Index*) atau *Standart Rougness (SR)* dalam satuan mm/Km, nilai ini

diperoleh dari hasil pengukuran *NAASRA Roughness* (NR) dengan rumus konversi sebagai berikut<sup>1</sup> :

$$SR = 472 + \frac{1473}{b} - \frac{1473 \times a}{b} - \frac{450 \times a \times NR}{b^2} + \frac{225 \times a^2}{b^2} + \frac{225 \times NR}{b^2}$$

dimana :

SR = Standart Roughness ( mm/Km )

NR = NAASRA Roughness ( Count/Km )

a dan b = Parameter yang diperoleh dari kalibrasi

terhadap kendaraan tertentu yang dipakai.

di Indonesia terdapat 8 buah kendaraan yang dioperasikan diseluruh wilayah Indonesia dan telah dikalibrasikan sebagai berikut :

Jenis Kendaraan	Plat Nomor	Parameter	
		a	b
Holden	B 8909 FK	- 61,9	84,4
Holden	B 8592 SA	- 43,7	118,5
Ford	B 7355 AK	- 84,7	167,3
LUV	B 8730 VS	-61,1	117,4
LUV	B 8731 VS	- 31,4	109,9
LUV	B 8491 VS	- 32,1	116,0
LUV	B 8407 VS	- 89,0	124,9
LUV	B 8775 VS	- 91,5	115,9

Untuk menentukan kondisi jalan yang disurvei dipakai angka indek kekasaran yang

<sup>1</sup> J.B. Cox, T. Gentles; Measurement of Road Condition and The Introduction of NAASRA Roughness Meter Indonesia, PUSLITBANG JALAN Bandung, 1983 hal. 4.2

dinamakan *Road Condition Index* (RCI) yaitu skala dari tingkat kenyamanan atau kinerja dari jalan yang diperoleh dari hasil pengukuran dengan alat NAASRA Roughness Meter. Angka indek tersebut mempunyai nilai antara 0 sampai 10 dengan penilaian sebagai berikut:

- 8 - 10 Sangat rata dan teratur.
- 7 - 8 Sangat baik, umumnya rata.
- 6 - 7 Baik.
- 5 - 6 Cukup, sedikit sekali atau tidak ada lubang, tetapi permukaan jalan tidak rata.
- 4 - 5 Jelek, kadang-kadang ada lubang, permukaan jalan tidak rata.
- 3 - 4 Rusak, bergelombang, banyak lubang.
- 2 - 3 Rusak berat, banyak lubang dan seluruh daerah perkerasan hancur.
- $\geq 2$  Tidak dapat dilalui, kecuali dengan 4 WD Jeep.

Karena penetapan kondisi di tiap negara berbeda maka di Indonesia telah ditetapkan dengan standart Bina Marga dipergunakan korelasi antara RCI dan IRI dengan rumus korelasi<sup>2</sup> :

$$RCI = 10 \text{ Exp } ( -0,0501 \times IRI^{1,22092} )$$

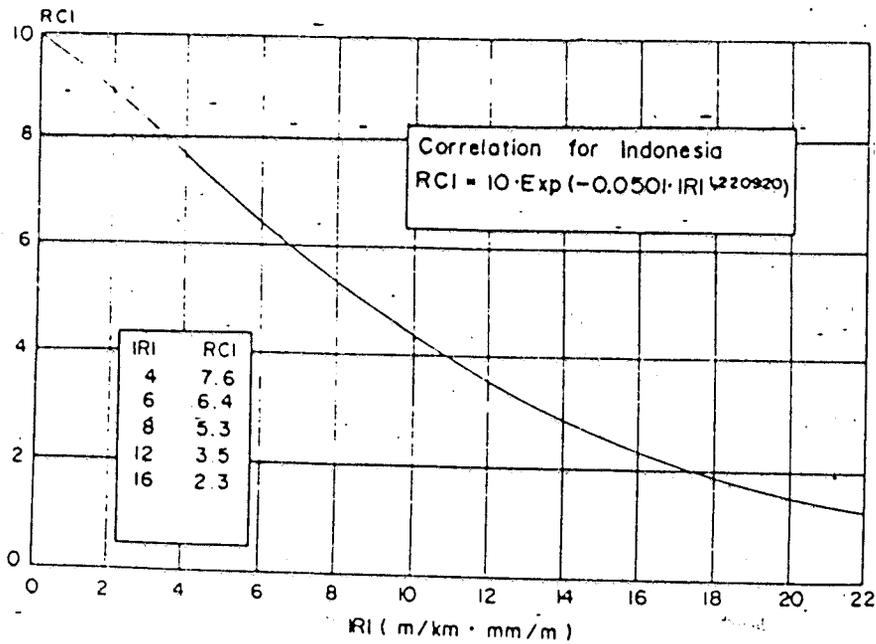
dimana,

RCI = Road Condition Index

IRI = International Roughness Index ( m/Km )

Grafik korelasi antara RCI dan IRI dapat dilihat pada gambar 2.2.

<sup>2</sup> Manual Pemeliharaan Jalan No.03/MN/B/1983, Direktorat Jendral Bina Marga.



Gambar 2.2  
 Grafik Korelasi antara RCI dan IRI di Indonesia

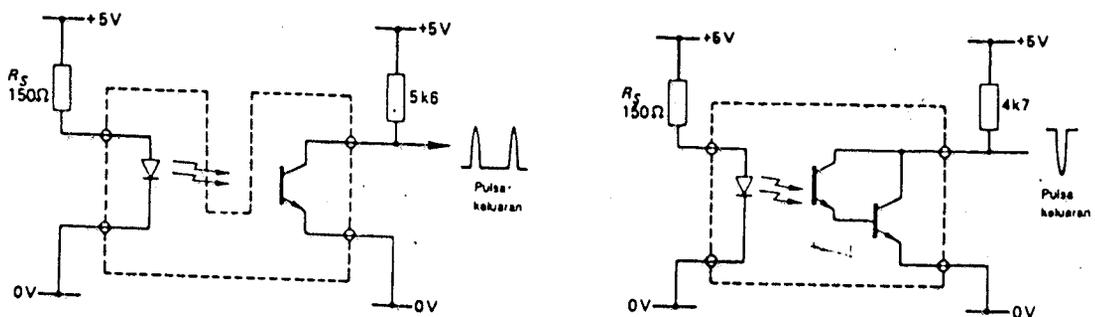
Dari data pengukuran (NR) setelah dikonversi ke dalam standart kekasaran (SR), kemudian dikorelasikan dengan rumus dan tabel di atas maka dapat ditentukan kondisi dari permukaan jalan tersebut.

### 2.3 SENSOR INFRA MERAH

Cahaya infra merah dihasilkan oleh komponen LED infra merah yang berfungsi sebagai komponen pemancar sinyal dalam bentuk cahaya. Sebagai piranti penerimanya adalah fototransistor yang mempunyai karakteristik mengalirkan arus listrik bila pada badannya dikenai cahaya infra merah tersebut. Dipasaran sensor ini telah ada dan dikemas menjadi satu.

Kedua komponen ini secara optik berhubungan tetapi secara elektrik terpisah. Sensor ini memberikan indikasi mati/hidup dan karena itu sangat cocok untuk mendeteksi batas, penginderaan posisi, pencacah batch dan indikasi taraf. Karena terpisah secara elektrik, piranti ini juga cocok untuk isolator listrik atau proteksi listrik pada tegangan tinggi.

Sambungan optik bisa membentangi suatu celah yang berkasnya dapat disela guna memberikan isyarat pulsa dari fototransistor, atau dengan permukaan pantul diantara LED dan fototransistor. Waktu respon dari fototransistor ini sangat cepat sampai 1 ms. Bentuk sensor ini dapat dilihat pada gambar 2.3



Gambar 2.3  
Sensor Infra Merah

## 2.4 PENCACAH (COUNTERS)

Mencacah berarti 'menghitung'. Mencacah jumlah pulsa, mencacah waktu yang lewat, mencacah frekuensi, dan sebagainya. Piranti guna mencacah dinamakan pencacah (*counters*).

Hampir semua sistem logika menerapkan pencacah. Ada 2 jenis pencacah, yaitu :

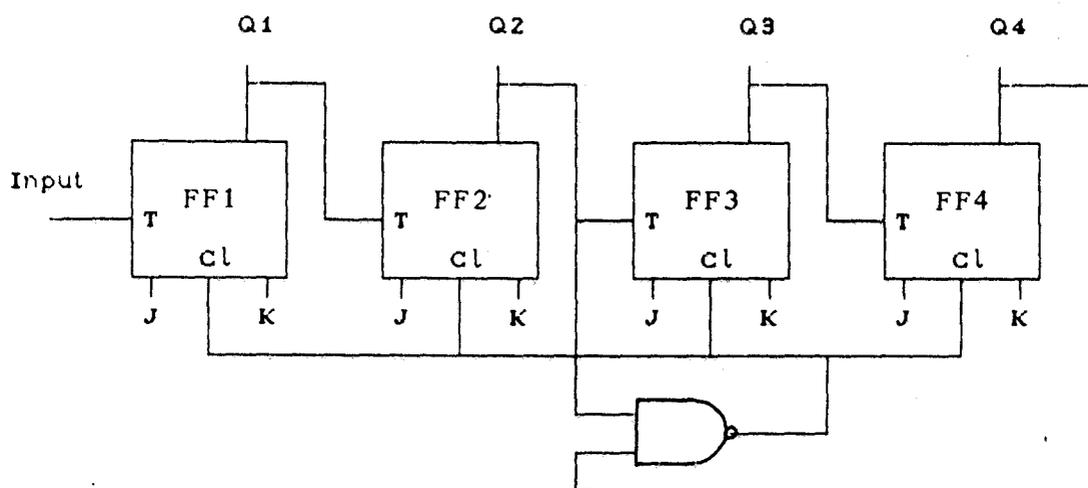
1. Pencacah sinkron (*synchronous counters*) atau pencacah jajar.
2. Pencacah tak-sinkron (*asynchronous counters*) atau pencacah deret (*series counters*).

Selanjutnya karakteristik penting dari pencacah adalah :

1. jumlah cacahan maksimum yang dapat dicapai.
2. mencacah maju, atau mencacah mundur.
3. pencacah sinkron atau tak-sinkron.
4. Dapat berjalan sendiri (*free running*) atau dapat berhenti sendiri (*self-stopping*).

#### 2.4.1 Pencacah-10 BCD Tak-sinkron (Decade Counter)

Adalah rangkaian pencacah yang mempunyai 10 keadaan dan karena itu membagi frekuensi masukannya dengan 10. Pencacah ini merupakan dasar bagi mayoritas sistem peraga. Setiap pencacah dekadanya pertama disimpan dalam lach 4 bit dan kemudian didekoder guna menggerakkan piranti peraga. Pencacah ini dapat dibentuk dari pencacah 16 dengan menambah gerbang NAND yang inputnya diambilkan dari output flipflop 2 dan flipflop 4 selanjutnya output dari NAND dihubungkan ke Cl (*Clear*) guna mereset semua flipflop untuk kembali ke biner 0000 pada saat terjadi perubahan biner dari 1001 (9) ke 1010 (10). Rangkaian pencacah 10 ini dapat dilihat pada gambar 2.4.



Gambar 2.4

Rangkaian Pencacah-10 BCD Tak-sinkron

## 2.5 PENGGERAK PERAGA 7-SEGMENT

Untuk menggerakkan display 7-segment dipakai IC decoder yang merubah kode biner 4-bit (BCD) ke kode 7-segment a,b,c,d,e,f,dan g. Type IC yang sesuai dengan ini dan secara umum banyak dipakai diantaranya 74LS47. Logika dari IC ini dapat dilihat pada gambar 2.5.

Desimal atau Fungsi	masukan				keluaran							Note			
	$\overline{\text{LT}}$	$\overline{\text{RBI}}$	D	C	B	A	$\overline{\text{BI/RBO}}$	$\overline{\text{a}}$	$\overline{\text{b}}$	$\overline{\text{c}}$	$\overline{\text{d}}$		$\overline{\text{e}}$	$\overline{\text{f}}$	$\overline{\text{g}}$
0	1	1	0	0	0	0	1	0	0	0	0	0	0	1	1
1	1	X	0	0	0	1	1	1	0	0	1	1	1	1	1
2	1	X	0	0	1	0	1	0	0	1	0	0	1	0	1
3	1	X	0	0	1	1	1	0	0	0	0	1	1	0	1
4	1	X	0	1	0	0	1	1	0	0	1	1	0	0	1
5	1	X	0	1	0	1	1	0	1	0	0	1	0	0	1
6	1	X	0	1	1	0	1	1	1	0	0	0	0	0	1
7	1	X	0	1	1	1	1	0	0	0	1	1	1	1	1
8	1	X	1	0	0	0	1	0	0	0	0	0	0	0	1
9	1	X	1	0	0	1	1	0	0	0	1	1	0	0	1
10	1	X	1	0	1	0	1	1	1	1	0	0	1	0	1
11	1	X	1	0	1	1	1	1	1	0	0	1	1	0	1
12	1	X	1	1	0	0	1	1	0	1	1	1	0	0	1
13	1	X	1	1	0	1	1	0	1	1	0	1	0	0	1
14	1	X	1	1	1	0	1	1	1	1	0	0	0	0	1
15	1	X	1	1	1	1	1	1	1	1	1	1	1	1	1
$\overline{\text{BI}}$	X	X	X	X	X	X	0	1	1	1	1	1	1	1	2
$\overline{\text{RBI}}$	1	0	0	0	0	0	0	1	1	1	1	1	1	1	3
$\overline{\text{LT}}$	0	X	X	X	X	X	1	0	0	0	0	0	0	0	4

Gambar 2.5

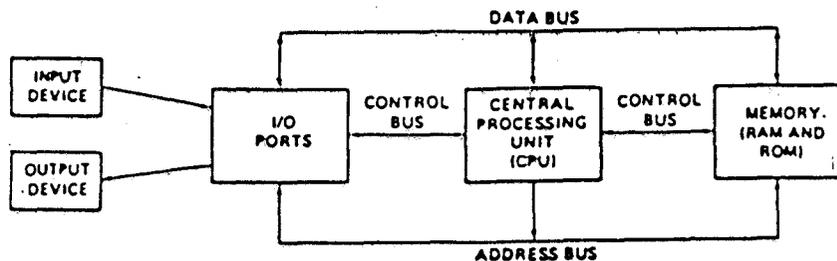
Logika IC 74LS47

Keterangan :

1. BI /RBO sebagai input pemadam (*blanking input*) dan/atau sebagai ouput pemadam kerut (*riple blanking output*). Jika dikehendaki fungsi 0 sampai dengan 15 maka BI = '1', Jika tidak tidak dikehendaki pemadaman desimal 0 maka RBI = '1'. x = *don't care*.
2. Jika input BI = '0' maka ouput semua segment = '0'
3. Jika input RBI, A, B, C, dan D = '0' sedangkan Lamp Test = '1', maka semua ouput segment = '1' dan output RBO = '0'.

## 2.6 MINIMUM SISTEM 8088<sup>3</sup>

Gambar 2.6 menunjukkan blok diagram sistem minimum 8088 yang dipakai.



Gambar 2.6 Blok Diagram Sistem Minimum

Dapat dilihat dari gambar 2.6 tersebut diatas, bahwa dalam sebuah minimum sistem terdiri dari beberapa bagian, yaitu memory, input/output, central processing unit, address bus, data bus dan control bus, dimana masing-masing bagian memiliki tugas tertentu dalam sebuah sistem minimum. Bagian-bagian tersebut dapat dijelaskan sebagai berikut :

### 2.6.1 MIKROPROSESOR 8088

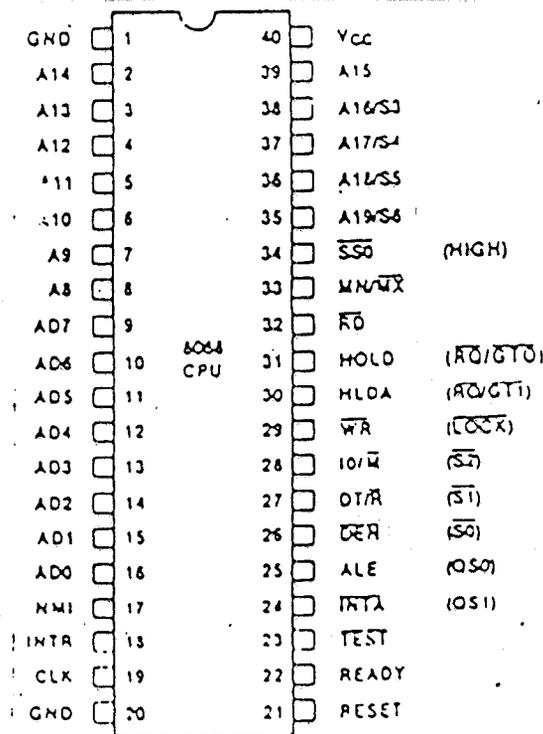
Mikroprosesor 8088 adalah produksi dari Intel Corporation, merupakan hasil perubahan dari mikroprosesor 8086. Perbedaan pokok dari keduanya adalah mengenai jumlah bit data. Mikroprosesor 8086 menggunakan 16 bit data internal maupun eksternal, sedangkan mikroprosesor 8088 menggunakan 16 bit data internal dan 8 bit data eksternal. semua instruksi yang berlaku pada mikroprosesor 8086 berlaku pada mikroprosesor 8088.

Mikroprosesor 8088 mempunyai jalur address bus sebanyak 20 bit yaitu A0 - A19, sehingga mampu melakukan pengalamatan sampai 1.048.576 lokasi atau 1 Megabyte.

#### 2.6.1.1 Konfigurasi Pin 8088

<sup>3</sup> Douglas V. Hall, *Microprocessor and Interfacing*, McGraw Hill, 1986, hal 24

Konfigurasi pin mikroprosesor 8088 dapat dilihat secara lengkap pada gambar 2.7.



Gambar 2.7<sup>4</sup>

#### Konfigurasi Pin 8088

##### - ADO - AD7 (*Address Data Bus*)

Merupakan jalur address dan data yang bekerja secara *multiplex*. Jalur ini akan bekerja sebagai address (A0 - A7) pada *bus cycle* yang pertama (T1), dan pada *bus cycle* berikutnya (T2, T3, dan T4) jalur ini akan berfungsi sebagai jalur data. Pada gambar terlihat timing diagram dari *bus cycle* dari sistem. Disini dapat dilihat untuk jalur ADDRESS/DATA dapat berfungsi baik sebagai *address* maupun data yang bergantung dari *bus cycle*.

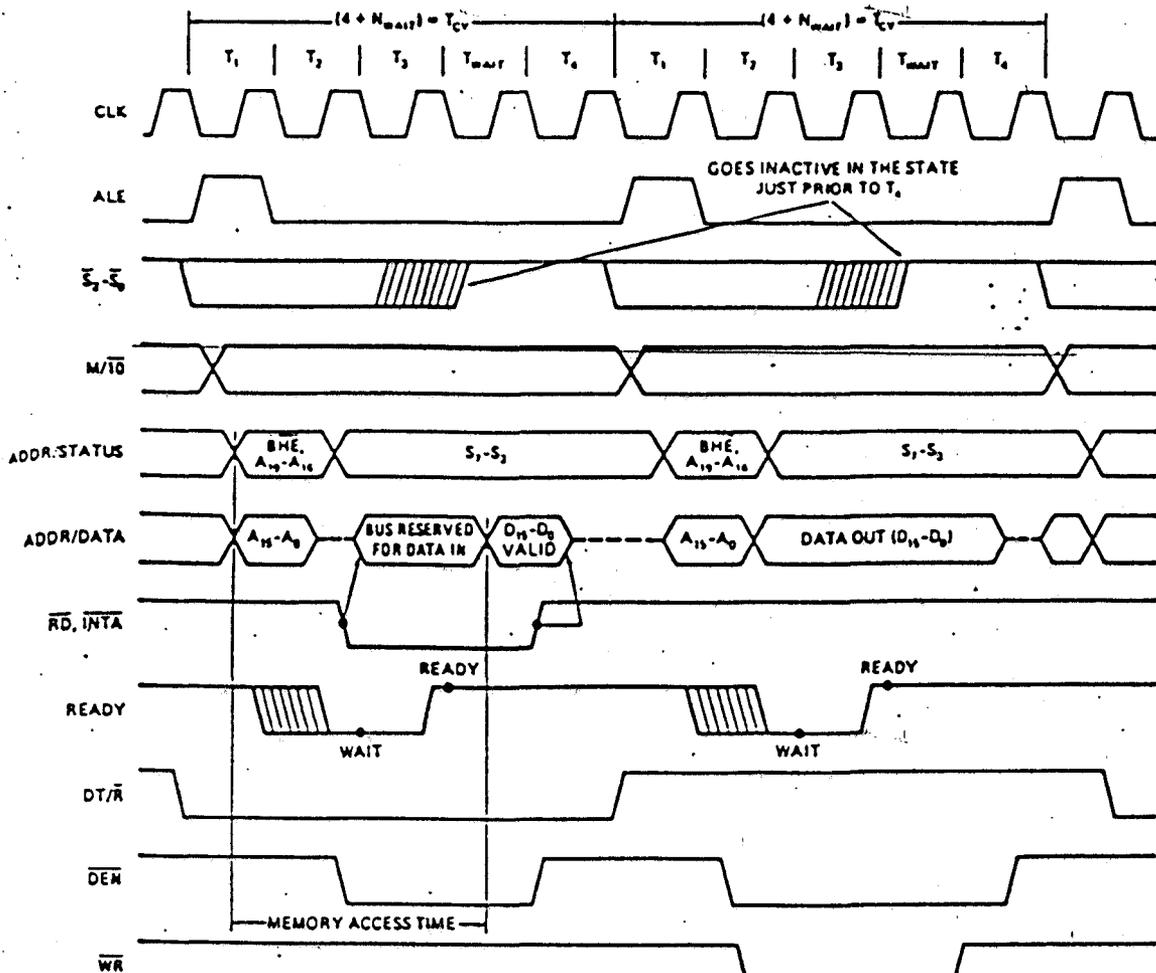
##### - A8 - A15 (*Address Bus*)

Merupakan address bus untuk seluruh bus cycle (T1 - T4). Bus ini tak memerlukan *latch* oleh ALE agar valid.

<sup>4</sup> Intel, Microsystem Component Handbook - Microprocessor Volume 1 Intel Co., 1986, hal 3-106

- A16/S3 - A19/S6 (Address/Status)

Jalur ini berfungsi sebagai jalur address dan status yang bekerja secara *multiplex*. Selama *bus cycle* yang pertama (T1), jalur ini berfungsi sebagai address (A16 - A19) untuk operasi memori, selama operasi I/O jalur ini akan low. Untuk bus cycle berikutnya jalur ini berfungsi sebagai penunjuk dari 8088. S6 selalu dalam kondisi low.



Gambar 2.8<sup>5</sup>

Timing Diagram Dari 8088

Pada setiap permulaan bus cycle S5 akan menunjukkan status *dari interrupt enable flag*. S4

<sup>5</sup> Ibid., hal 3-115

dan S3 merupakan petunjuk dari segmen register yang digunakan untuk mengakses data.

- RD (Read)

Sinyal read menyatakan bahwa prosesor 8088 melaksanakan operasi *memory read* atau I/O read, yang tergantung pada status yang terdapat pada pin IO/M. Jika RD *low* dan IO/M *high*, maka yang dilakukan adalah operasi *read I/O*. Sedangkan jika RD *low* dan IO/M *low*, maka yang dilakukan adalah operasi *memory read*.

- READY

Adalah suatu sinyal pemberitahuan yang menyatakan bahwa transfer data pada peralatan I/O atau memori telah selesai.

- INTR (*Interrupt Request*)

Merupakan sinyal *maskable interrupt* yang digunakan untuk meng-*interrupt* mikroprosesor 8088. Sinyal tersebut aktif *high*. Pada proses pelayanan *interrupt*, harus ada rutin *interrupt* yang dilaksanakan dimana address awal rutin ini ditunjukkan *vector interrupt* yang terdapat dalam memori.

- TEST

Merupakan sinyal input yang akan dipergunakan oleh instruksi *wait for test*. Jika sinyal ini *low*, maka operasi dilanjutkan dan jika *high* maka prosesor tidak akan melakukan apa-apa.

- NMI (*Non Maskable Interrupt*)

Merupakan jenis *interrupt* yang berasal dari hardware, dan *interrupt* ini tidak bisa dihalangi oleh software.

- RESET

Dengan adanya sinyal RESET ini, maka mikroprosesor akan menghentikan semua operasi yang sedang berlangsung, dan akan membuat instruksi dijalankan mulai dari awal lagi.

- CLK (*Clock*)

CLK merupakan input clock dari mikroprosesor dan *Bus Controller* dan berfungsi untuk mengatur timing dari operasi mikroprosesor dan *Bus Controller*.

- MN/MX (*Minimum/Maximum*)

Suatu logika yang diberikan kepada pin ini menyatakan mode operasi dari mikroprosesor 8088. Jika input pin ini *low*, maka 8088 beroperasi dengan mode maksimum, sebaliknya jika input pin ini *high* maka 8088 beroperasi dengan mode minimum.

- VCC

Merupakan pin power supply. Besarnya input yang diberikan pada pin ini adalah +5 volt dengan toleransi 10%.

- GND (Ground)

Merupakan pin ground dari prosesor.

Berikut ini adalah penjelasan pin-pin 8088 bila dioperasikan dalam mode minimum.

- IO/M (*Status Line*)

Berfungsi untuk membedakan *memory access* atau *I/O access*. Bila kondisinya *low* menyatakan *I/O access* dan bila kondisinya *high* menyatakan *memory access*.

- WR (*Write*)

Sinyal ini menyatakan bahwa 8088 sedang melaksanakan operasi *I/O write* atau *memory write* yang tergantung dari pin IO/M. WR aktif pada cycle T2 dan T3 dan Tw dari *Bus Cycle*.

- INTA (*Interrupt Acknowledge*)

Menyatakan bahwa 8088 telah mendeteksi adanya permintaan *interrupt*.

- ALE (*Address Lact Enable*)

Merupakan sinyal output dari 8088 yang digunakan untuk menyimpan address ke dalam address lath.

- DT/R (*Data Transmit/Receive*)

Diperlukan untuk mengatur arah pengiriman data pada *Bus Transceiver*.

- HOLD, HLDA

Hold digunakan apabila sebuah prosesor lain ingin mempergunakan bus, dengan memerintahkan 8088 untuk melepaskan sistem bus, sehingga bus dapat dipakai oleh prosesor yang lain.

- DEN (*Data Enable*)

Sinyal ini berfungsi agar data pada transceiver dapat dienablekan. DEN ini aktif low untuk masing-masing I/O dan memory access, dan pada INTA cycle.

Tabel 2.1<sup>6</sup> Operasi Bus Cycle Mode Minimum

IO/M	DT/R	SSO	Operasi
1	0	0	Interrupt Acknowledge
1	0	1	Read I/O Port
1	1	0	Write I/O Port
1	1	1	Halt
0	0	0	Code Access
0	0	1	Read Memory
0	1	0	Write Memory
0	1	1	Passive

- SSO (*Status Line*)

<sup>6</sup> Ibid. hal 125

Merupakan status line yang bersama-sama dengan IO/M dan DT/R membentuk suatu sistem decoding dari status Bus Cycle. Pada Tabel 2.1 dapat dilihat fungsi yang dibentuk oleh ketiga sinyal tersebut.

### 2.6.1.2 Register Mikroprosesor 8088

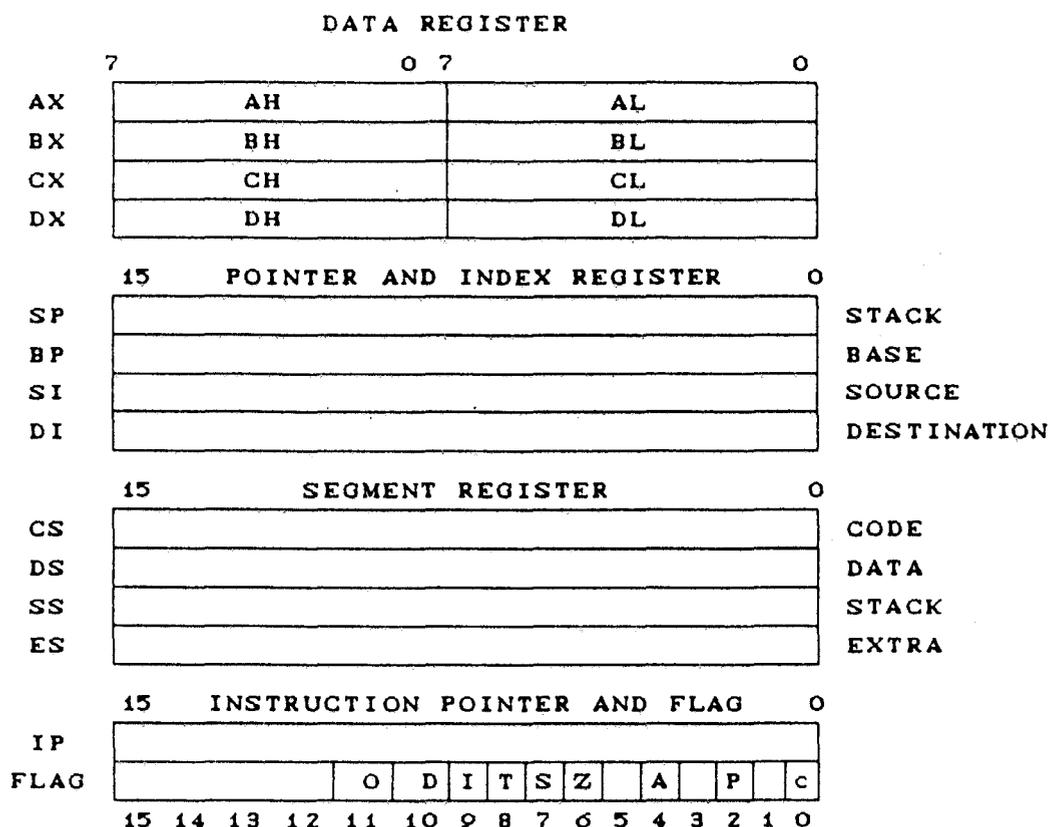
Mikroprosesor 8088 mempunyai 14 register 16 bit. Register-register ini dikelompokkan atas 4 buah grup, yaitu:

- Data grup register
- Pointer dan Index grup register
- Segment grup register
- Kelompok register pelengkap

Pengelompokan dari register 8088 ini dapat dilihat pada gambar 2.9.

Data grup terdiri atas 4 buah register yaitu AX, BX, CX, dan DX. Register-register 16 bit ini dapat dioperasikan baik untuk operasi 8 bit. Jika register-register dioperasikan untuk operasi 8 bit, maka tiap-tiap register akan terbagi menjadi 2 register yaitu: AH, AL, BH, CH, CL, dan DL.

*Pointer dan Index Grup* terdiri atas 4 buah register 16 bit yaitu: SP (*stack pointer*), BP (*base pointer*), SI (*source index*), DI (*destination index*). Register-register ini hanya dapat dioperasikan untuk 16 bit berfungsi untuk mencatat address stack dan indek yang isinya akan digarap.

Gambar 2.9<sup>7</sup>

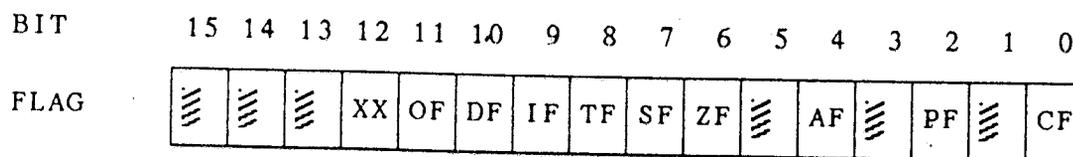
## Register Mikroprosesor 8088

Setiap segment register adalah register 16 bit khusus untuk mencatat address awal segment memori dimana tiap segment mencakup maksimum 64 Kbyte lokasi. Akan tetapi mikroprosesor 8088 dapat menggarap isi memori hingga 1 Mbyte dan secara internal tetap bekerja dengan data 16 bit, yaitu dengan menambahkan 16 bit offset address dengan 16 bit segment address yang digeser kekiri sebanyak 4 kali atau dengan kata lain segment addressnya dikalikan dengan 16 dan baru ditambahkan dengan offset addressnya. Segment

<sup>7</sup> Lewis C. Eggebrechth, *Interfacing To IBM PC*, Howard W. Sam & Co, Inc., 1985, hal 35

register ini terdiri dari *Code Segment (CS)*, *Data Segment (DS)*, *Stack Segment (SS)* dan *Extra Segment (ES)*.

Kelompok register pelengkap terdiri dari Flag register dan Instruction Pointer. Flag register adalah register 16 bit, dimana 9 bit diantaranya, aktif digunakan untuk menampung tanda yang berkaitan dengan operasi khusus tentang kerja mikroprosesor terdiri dari *overflow flag (OF)*, *direction flag (DF)*, *interrupt flag (IF)*, *trap flag (TF)* dan tanda akibat operasi aritmatika dsan logik yang terdiri dari bit-bit *Sign Flag (SF)*, *Zero Flag (ZF)*, *Auxiliary Carry Flag (AF)*, *Parity Flag (PF)*, dan *Carry Flag (CF)*. Posisi tiap bit pada register flag ini dapat digambarkan sebagai berikut :



Bit bertanda /// berarti tidak dipergunakan sama sekali

Gambar 2.10

Posisi tiap bit pada Flag Register

## 2.6.2 MEMORI

Fungsi utama memori pada sistem komputer adalah sebagai tempat yang dipergunakan untuk menyimpan data program. Data yang disimpan dapat bersifat sementara atau permanen. Data yang disimpan secara permanen biasanya adalah program instruksi bagi mikroprosesor. Data-data yang disimpan dalam memori untuk sementara adalah hasil-hasil perhitungan dan

data yang dipergunakan pada proses manipulasi.

Berdasarkan sifat dan kemampuannya dalam menyimpan data, maka komponen penyimpan data ini dapat dibedakan menjadi dua jenis. Yang pertama adalah memori yang menyimpan data secara permanen, isinya hanya bisa dibaca dan tidak bisa dihapus. Memori jenis ini disebut ROM (*Read Only Memory*). Yang kedua adalah memori untuk menyimpan data bersifat sementara, isinya bisa dibaca dan dihapus atau diganti. Jenis memori yang kedua disebut RAM (*Random Access Memory*).

#### 2.6.2.1 ROM (READ ONLY MEMORY)

Pada suatu mikroprosesor diperlukan suatu memori yang menyimpan data untuk selamanya (*non volatile*) dimana data yang terdapat pada memori tersebut tidak akan hilang bila power dari sistem dimatikan. Informasi yang terdapat pada ROM hanya dapat dibaca tidak dapat diubah sehingga ROM ini sangat berguna dalam suatu sistem karena memungkinkan mikroprosesor untuk menginisialisasikan semua peripheral hardware pada keadaan logika yang sesuai pada saat pertama kali sistem dinyalakan.

Ada beberapa jenis memori yang bersifat *non volatile* yang dapat dipakai dalam suatu sistem mikroprosesor yaitu ROM, PROM (*Programamable Read Only Memory*), EPROM, (*Eraser Programable Read Only Memory*), dan EAROM (*Electrically Alterable Read Only Memory*).

Organisasi internal dari data pada EPROM biasanya diatur sebagai 1024x8, 2048x8 atau 4096x8. Angka yang pertama menunjukkan jumlah dari lokasi address yang terdapat pada suatu IC sedangkan angka yang kedua dari EPROM pada setiap lokasi address. Misalkan IC

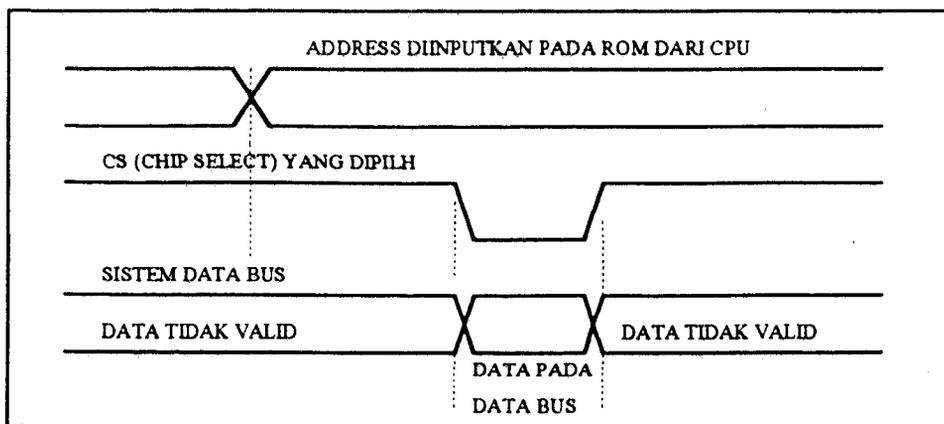
EPROM 27128 mempunyai 16384 lokasi dan 8 bit data paralel yang dapat dibaca. Oleh karena mempunyai 16384 lokasi, maka 27128 disebut EPROM 16 Kbyte.

### Membaca Data Dari ROM

Proses pembacaan data dari ROM mempunyai urutan tertentu sebagai berikut:

1. ROM memerlukan input address yang sesuai dengan lokasi yang akan dibaca.
2. Kemudian mikroprosesor menunggu untuk selang waktu tertentu yang disebut *Read Access Time* yang berkisar antara 100 - 300 ns tergantung dari jenis ROM yang dipakai. Selang waktu ini digunakan untuk menunggu stabilnya di dalam ROM dan tercapainya lokasi yang dituju.
3. Sinyal Chip Select menjadi aktif sehingga data yang diinginkan sekarang berada pada data bus. Data tersebut langsung dibaca oleh mikroprosesor.
4. Sinyal Chip Select dinonaktifkan agar data dari ROM lenyap dari data bus.

Untuk lebih jelasnya, dapat dilihat pada gambar 2.11.



Gambar 2.11<sup>8</sup> Timing Diagram Pembacaan Data Dari ROM

### 2.6.2.2 RAM (RANDOM ACCESS MEMORY)

<sup>8</sup> Coffron, James W., Z80 Application., Sybex., Inc., 1983, hal 6.

Berlainan dengan ROM, maka RAM termasuk jenis memori yang *volatile*, artinya data yang tersimpan pada RAM akan hilang pada saat power dimatikan. Oleh karena itu RAM hanya digunakan untuk menyimpan data ataupun variabel yang sifatnya sementara. Dinamakan *Random Access Memory* karena lokasi manapun dapat dicapai secara langsung dengan menempatkan adress yang tepat pada inputnya.

Pada RAM terdapat parameter *Read Access Time* yaitu waktu yang diperlukan oleh data untuk menjadi stabil pada output pin dari memori setelah menerima permintaan dari mikroprosesor. *Read Access Time* ini merupakan parameter penting karena menentukan kecepatan dari RAM.

Parameter lain yang tidak kalah pentingnya adalah *Write Access Time* yaitu waktu yang diperlukan sesudah Address Bus yang diberikan pada input stabil sesudah Address Clock diberikan. Ada dua jenis RAM yang terdapat dewasa ini yaitu Static RAM dan Dynamic RAM.

Bila suatu informasi disimpan dalam suatu memori semikonduktor maka data tersebut ditulis ke memori. Sedangkan bila informasi diambil dari memori dikatakan informasi tersebut dibaca dari memori. Hanya dua fungsi inilah yang dapat dikerjakan pada static memory.

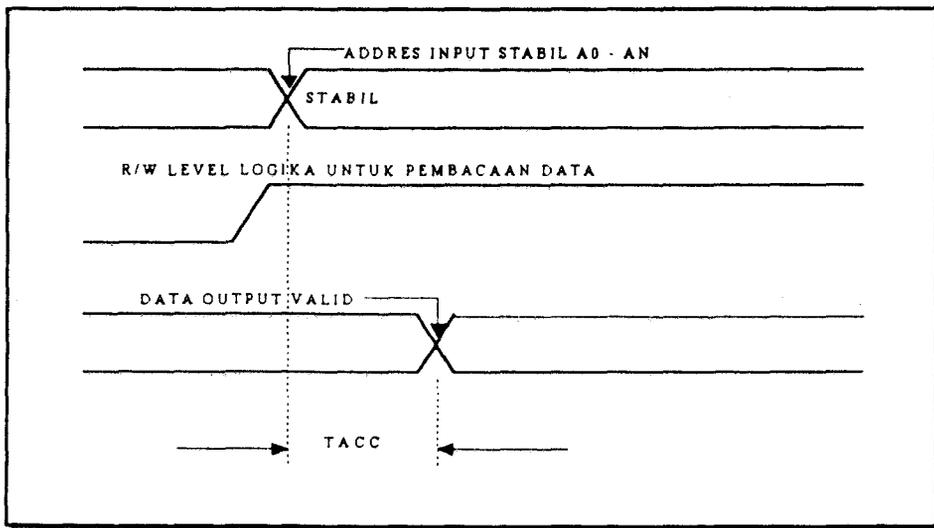
Static RAM dapat menyimpan data selama DC Power tidak diputuskan sedangkan Dynamic RAM harus diberi *Refresh Cycle* agar informasi dapat dipertahankan.

Menulis pada memori dikerjakan dalam suatu write cycle. Membaca informasi dari memori dikerjakan dalam suatu read cycle. Istilah *cycle* disini didefinisikan sebagai suatu perioda yang tetap dari waktu yang diperlukan untuk fungsi menulis dan membaca memori. Informasi yang ditulis pada memori berupa data elektris dalam suatu DC level tertentu yang dapat berupa logika high (1) atau low (0) untuk menulis atau membaca memori diperlukan suatu hubungan input dan output antara mikroprosesor dengan memori.

## Membaca Data Dari RAM

Pada saat sistem akan membaca data dari RAM microprocessor data lines harus dalam keadaan input. Untuk membaca data mikroprosesor melakukan beberapa urutan langkah-langkah sebagai berikut :

1. Pertama-tama memori menerima address yang menentukan lokasi tertentu. Rangkaian dekoder di dalam RAM tersebut memilih elemen manakah yang harus diaktifkan.
2. Memori read sinyal menjadi aktif dan memori langsung menerima sinyal ini.



Gambar 2.12<sup>9</sup>

Timing Diagram Proses Baca Data dari RAM

3. Kemudian sistem menunggu selama selang waktu tertentu, yang disebut *Read Access Time*, untuk menunggu rangkaian dari RAM menjadi stabil.

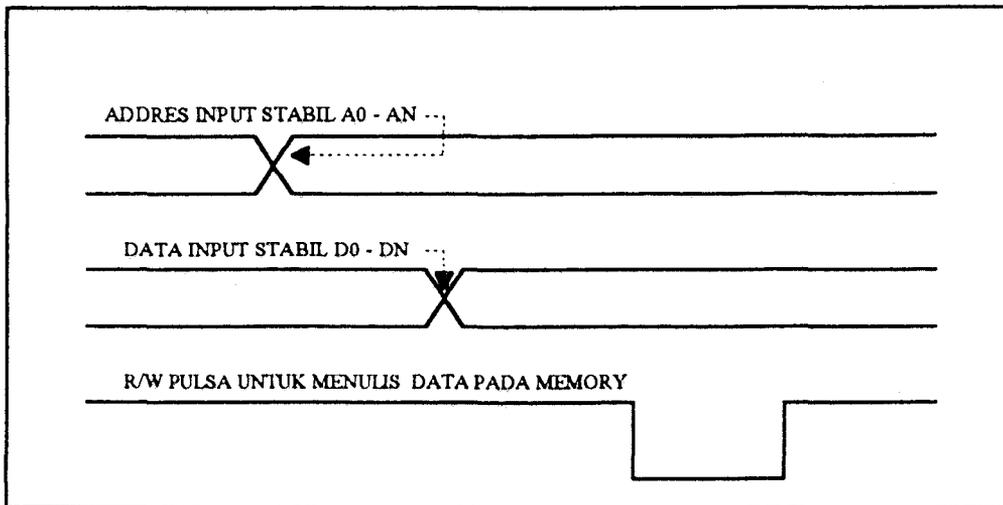
<sup>9</sup> Ibid., hal 30

4. Setelah itu RAM akan mengeluarkan data yang dikehendaki pada data bus dimana data tersebut akan segera dibaca oleh mikroprosesor.

### Menulis Data Ke RAM

Urutan penulisan data ke RAM adalah sebagai berikut :

1. Memori menerima address yang sesuai dengan lokasi yang akan ditulis.
2. Kemudian data yang akan ditulis pada memori ditempatkan pada Data Bus.



Gambar 2.13<sup>10</sup>

Timing Diagram Penulisan Data ke RAM

## 2.7 DECODER

Rangkaian decoder digunakan agar pada suatu saat hanya ada satu device yang bekerja, agar tidak terjadi konflik data. Yang dimaksud device disini dapat berupa RAM, ROM atau

<sup>10</sup> Ibid., hal 31

*Peripheral Interface*. Rangkaian Decoder untuk RAM dan ROM berdasarkan I/O mapping.

Ada dua teknik yang biasa digunakan dalam decoding, yaitu teknik *partial decoding* dan *full decoding*. Pada teknik *full decoding*, seluruh zone bit terhubung ke rangkaian decoder. Dengan teknik ini suatu device hanya mempunyai satu lokasi memory atau dengan kata lain tidak ada duplikasi lokasi sehingga dapat menghemat lokasi memory. Pada teknik *partial decoding* tidak seluruh zone bit terhubung ke decoder yang mengakibatkan terdapat duplikasi lokasi untuk satu device.

## 2.8 PROGRAMABLE PERIPHERAL INTERFACE (PPI) 8255

PPI 8255 dirancang sebagai IC yang menangani interface dalam sistem mikroprosesor yang dikemas dalam bentuk 40 pin *Dual In-line Package* (DIP) dan termasuk jenis *Large Scale Integration* (LSI). Gambar 2.14 menunjukkan diagram blok dari PPI 8255.

Pada gambar tersebut terdapat 2 kelompok besar yang disebut kelompok kendali A dan kelompok kendali B. Kedua kelompok kendali tersebut mengendalikan 4 kelompok I/O yang disebut :

- Port A (PA0 - PA7)
- Port B (PB0 - PB7)
- Port C Lower (PC0 - PC3)
- Port C Upper (PC4 - PC7)

Kelompok A mengontrol fungsi dari port A dan port C Upper, sedangkan kelompok B mengontrol fungsi dari port B dan port C Lower. Semua bagian dalam 8255 tersebut dihubungkan dengan interval data bus, dan melalui interval data bus inilah data dikirim atau diterima oleh setiap port. Port-port tersebut dapat digunakan dalam 3 mode, yaitu :

- Mode 1 : Basic Input/Output
- Mode 2 : Strobed Input/Output
- Mode 3 : Bi-Directional Bus.

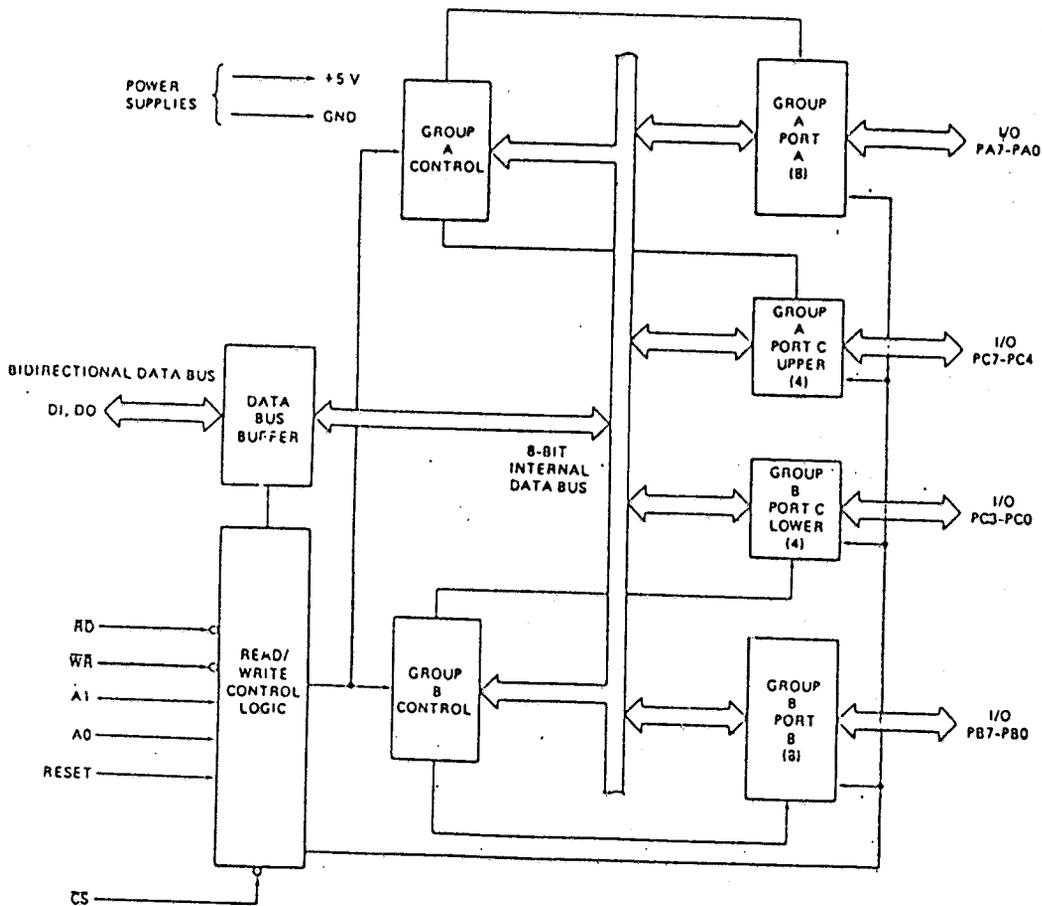
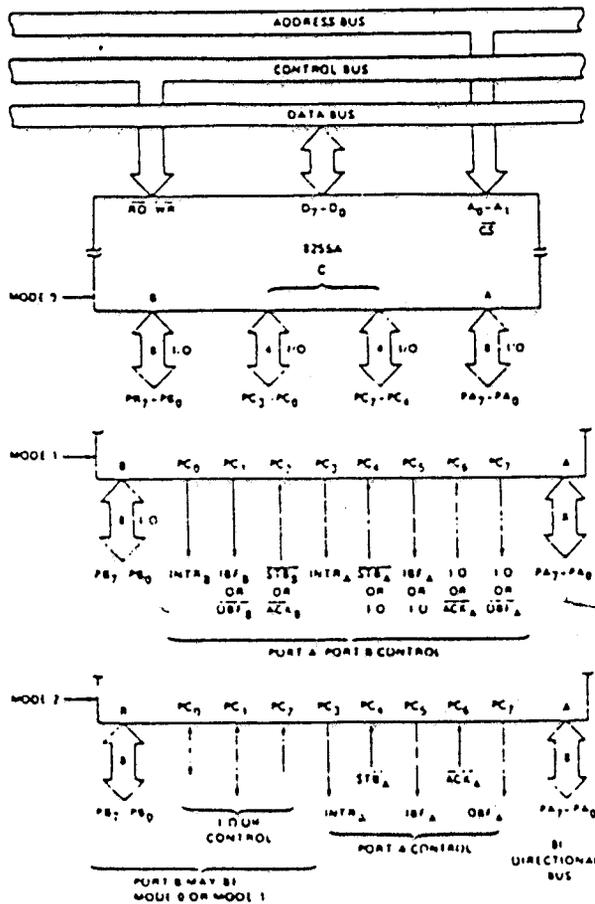
Gambar 2.14<sup>11</sup>

Diagram Blok dari PPI 8255

Penggunaan mode-mode diatas dapat dilihat dari diagram blok pada gambar 2.15. Untuk menggunakan mode yang dipakai maka diperlukan suatu inisialisasi sebelum menggunakan PPI ini, yaitu dengan mengisi register *control word*. Format dari control word dapat dilihat pada

<sup>11</sup> Microprocessor and Peripheral Handbook, Vol. II, INTEL Corporation, 1988

gambar 2.16.



Gambar 2.15

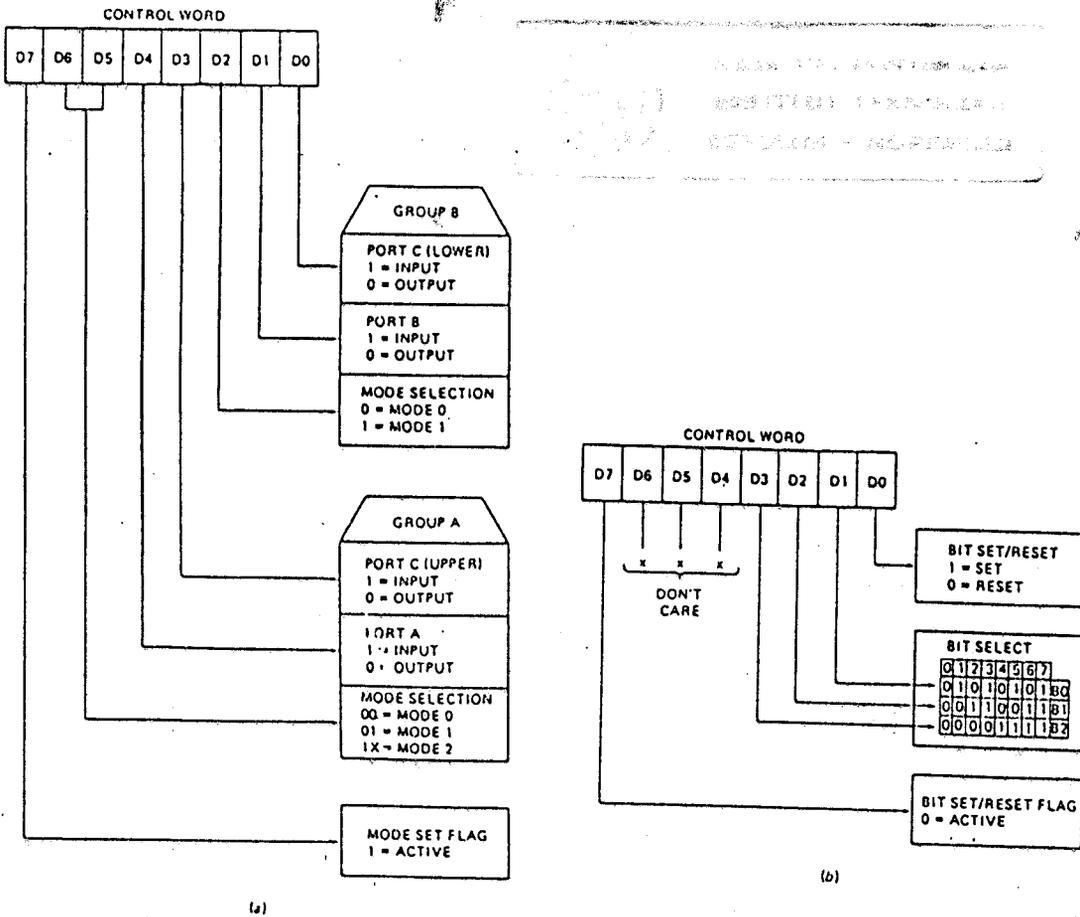
Definisi Dasar Mode dan Interface bus PPI 8255

2.8.1 Fungsi Pin Pada PPI 8255

PPI 8255 berbentuk 40 pin Dual In-line Package (DIP) dengan bentuk fisik seperti pada gambar 2.17. Fungsi dari tiap-tiap pin dapat dijelaskan sebagai berikut :

- Data Bus (D0 - D7) :

Digunakan untuk input/output data dari/ke mikroprosesor, dimana semua informasi diterima dan dikirim melalui 8 bit data bus ini.

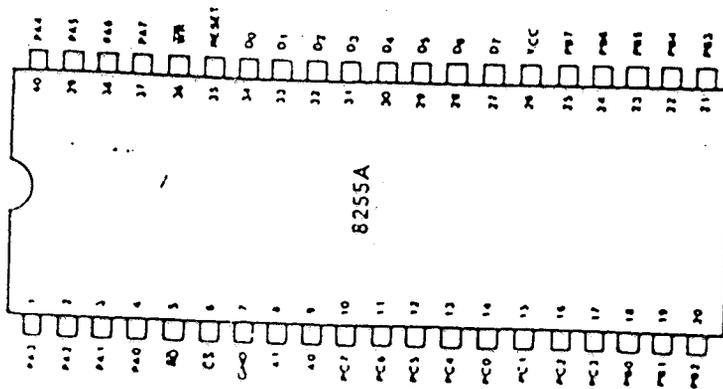


Gambar 2.16<sup>12</sup>  
 Format Control Word dari PPI 8255

- Chip Select (CS) :

Digunakan untuk mengaktifkan chip 8255. Chip 8255 akan aktif bila chip select mendapat sinyal logika low

<sup>12</sup> Ibid. hal 265



Gambar 2.17  
Konfigurasi pin 8255

- Read (RD) :

Digital kontrol yang memungkinkan operasi baca, dimana jika signal berlogika 0, maka mikroprosesor membaca data pada I/O port.

- Write (WR) :

Sinyal kontrol yang memungkinkan operasi penulisan (*write operation*) dari I/O port yang dipilih.

- Address Input (A) - A1) :

Kombinasi dari kedua address input ini menentukan register mana dari 8255 yang menerima atau mengirim data dari atau ke prosesor.

- Reset :

Fungsi dari pin ini adalah untuk mereset 8255, dengan memberikan input high. Pada saat reset ini semua I/O port diset dalam mode input.

- Port A (PA0 - PA7) :

Digunakan sebagai 8 bit input/output port untuk berhubungan dengan peralatan luar.

- Port B (PB0 - Pb7) :

Digunakan sebagai 8 bit input/output port untuk berhubungan dengan peralatan luar.

- Port C (PC0 - PC7)

Port C dibagi menjadi 2 yaitu port C lower (PC0 - PC3) dan port C upper (PC4 - PC7), untuk input/output port yang berhubungan dengan peralatan luar.

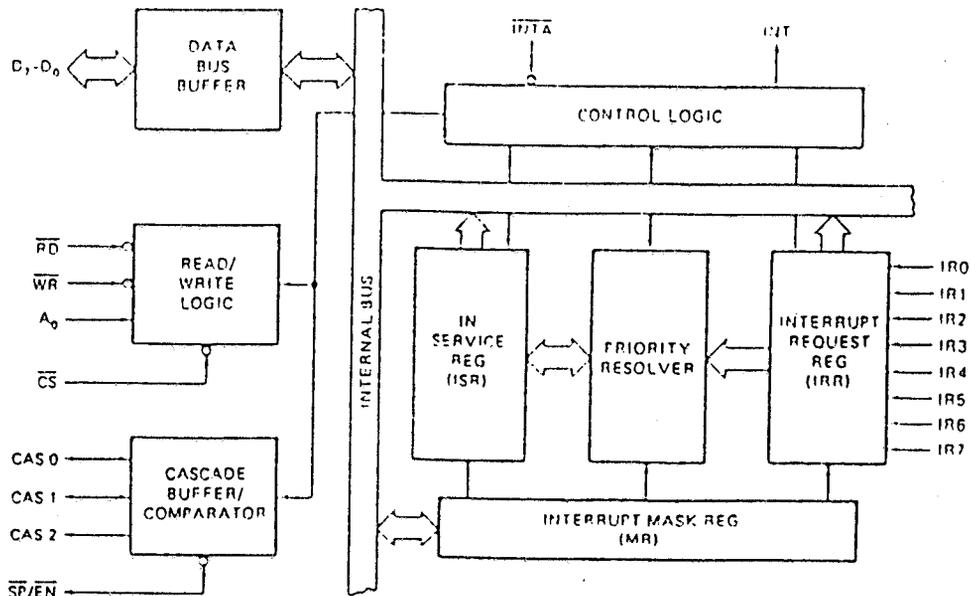
## 2.9 PRIORITY INTERRUPT CONTROLLER (PIC) 8259

*Priority Interrupt Controller* PIC 8259 adalah suatu device yang dihubungkan dengan mikroprosesor dan dikontrol melalui program untuk melakukan interupsi pada pelaksanaan suatu program. Internal bloknya dapat dilihat pada gambar 2.18.

Data bus pada PIC terhubung dengan data bus pada mikroprosesor. Data bus memungkinkan mikroprosesor mengirimkan control word ke 8259 dan membaca status. RD\_ dan WR\_ mengontrol transfer ini saat CS\_ "low". Data bus juga memungkinkan 8259 mengirimkan type interrupt ke mikroprosesor. Pengalamatan PIC ini dipilih dari output decoder yang masuk ke pin CS\_, sebuah PIC memiliki dua buah alamat yang diatur dari pin A0.

PIC 8259 memiliki 8 buah pin interrupt dimana prioritas interruptnya dapat diatur menurut mode yang dipilih. Antara lain ke-delapan interrupt memiliki tingkat prioritas yang sama atau yang lebih kecil tingkat prioritasnya lebih tinggi, dengan kata lain IRQ0 memiliki tingkat prioritas tertinggi sedangkan IRQ7 memiliki tingkat prioritas terendah dan sebagainya.

Ini berarti apabila CPU sedang menjalankan interrupt IRQ2, interrupt yang lebih besar seperti IRQ3 atau lebih tidak dapat menginterrupt tetapi IRQ1 dan IRQ0 dapat menginterrupt interrupt dari IRQ2 tersebut. PIC juga menyediakan cascade lines CAS0, CAS1 dan CAS2.



Gambar 2.18<sup>13</sup>

Internal Blok Diagram PIC 8259

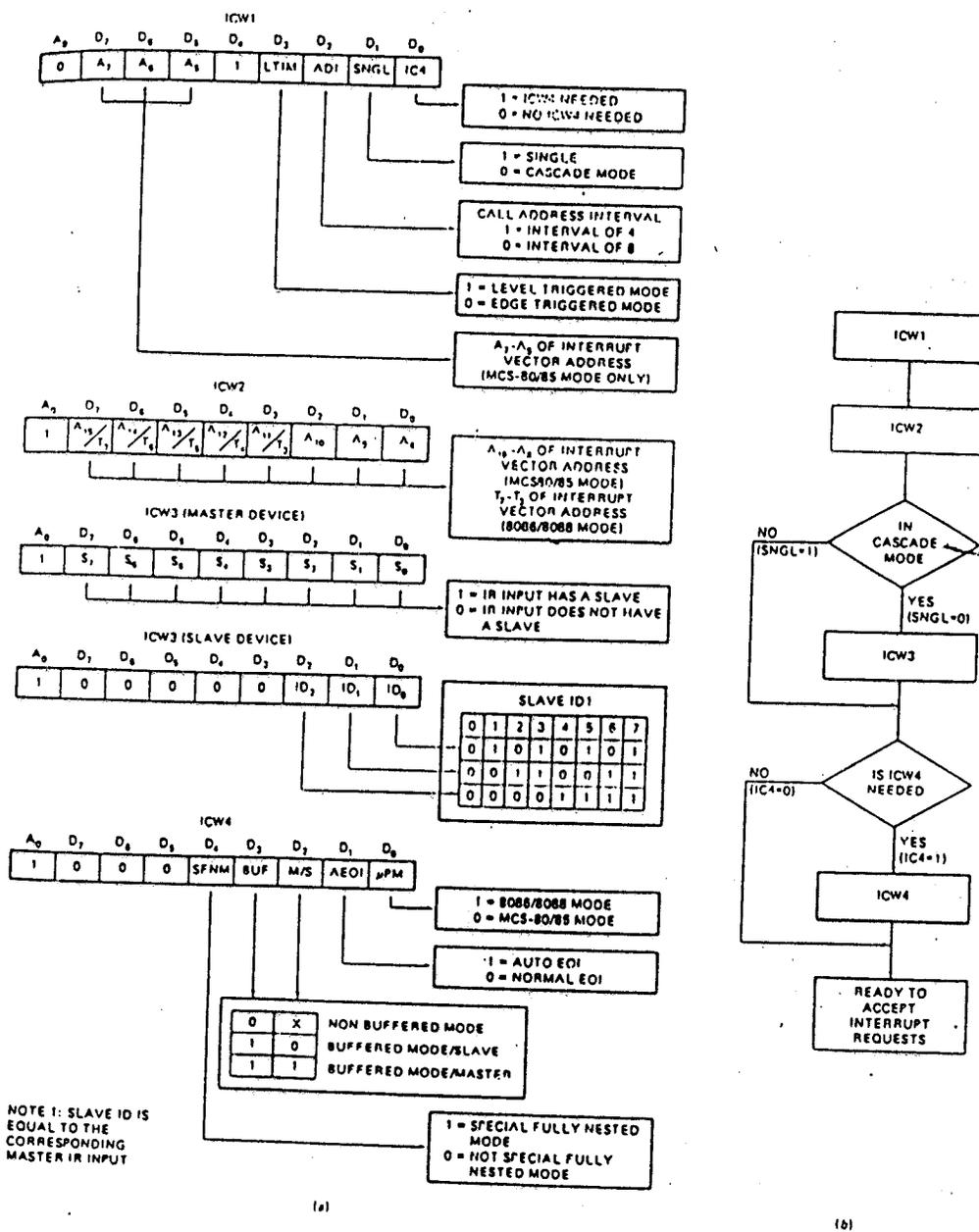
PIC 8259 ini memiliki 2 jenis Control Word yaitu Initialization Command Word (ICW) dan Operational Command Word (OCW). Format ICW dapat dilihat pada Gambar 2.19 sedang format OCW pada gambar 2.20.

## 2.10 PRINSIP KOMUNIKASI DATA SERIAL

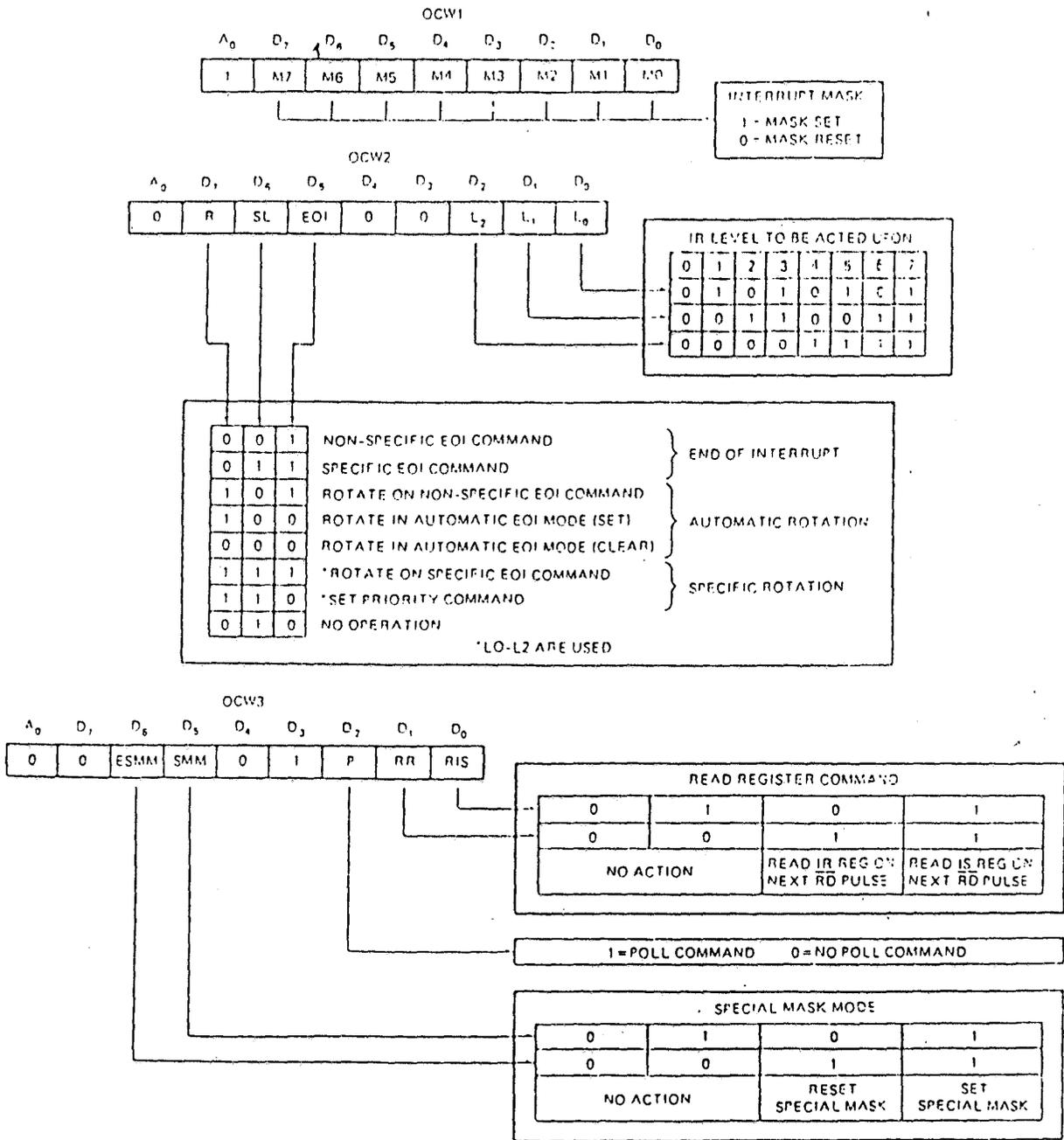
Pada komunikasi serial, data ditransmisikan melalui satu kabel, satu bit pada suatu waktu. Jadi bit-bit yang ada pada suatu byte data harus menunggu gilirannya untuk

<sup>13</sup> Ibid. hal. 250

dipindahkan. Ada dua metode komunikasi seri, yaitu sinkron dan asinkron.



Gambar 2.19  
Format Initialization Command Word



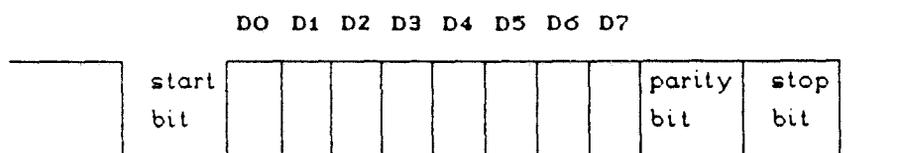
Gambar 2.20

Format Operational Command Word

Pada komunikasi sinkron data dikirim dalam suatu blok-blok dengan kecepatan tetap.

Kecepatan yang tetap ini diperoleh dengan mensinkronkan pengirim dan penerima. Tidak ada bit-bit khusus, sebab penerima mengetahui bahwa setiap 8 bit yang diterima setelah sinkronisasi, adalah mewakili satu byte data.

Pada komunikasi asinkron, setiap data karakter mempunyai satu bit khusus yang menandai awal data dan satu atau dua bit lain yang menandai akhir data, karena setiap data karakter mempunyai tanda tersendiri, maka data dapat dikirim pada sebarang waktu. Gambar 2.20 memperlihatkan bagaimana suatu data karakter dikirim. Ketika tidak ada yang dikirimkan, level sinyal adalah tinggi. Keadaan ini disebut *marking state*. Awal dari data ditandai dengan keadaan sinyal menjadi rendah selama satu bit, bit ini dinamakan bit start. Kemudian bit demi bit dikirim mulai dari *least significant bit* (LSB). Jumlah bit yang terkandung dalam satu karakter data tergantung dari sistem yang digunakan. Bit paritas digunakan untuk mendeteksi kesalahan data yang diterima. Pada akhir data karakter, sinyal menjadi tinggi selama satu atau dua bit, bit ini dinamakan bit stop. Bit stop ini mengembalikan sinyal pada kondisi marking state dan siap memulai data baru.



Gambar 2.21

Format Data Karakter pada Komunikasi Asinkron

Misalkan dipakai konfigurasi 1 bit start, 7 bit data dan 2 bit stop, tanpa bit paritas, berarti 10 bit dikirim untuk 1 byte. Jadi jika kecepatan transmisi adalah 9600 bit per detik, maka akan

dipindahkan 960 byte/detik.

### 2.10.1 Standart Komunikasi RS-232

Komunikasi data melalui port komunikasi pada IBM PC menggunakan standard protokol RS 232. Standard ini dikembangkan oleh *Electronic Industries Association*. Sistem ini semula digunakan untuk mengatur hubungan antara komputer dan modem. Tetapi perkembangan selanjutnya, RS 232 juga digunakan untuk piranti peripheral lainnya misal printer, ploter, ataupun mouse.

RS 232 menggunakan logika negatif, yaitu tingkat logika 1 berada dalam daerah tegangan -3 Volt sampai -15 Volt, dan tingkat logika 0 berada dalam daerah 3 Volt sampai 15 Volt.

### 2.10.2 Hubungan RS-232

RS 232 mempunyai 25 pin, namun tidak mutlak harus digunakan semua, tergantung dari sistem yang digunakan. Dari ke 25 pin tersebut dapat digolongkan atas 4 fungsi yaitu : ground, pertukaran data, kontrol dan pewaktu.

Tabel 2.2 memperlihatkan nama sinyal dari DTE (*Data Terminal Equipment*). Pin yang paling umum digunakan dalam suatu sistem yang sederhana adalah sinyal TxD, RxD, dan sinyal handsaking. Tetapi pada beberapa sistem tersebut tidak diperlukan sinyal handsaking. Pin sinyal TxD (no.2) digunakan untuk transmisi data serial, sedang pin sinyal RxD (no.3) digunakan untuk menerima data serial. Sinyal-sinyal handsaking umumnya digunakan antara komputer dan modem.

Tabel 2.2 Pin-pin pada RS-232

Nama	Deskripsi	No. Pin	Arah
TXD	Transmitted Data	2	out
RXD	Received Data	3	in
RTS	Request to Send	4	out
CTS	Clear to Send	5	in
DSR	Data Set Ready	6	in
GND	Signal Ground	7	-
CD	Carrier Detect	8	in
DTR	Data Terminal Ready	20	out

### 2.10.3 Port Komunikasi Asinkron pada IBM PC

Pada komputer IBM PC terdapat suatu interface serial komunikasi. Port komunikasi ini menggunakan standard RS 232 yang merupakan fasilitas standard bagi tiap komputer IBM PC Interface ini menempati alamat yang telah ditentukan oleh sistem IBM PC (Tabel 2.3)

Tabel 2.3 Alamat Interface serial komunikasi

Designation	Range Alamat
COM 1	3F8H - 3FFH
COM 2	2F8H - 2FFH
COM 3	3E8H - 3EFH
COM 4	2E8H - 2EFH

Port 3F8h, digunakan untuk mengirim dan menerima data, dan dapat digunakan juga (jika bit 7 port 3FBh diset) untuk menentukan baud rate peralatan komunikasi. Bilangan pembagi tersebut digunakan untuk membagi sistem clock dari peralatan komunikasi untuk mendapatkan baud rate yang diinginkan.

Port 3F9h juga digunakan untuk menentukan byte tinggi (jika bit 7 port 3FBh di set) bagi pembagi baud rate. Jika bit 7 port 3F9h di-clear maka akan berfungsi sebagai interrupt enable register.

Port 3FBh, *Line Control Register*, digunakan untuk menentukan format data yang dikirim atau yang diterima melalui port komunikasi. Port 3FDh, *Line Status Register*, digunakan untuk menandai keadaan transfer data.

Tabel 2.4

Line Control Register Port 3F8h (atau 2F8h, 3E8h, 2E8h)

Bit	7	6	5	4	3	2	1	0	Arti
	0								Akses normal ke port 3F8h/3F9h
	1								Memakai port 3F8h/3F9h unt. baud rate divisor
		1							Kondisi Transmit Break
		0							Parity pada bit 4
			1						Parity beroperasi normal
			0						Parity ganjil
				1					Parity genap
					0				Parity disabled
						1			Parity enable
							0		1 Stop bit
								1	2 Stop bit
							0	0	5-Bit data
							0	1	6-Bit data
							1	0	7-Bit data
							1	1	8-Bit data

Tabel 2.5

Line Status Register - Port 3FDh (2FDh, 3EDh, atau 2edh)

Bit	7	6	5	4	3	2	1	0	Arti
	0								Akses normal ke port 3F8h/3F9h
		0							Transmitter Shift Register penuh
			1						Transmitter Shift Register kosong
				0					Transmitter Holding Register penuh
					1				Transmitter Holding Register kosong
						1			Kondisi Break terdeteksi
							1		Framming Error
								1	Parity Error
								1	Overrun Error
								0	Tidak ada karakter siap
								1	Data siap diterima

Sebelum mengoperasikan port komunikasi, maka harus dilakukan inisialisasi terlebih dahulu. Inisialisasi ini digunakan untuk menentukan *baud rate* sistem, format data, dan metode komunikasi dengan PC. Setelah dilakukan inisialisasi maka port komunikasi digunakan untuk mengirim maupun menerima data.

#### 2.10.4 Inisialisasi Port Komunikasi

Dalam inisialisasi, data-data yang digunakan untuk inisialisasi dikirim pada line control register. Misalkan pada suatu sistem komunikasi digunakan baud rate 300, 600, 1200, 7 bit data, parity genap, 1 start bit dan 2 stop bit, dan mode komunikasi dengan PC ialah pooling.

#### 2.10.5 Membaca Data dari Port Komunikasi

Setelah dilakukan inisialisasi maka port komunikasi telah siap menerima atau mengirim data. Untuk dapat menerima data maka harus dilakukan test untuk kesiapan data dan adanya kesalahan misalnya parity atau overrun. Test ini dilakukan pada Line Status Register. Jika semua test memenuhi syarat maka data yang valid dapat dibaca dari data register.

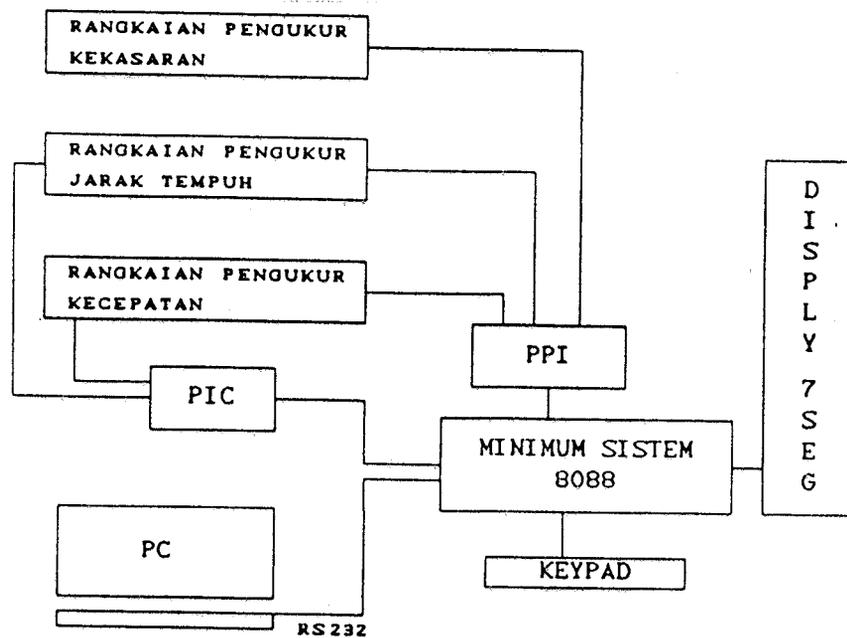
#### **2.10.6 Mengirim Data Melalui Port Komunikasi**

Untuk dapat mengirim data maka harus dilakukan test untuk kesiapan register data pada port komunikasi agar tidak terjadi kesalahan pengiriman data. Status register yang dilihat ialah kosongnya Transmit ialah kosongnya *Transmit Shift Register* dan *Transmit Holding Register*. Test ini dilakukan pada Line Status Register. Jika semua test memenuhi syarat maka data dikirim ke data register.

# BAB III

## PERENCANAAN HARDWARE DAN SOFTWARE

Urutan langkah pengerjaan hardware secara umum dapat dilihat dari blok diagram dibawah ini.



Gambar 3.1  
Blok diagram Perencanaan Hardware

Sampel kekasaran jalan diambil dari sensor Infra merah yang mendeteksi jumlah kisi-kisi piringan yang bergerak memutar searah akibat gerakan vertikal (naik turun) gardan kendaraan terhadap lantai mobil. Nilai kekasaran merupakan hasil penjumlahan kisi-kisi yang terdeteksi dengan menggunakan sejumlah IC Decade Counter 74ls90.

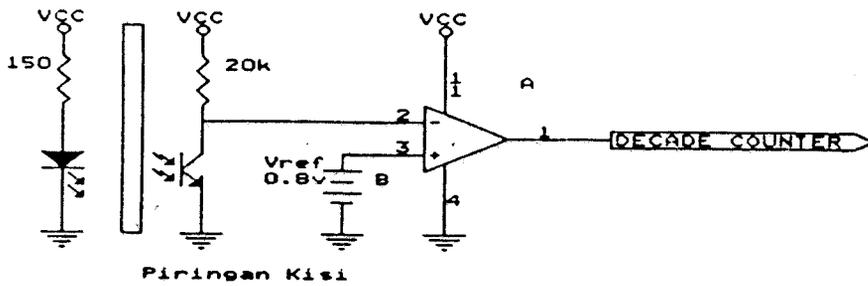
Kode Biner yang dihasilkan diubah ke penggerak 7-segment dengan menggunakan IC Decoder BCD ke 7-segment 74ls47 untuk ditampilkan pada display. Disamping itu data biner tersebut juga diambil untuk diolah lebih lanjut pada rangkaian minimum sistem 8088 yang selanjutnya data-data tersebut dikirim untuk ditampilkan pada monitor PC melalui komunikasi serial RS232.

Nilai kekasaran ditampilkan setiap ada interrupt dari pengukur jarak (*Odometer*) tiap 1 meter tempuh. Sedangkan data pengukur kecepatan diambil guna menentukan kecepatan yang paling sesuai saat pengukuran kekasaran.

### **3.1 Sensor Infra Merah**

Sensor atau pendeteksi yang direncanakan untuk dipakai pada alat ini adalah infra merah. Sensor ini ditempatkan sedemikian rupa sehingga dapat mendeteksi kisi-kisi dari piringan yang bergerak-gerak berputar searah melintasi celah sensor.

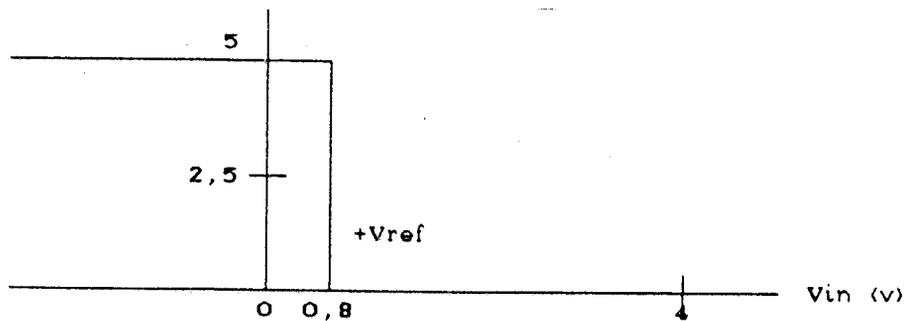
Sensor infra merah akan mendeteksi buka tutup dari piringan kisi untuk diubah menjadi pulsa-pulsa digital. Pada saat perubahan keadaan dari buka ke tutup atau sebaliknya intensitas cahaya infra merah yang diterima phototransistor berubah-ubah sangat cepat, sedangkan respon time dari phototransistor sangat lambat sehingga menyebabkan terjadinya osilasi atau pulsa cacat pada outputnya, dimana kondisi ini tidak diinginkan. Untuk mengatasi hal ini pada output ditambahkan rangkaian detektor pembalik dengan tegangan referensi positif LM324 seperti pada gambar 3.2.



Gambar 3.2  
Sensor Infra merah dan Detektor pembalik LM324

Dengan tegangan referensi sebesar 0,8V, sehingga output hanya mempunyai tegangan 0V dan 5V saja yang menyatakan logic 0 dan 1. Pengambilan tegangan referensi sebesar 0,8 V ini didasarkan pada tegangan osilasi distorsi output phototransistor yang berkisar antara 0 V sampai 0,8 V, yang oleh input detektor dianggap berlogika 0. Respon tegangan dari detektor ini dapat dilihat pada gambar 3.3.

Pada alat ini digunakan piringan kisi dengan jumlah kisi 145, diameter piringan 7 cm dan kerapatan kisi 1,1 mm.

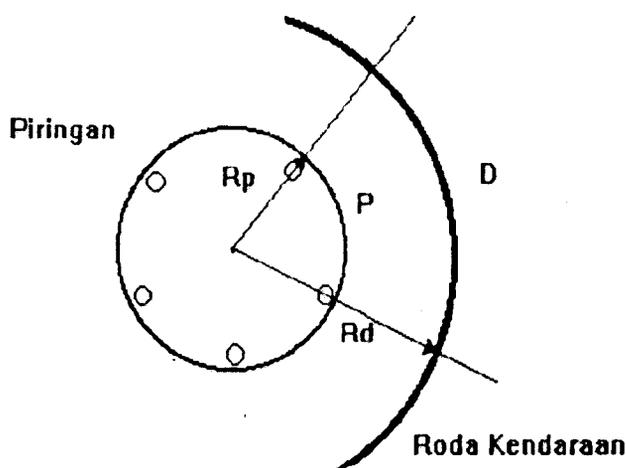


Gambar 3.3  
Tegangan input output dari Detektor LM324

Sensor Infra merah mendeteksi jumlah kisi setiap gerakan putar dengan

ketelitian 1,1 mm. Penjumlahan dari setiap pulsa akibat getaran inilah yang menjadi dasar nilai pengukuran kekasaran jalan dengan satuan Count/m yang nantinya dikonversi ke indeks kekasaran atau IRI (*International Roughness Index*) mm/km.

Disamping digunakan sebagai pendeteksi kekasaran, sensor Infra merah ini juga digunakan sebagai sensor pengukur jarak tempuh kendaraan (*Odometer*). Sensor ini akan mendeteksi jumlah lubang yang diklam pada as roda kendaraan. perkalian antara jumlah lubang piringan dengan jarak antar lubang menyatakan keliling dari piringan yang dideteksi, sama dengan jarak yang telah ditempuh piringan dalam satu kali putaran, sedangkan untuk menentukan jarak tempuh roda kendaraan yaitu jarak tempuh dari kendaraan sesungguhnya harus dikalikan dengan konstanta pembanding antara jari-jari roda kendaraan dengan jari-jari piringan seperti penjelasan dibawah ini.



$$\frac{P}{D} = \frac{R_p}{R_d}$$

$$D = P \times \frac{R_d}{R_p} \text{ (cm)}$$

$R_p$  = Jari-jari piringan (cm)

$R_d$  = Jari-jari Roda kendaraan (cm)  
 $D$  = Lintasan Roda kendaraan (cm)  
 $p$  = Jarak antar lubang pada Piringan (cm)

Pada perencanaan hardware disini digunakan piringan dengan jari-jari ( $R_p$ ) 8 cm (keliling 50 cm). Pada bagian tepi dibuat lubang sebanyak 5 dengan jarak antar lubang ( $P$ ) 10 cm. Jika jari-jari roda kendaraan surcai  $R_d$  cm, maka lintasan roda kendaraan ( $D$ ) yang ditempuh (dalam 1 pulsa) :

$$D = 10 \times \frac{R_d}{8} = 1,25R \quad (\text{cm})$$

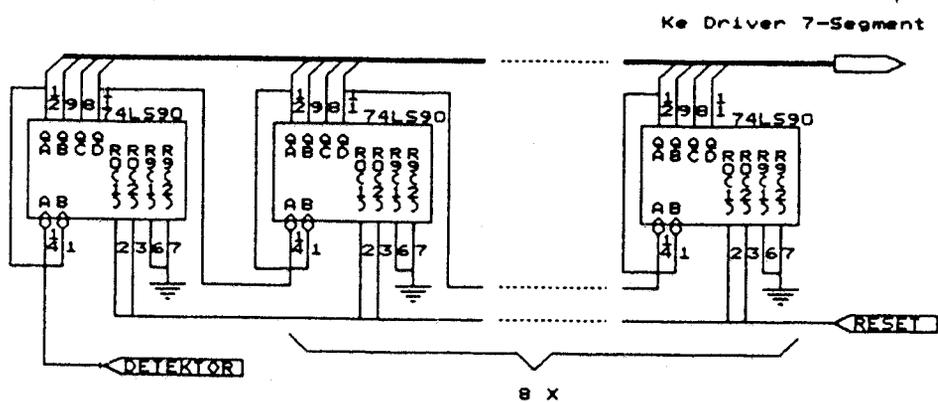
### 3.2 DECADE COUNTER

Output dari sensor yang berupa pulsa digital dimasukkan ke rangkaian pencacah sepuluh (*Decade Counter*) dengan menggunakan IC 74LS90. Pencacah ini mempunyai sepuluh keadaan dan karena itu akan membagi frekuensi masukan dengan sepuluh. Output dari IC ini merupakan hasil cacahan asinkron dalam kode BCD. Isi setiap pencacah dekade pertama-tama disimpan dalam 4 bit dan dipertahankan sampai input mendapat triger ujung pulsa turun.

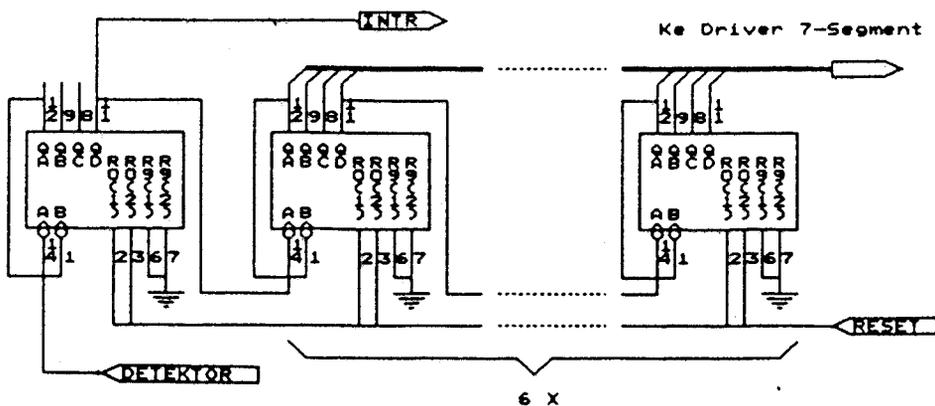
Dalam perencanaan alat ini digunakan IC Decade Counter sebanyak 8 buah untuk konter data kekasaran (nilai ukur kekasaran), 7 buah untuk konter jarak (nilai ukur jarak). Rangkaian dapat dilihat seperti pada gambar 3.4 dan gambar 3.5.

Tiap-tiap IC decade counter ini dihubungkan ke driver 7- segment 74LS47 yang mengkonversi kode BCD ke dalam kode 7-segment.

Pada Rangkaian decade counter data kekasaran IC 74LS90 yang pertama digunakan untuk menampilkan kode BCD yang berlaku sebagai satuan desimal. Output bit tinggi dari IC yang pertama ini dihubungkan sebagai input dari IC decade konter yang kedua yang akan menggerakkan counter kedua setiap perubahan sepuluh desimal. IC kedua ini akan berlaku sebagai nilai puluhan dari pengukuran. Demikian seterusnya bit tinggi dihubungkan ke counter selanjutnya yang akan berlaku sebagai nilai ratusan, ribuan, sampai puluhan juta.



Gambar 3.4  
Rangkaian Decade Counter Data Kekasaran



Gambar 3.5  
Rangkaian Decade Counter Jarak

Pada rangkaian decade counter untuk jarak IC 74LS90 yang pertama tidak terhubung ke penggerak &-segment 74LS47 akan tetapi hanya digunakan sebagai konstanta pengali 10 dari sepuluh pulsa yang terdeteksi pada piringan, sedangkan jarak antar pulsa direncanakan 10 cm, sehingga pengukuran jarak ini mempunyai konstanta pengali 10 x 10 cm sama dengan 100 cm sama dengan dikalikan 1 m untuk setiap satuan. Counter selanjutnya terhubung ke bit tinggi dari counter sebelumnya sama seperti pada counter data yang telah dijelaskan diatas.

IC Decade Counter jarak yang pertama output bit tinggi selain untuk menggerakkan counter-counter selanjutnya juga digunakan sebagai pembangkit interrupt pada PIC 8259 yang akan menginterrupt setiap kendaraan menempuh jarak 1 m. Interrupt ini digunakan untuk membandingkan jarak yang telah ditempuh dengan set jarak yang telah ditentukan sebelumnya pada pencatatan dan pengiriman data kekasaran ke PC.

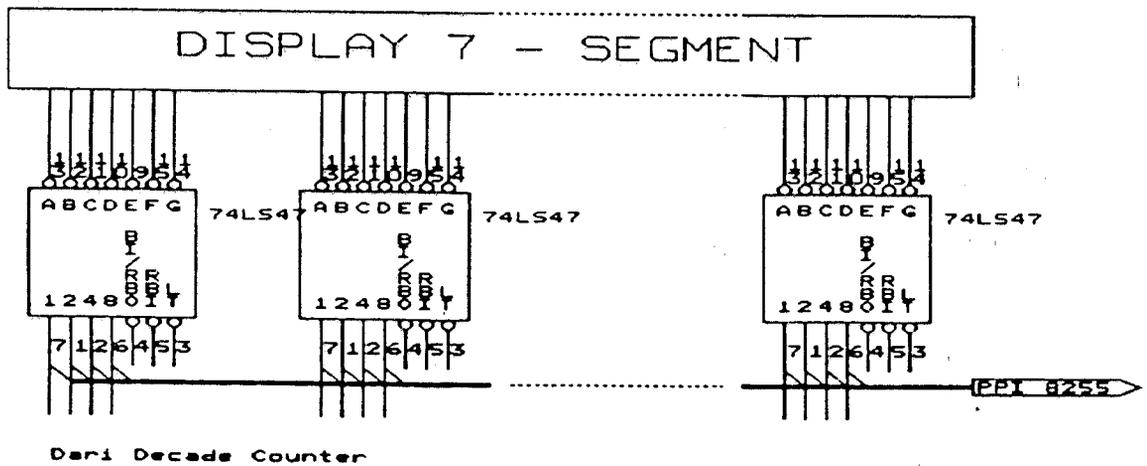
### **3.3 PERAGA (DISPLAY) DAN PENGGERAKNYA**

Peraga untuk menampilkan nilai numerik dari hasil pengukuran digunakan sederetan 7-segment disamping monitor PC. Dalam perencanaan alat ini terdapat tiga deretan alat peraga 7-segment. Peraga pertama digunakan untuk menampilkan nilai hasil penyampelan kekasaran permukaan jalan, peraga kedua untuk menampilkan jarak yang ditempuh kendaraan selama survey, dan peraga ke tiga untuk menampilkan set jarak untuk pencatatan nilai kekasaraan selama survey.

Untuk peraga kekasaran digunakan 8 buah 7-segment, peraga jarak digunakan 6 buah 7-segment, sedangkan untuk peraga set jarak digunakan 4 buah 7-segment.

Rangkaian penggerak dapat dilihat pada gambar 3.6.

Penggerak peraga 7-segment dalam hal ini digunakan IC 74LS47 yaitu IC



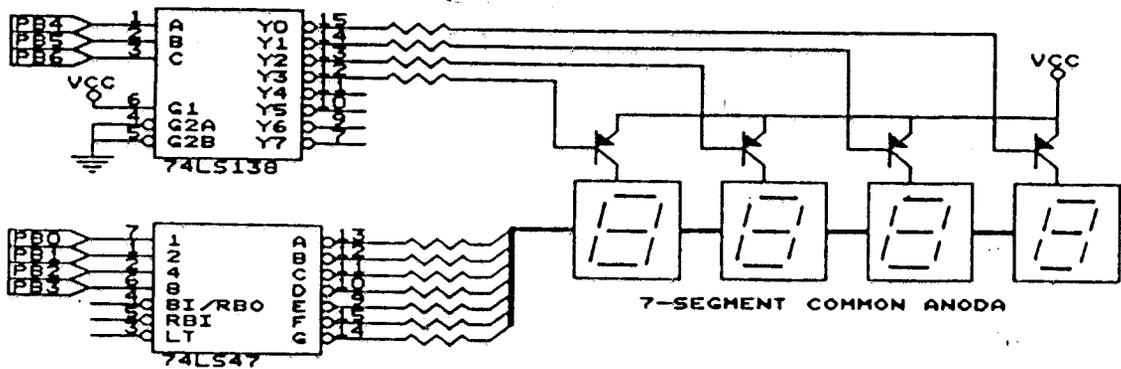
Gambar 3.6  
Rangkaian Peraga 7-segment  
Untuk Kekasaran dan Pengukuran Jarak

decoder BCD ke 7-segment, dimana IC ini mengubah kode BCD 4 bit (DCBA) dari tiap-tiap IC pencacah (Decade Counter) hasil pengukuran ke kode 7-segment (abcdefg) untuk peraga kekasaran dan peraga jarak, sedangkan peraga set jarak kode BCD diambil dari pemrograman dari minimum sistem, dengan metode scanning tiap-tiap 7-segment. Data BCD diambil dari pemrograman melalui IC Buffer 74LS245, sedangkan scan tiap-tiap 7-segment diatur oleh IC decoder 74LS138 yang berlaku sebagai Multiplexer.

### 3.4 PERENCANAAN PENGUKUR KECEPATAN

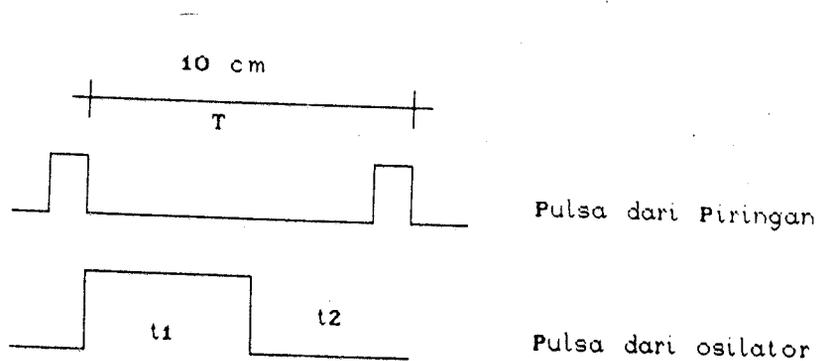
Rangkaian pengukur kecepatan didasarkan pada perhitungan pulsa dari osilator pada selang jarak antar lubang pada piringan. Sebagai pembangkit osilator digunakan rangkaian astable multivibrator (multivibrator tak stabil) dari IC NE 555. Frekuensi

yang dihasilkan ditentukan oleh perhitungan dari rangkaian eksternal RA, RB dan C.



Gambar 3.7  
Rangkaian Peraga untuk Set Jarak

Jarak antar lubang pada piringan adalah 10 cm jarak ini dideteksi oleh sensor infra merah, sehingga akan dihasilkan pulsa-pulsa dengan jarak antar pulsa 10 cm. Kecepatan yang diinginkan adalah 30 Km/jam, yang berarti bahwa waktu yang dibutuhkan untuk menempuh jarak 30 km adalah 1 jam atau sama dengan 3600 detik. Sedangkan jika jarak yang dilalui sepanjang 10 cm (jarak antar pulsa piringan) maka dibutuhkan waktu sebesar 0,012 detik. Dalam waktu tersebut minimal terdeteksi 1 pulsa dari osilator. Periode minimal osilator sama dengan 0,012 detik atau sama dengan 83,3 Hz.



$$V = 30 \text{ km/jam}$$

$$S = 10 \text{ cm}$$

$$T = \frac{S}{V} = \frac{10 \times 3600}{3000000} = 0,01 \text{ detik}$$

$$T_{\min} = t_1 + t_2 = 0,012 \text{ detik}$$

$$F_{\min} = \frac{1}{0,01} = 83,3 \text{ Hz}$$

$$F_{\text{osc}} \geq F_{\min}$$

$$\frac{F_{\text{osc}}}{F_{\min}} = n \quad \text{dimana } n = \text{jumlah pulsa yang terdeteksi}$$

Pada perencanaan rangkaian disini, diambil  $n = 15$  pada kecepatan 30 Km/jam, maka frekuensi osilator ( $F_{\text{osc}}$ ) dapat dihitung,

$$\begin{aligned} F_{\text{osc}} &= n \times F_{\min} \\ &= 15 \times 83,3 \text{ Hz} \\ &= 1,25 \text{ KHz} \end{aligned}$$

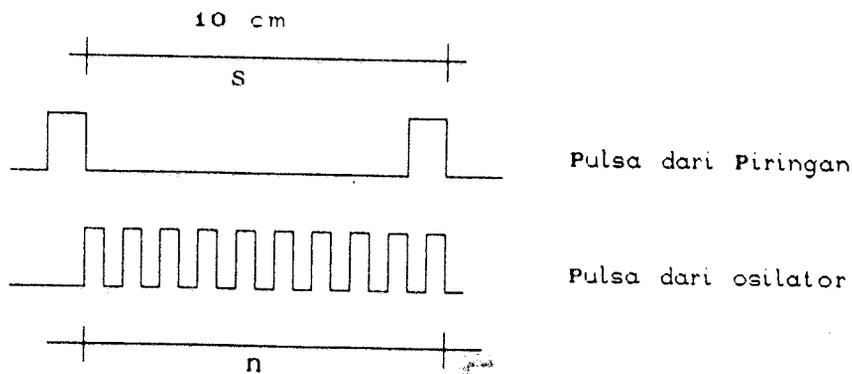
Untuk mendapatkan osilator dengan frekuensi tersebut dengan mengambil nilai tahanan  $R_B$  sebesar 5,2 Kohm dengan kapasitor  $C$  sebesar 100 nF. maka  $R_A$  dapat dihitung seperti dibawah :

$$\begin{aligned} T &= t_1 + t_2 = 0,7 (R_A + R_B) C + 0,7 R_B C \\ 8 \cdot 10^{-4} &= 1,4 \times 5,2 \cdot 10 \times 100 \cdot 10^{-9} + 0,7 R_A \times 100 \cdot 10^{-9} \\ 8 \cdot 10^{-4} &= 7,28 \cdot 10^{-4} + 7 \cdot 10^{-8} R_A \\ R_A &= 1028,6 \text{ Ohm} \end{aligned}$$

$R_A$  diganti dengan VR-multiturn sebesar 1,5 Kohm.

Rangkaian osilator ini dapat dilihat pada gambar 3.7.

Hubungan antara jumlah pulsa dengan kecepatan dapat dijelaskan sebagai berikut :



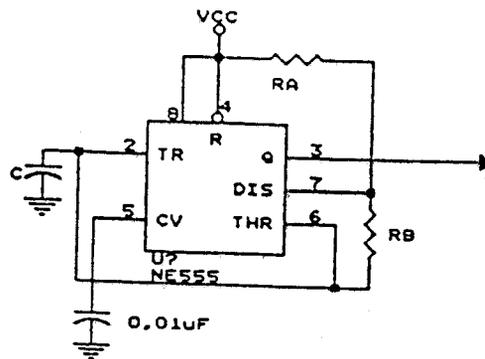
$$\begin{aligned}
 V &= \frac{S}{T} \\
 &= \frac{10 \times F_{osc}}{n} \text{ cm/det} \\
 &= \frac{10 \times 1,25 \cdot 10^3 \times 3600 \times 10^{-7}}{n} \text{ Km/jam} \\
 &= \frac{450}{n} \text{ Km/jam} \quad ; (n = 1, 2, \dots, 255)
 \end{aligned}$$

Artinya jika jumlah pulsa (n) yang terdeteksi 15 maka kecepatan sebenarnya adalah 30 Km/jam. Untuk n sama dengan 1, V sama dengan 450 Km/jam, untuk n sama dengan 255 (jumlah maksimum dalam 1 byte) V sama dengan 1,76 Km/jam. Maka dapat ditentukan bahwa alat pengukur kecepatan yang direncanakan dapat mengukur kecepatan antara 1,76 Km/jam sampai 450 Km/jam.

Data kecepatan diambil pada hitungan ke 15 dari IC pencacah-16 74ls93 yang mendapat clock dari sensor infra merah yang mendeteksi piringan. Output BCD dari IC ini digunakan untuk mereset IC 74ls393 dan sekaligus untuk menahan (latch) data pada IC 74ls373 (data valid).

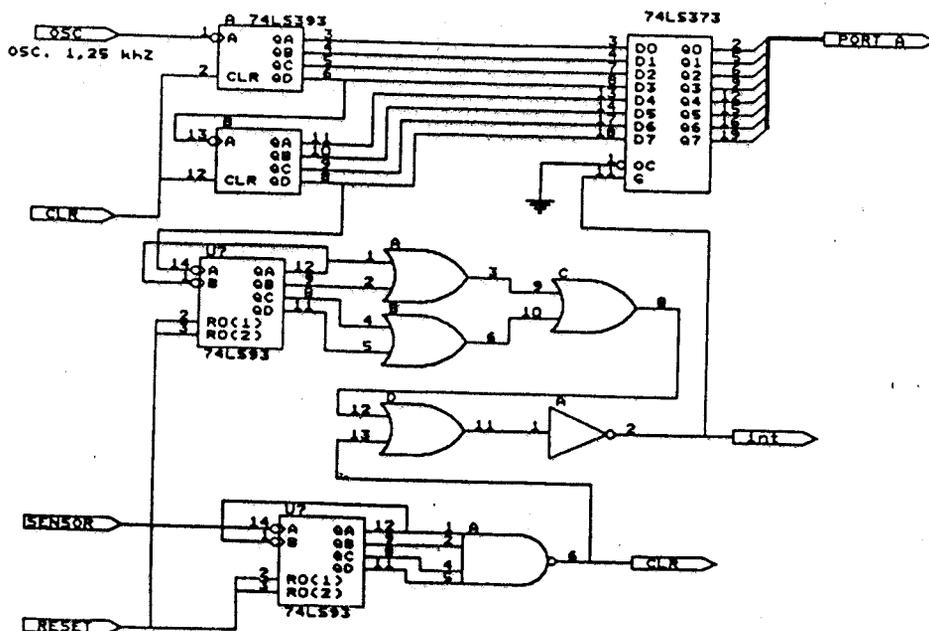
Pada saat hitungan ke 15 dari IC 74ls93, IC counter 74ls393 memperbolehkan

osilator NE 555 untuk masuk. IC counter 74ls393 akan menghitung jumlah pulsa (pulsa osilator) sampai pada keadaan reset kembali. Jumlah pulsa yang terhitung disini adalah sama dengan jumlah frekuensi

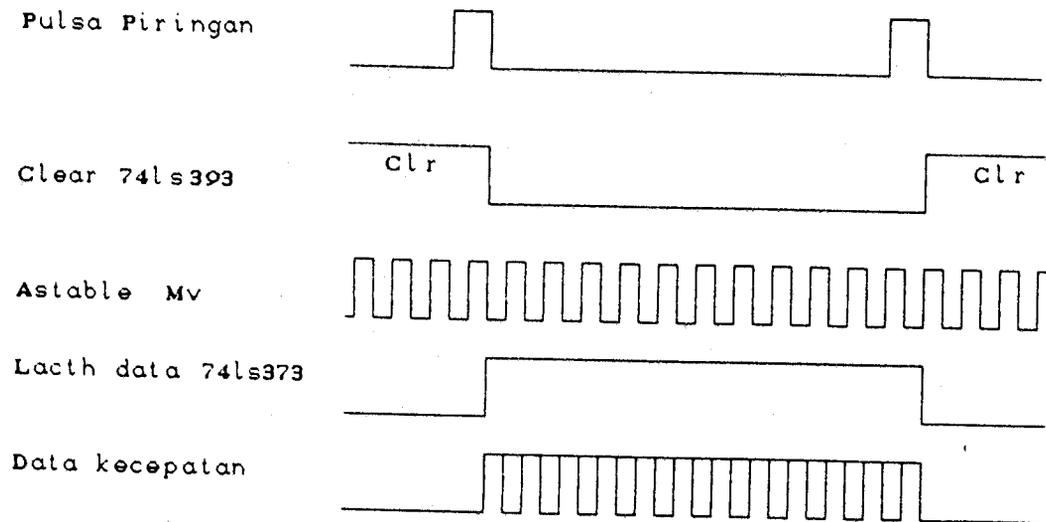


Gambar 3.7  
Astable Multivibrator NE 555 sebagai osilator

yang terdeteksi selama satu pulsa piringan. Jumlah pulsa ini kemudian dikonversi kedalam waktu yaitu waktu yang diperlukan untuk menempuh satu pulsa piringan (10 cm) atau sama dengan besaran kecepatan. Rangkaian pengukur kecepatan ini dapat dilihat pada gambar 3.8.



Gambar 3.8  
Rangkaian Pengukur Kecepatan



Gambar 3.9  
Sinyal pada rangkaian pengukur kecepatan

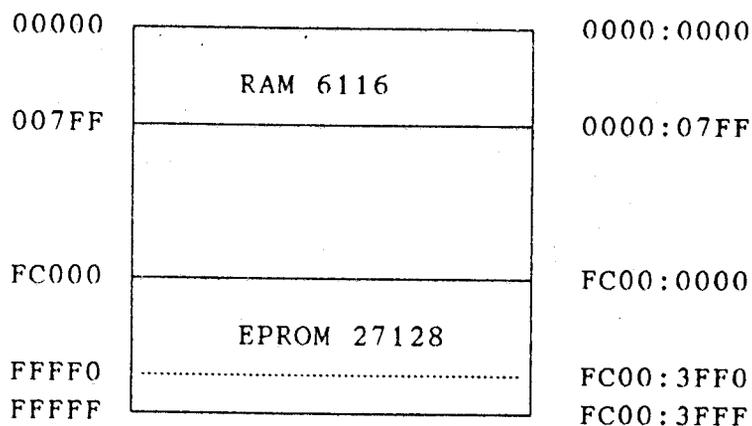
### 3.5 CENTRAL PROCESSING UNIT MINIMUM SISTEM 8088

Sebagai prosesor pengendali digunakan minimum sistem 8088 yang dilengkapi dengan sebuah RAM dan sebuah ROM. Memori yang direncanakan menggunakan RAM Statis 6116 dan satu EPROM 27128. Memori mapping yang direncanakan dapat dilihat pada gambar 3.10.

EPROM diletakkan pada address dengan lokasi tertinggi hal ini dilakukan dengan alasan setelah terjadi reset pada mikroprosesor 8088, mikroprosesor akan menjalankan program pada physical address FFFF0h. Pada lokasi memori FFFF0h

diletakkan instruksi jump far ke awal program sehingga program yang telah direncanakan dapat dijalankan.

Program yang direncanakan diletakkan pada EPROM dengan lokasi physical address FC000h - FFFFFh (16 kbyte), sedangkan lokasi 00000h - 007FFh (2 kbyte) digunakan sebagai RAM untuk data dan stack.



Gambar 3.10  
Memory Map minimum sistem 8088

Internal memori dan zone bit dari RAM 6116 dan EPROM 27128 bila digunakan sistem full decoding, maka seluruh zone bit ikut diperhitungkan.

Perangkat luar seperti decade counter data kekasaran, dan jarak tempuh, display set-jarak, tombol dan alarm dikontrol melalui CPU ini dengan perantara rangkaian interface PPI 8255 dengan pengalamatan yang saling disesuaikan sehingga tidak ada alamat yang sama. Semua proses dilaksanakan sesuai dengan program yang disimpan pada EPROM.

### 3.6 DECODER

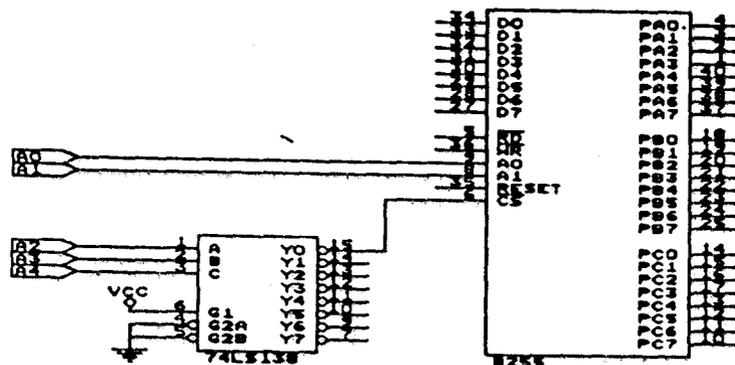
Hardware pada tugas akhir ini digunakan 2 buah PPI 8255, masing-masing

dialamatkan ke 000h - 003h untuk PPI yang menjadi satu dengan board minimum sistem dan 224h - 227h yang dibuat tersendiri dan ditempatkan pada slot dari minimum sistem. Desain rangkaian decoder untuk pengalamatan PPI tersebut didasarkan pada tabel logika seperti yang ditunjukkan pada tabel 3.1.

Decoder untuk alamat 000h - 003h cukup digunakan sebuah IC decoder 74ls138 dengan menghubungkan chip select PPI 8255 ke Y0 dari pin 74ls138. Sedangkan decoder untuk alamat 224h - 227h digunakan 4 buah IC decoder 74ls139 yang dirangkai seperti pada gambar 3.11.

Tabel 3.1 Logika Pengalamatan 000h-003h dan 224h-227h

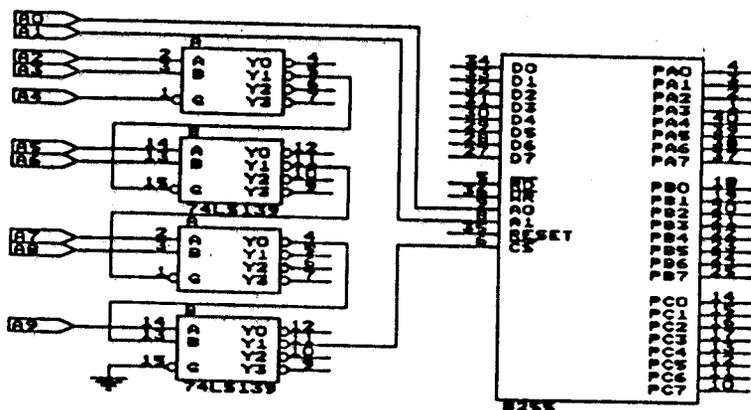
A9	A8	A7	A6	A5	A4	A3	A2	A1	A0	ADDRESS
0	0	0	0	0	0	0	0	0	0	000H
0	0	0	0	0	0	0	0	0	1	001H
0	0	0	0	0	0	0	0	1	0	002H
0	0	0	0	0	0	0	0	1	1	003H
0	0	0	0	0	0	0	0	0	0	224H
0	0	0	0	0	0	0	0	0	0	225H
0	0	0	0	0	0	0	0	0	0	226H
0	0	0	0	0	0	0	0	0	0	227H



Gambar 3.11  
Rangkaian Decoder Untuk Alamat 000h-003h

### 3.7 PROGRAMABLE PERIPHERAL INTERFACE 8255

Pada pengerjaan tugas akhir ini digunakan 2 buah PPI 8255, sebuah PPI dengan alamat 000h - 003h digunakan untuk mengambil data jarak tempuh dari 6 buah IC decade counter yang diinputkan ke port A secara multiplex. Sedangkan sebagai kontrol input multipleknya IC 74ls138 melalui port B (PB0 - PB2), sebagai kontrol display set jarak dipakai PB0 - PB3 untuk data input dan PB4 - PB6 sebagai scanning display. Sebagai pengontrol tombol (*push button switch*) yaitu Port C lower (PC0 - PC3) sedangkan kontrol alarm dan lampu pilot oleh Port C upper (PC4 - PC7). Sehingga alamat control word pada PPI ini diberikan data 10010001b, yang berarti PPI diset pada mode 0, Port A input, Port B output, Port C upper output dan Port C lower input.



Gambar 3.12  
Rangkaian Decoder Untuk Alamat 224h-227h

Sedangkan PPI yang lain dengan alamat 224h - 227h digunakan untuk mengambil data kekasaran dari 6 buah IC decade counter untuk ditransfer ke PC melalui komunikasi serial RS 232.

### 3.8 RANGKAIAN INTERRUPT (PIC 8259)

Pengambilan data dilakukan setiap 1 meter jarak tempuh dari kendaraan. Setiap counter mendeteksi sepuluh dari sepuluh kali pulsa piringan yang berarti piringan telah bergerak (berputar) menempuh jarak 100 cm (1m) maka hardware akan memberikan sinyal interrupt IRQ0 pada PIC 8259. Instruksi ini menyebabkan mikroprosesor menghentikan sementara programnya untuk melaksanakan program pengambilan data kekasaran kemudian mengirimkan data tersebut ke komputer.

IC 8259 diletakkan pada alamat 0C4h-0C5h. Interrupt ini dijalankan dengan program. Pada awal program interrupt diinisialisai terlebih dahulu untuk menentukan mode ataupun ketentuan lain yang diinginkan. Mula-mula di kirimkan ICW1 00010011b ke alamat 0C4h yang artinya ICW4 dibutuhkan, single bukan cascade, address interval 8 dan *edge triggered mode*. Pada ICW2 dikirimkan alamat interrupt

vektor dari PIC yaitu 08h.

Pada ICW4 dikirimkan control word 00001001b, yang berarti IRQ0 - IRQ7 memiliki prioritas yang sama, normal *End of Interrupt* (EOI) serta jenis mikroprosesor yang digunakan adalah 8088. Sedangkan untuk control word OCW1 dikirimkan data 11111110b, yang berarti IRQ2 - IRQ7 tidak digunakan.

Program untuk rangkaian interrupt diatur melalui prosedur program. Terdapat dua prosedur program untuk interrupt pada perencanaan software disini yaitu prosedur program interrupt untuk mengambil data jarak tempuh yang dilakukan oleh IRQ0 dan prosedur program interrupt untuk mengambil data kecepatan kendaraan yang dilakukan oleh IRQ1.

### 3.9 SERIAL KOMUNIKASI INTERFACE

Serial komunikasi yang direncanakan ini menggunakan card serial yang kompatibel dengan IBM Personal Computer, yaitu RS-232. Pada Tugas Akhir ini, serial komunikasi digunakan untuk menampilkan data hasil pengukuran dalam bentuk data-data numerik maupun grafik.

Data yang dikirim adalah data kekasaran yang diperoleh dari penjumlahan counter setiap 1 m tempuh kendaraan dengan satuan Count/m. Data diterima setiap ada interrupt IRQ0 PIC 8259 pada minimum sistem yang memerintahkan CPU mengirim data ke PC.

Pada inisialisasi RS-232 diatur alamat-alamat port yang dipilih. Disini dipilih

COM1 yaitu dari alamat 03F8h - 03FFh

Pada alamat Line Control Register dikirimkan data 10000000b, bit ke-7 diberi '1' untuk memilih *baudrate divisor*. Kemudian dikirim baudrate divisor 000Ch (12) dengan mengirimkan MSBnya lebih dulu pada alamat *Baudrate Divisor Latch*. Bilangan ini digunakan sebagai pembagi frekuensi clock pada IC INS 8250. Dengan mengirimkan baudrate divisor 000Ch berarti didapatkan baudrate sebesar 9600 bit/s. Kemudian dikirimkan 00000011b ke alamat *Line Control Register* yang berarti pengiriman data dalam 8 bit, 1 stop bit, dan parity disable. Dan yang terakhir dikirimkan data 00h ke alamat *Interrupt Enable Register* yang berarti Interrupt disable.

Pada pengiriman data serial ini pengiriman diawali dengan karakter kontrol untuk setiap jenis data yang dikirimkan karakter 'DDh' adalah karakter kontrol pengiriman data kekasaran, sedangkan 'EEh' adalah karakter kontrol pengiriman data kecepatan. Karakter kontrol 'CCh' digunakan untuk mengakhiri pekerjaan. Data yang dikirimkan melalui serial komunikasi ini adalah output dari Decade Counter dari pengukuran kekasaran maupun dari pengukuran jarak dan data kecepatan. Sebelum pengiriman data selalu dilakukan pengecekan pada Tx Buffer dan Tx Shift Register, pengecekan ini dilakukan dengan cara mengambil data yang ada pada line control register, bit ke-5 menunjukkan kondisi Tx Buffer, jika bit ke-5 '1' berarti Tx Buffer sudah kosong. Sedangkan bit ke-6 menunjukkan kondisi Tx Shift Register, jika bit ini '1' berarti Tx ShiftRegister kosong akan tetapi bila '0' berarti pada Tx masih ada data yang belum semuanya dikirim harus ditunggu sampai benar-benar kosong untuk

menghindari kesalahan data yang dikirim. Setelah mengetahui bahwa Tx buffer telah kosong, baru data mulai dikirimkan dan selalu diawali karakter kontrol awal dan akhir seperti yang telah disebut diatas. Data yang berada diantara karakter-karakter kontrol inilah data yang sesungguhnya. Sebelum pengiriman, data yang tadinya berupa biner diubah lebih dulu menjadi data desimal pada CPU minimum sistem baru dikirimkan. Pada PC data tersebut langsung ditampilkan dalam numerik atau diolah dulu untuk dibuat plot grafiknya pada monitor.

Pengiriman data ini secara otomatis data langsung dikirimkan setiap 1 meter jarak yang telah ditempuh dan terdapat penandaan khusus untuk data yang dicatat setiap besaran jarak yang telah ditentukan pada awal pengoperasian ('set jarak').

Hasil komunikasi serial ini diolah dengan menggunakan bahasa pemrograman tingkat tinggi, yaitu bahasa Pascal. Hal ini untuk mempermudah pengolahan data dan tampilan grafik.

### 3.10 TOMBOL DAN DISPLAY

Tombol dan display yang akan dipakai pada Tugas Akhir ini dikontrol menggunakan PPI 8255. Tombol yang diperlukan pada peraga adalah 6 buah, 1 buah berupa tombol on/off dan 5 buah berupa tombol push button. Tombol-tombol ini dikontrol melalui port B yang dirangkai pull up menggunakan resistor 3,3 kW dan Vcc 5 Volt sisi yang lain dihubungkan ke ground kemudian outputnya dihubungkan ke *buffer inverting* untuk membalik kondisi. Tombol ini dicek setiap saat oleh

mikroprosesor melalui port B PPI. Dalam keadaan tidak ada tombol yang ditekan, input yang masuk ke port B semua "low" dan apabila ada satu tombol yang ditekan, maka rangkaian akan langsung short ke ground, karena sinyal melalui buffer inverting maka logikanya akan dibalik, sehingga port yang tersambung dengan tombol tersebut akan mendapatkan input "high" dan mikroprosesor akan menjalankan program sesuai posisi tombol tersebut. Fungsi masing-masing tombol dapat dijelaskan sebagai berikut :

1. Tombol On/Off berupa switch on/off yang terhubung ke reset dari decade counter. Switch pada posisi "on" untuk mengawali pengukuran kekasaran maupun pengukuran jarak tempuh dengan terlebih dahulu men-set jarak pada pengambilan data kekasaran.
2. Tombol RESET terhubung ke reset dari minimum sistem untuk mereset semua sistem kembali ke awal pengukuran.
3. Tombol UP terhubung ke PC2 untuk memilih jarak pada saat pengambilan data. Jika tombol ini ditekan angka pada display "set jarak" akan naik dan siap untuk dipilih.
4. Tombol DOWN terhubung ke PC1 fungsinya sama dengan tombol UP. Jika tombol ini ditekan angka pada display "set jarak" akan turun dan siap untuk dipilih.
5. Tombol START terhubung ke PC3 untuk men-set jarak pengambilan data kekasaran. Jika tombol ini ditekan berarti pengambilan data sesuai dengan jarak yang tertera pada display "set jarak" yang telah dipilih sebelumnya melalui tombol UP atau DOWN.
6. Tombol STOP terhubung ke PC0 digunakan untuk mengakhiri pengukuran. Jika

tombol ini ditekan berarti pengukuran kekasaran jalan telah berakhir data hasil pengukuran yang tersimpan di RAM komputer siap untuk ditransfer ke disket.

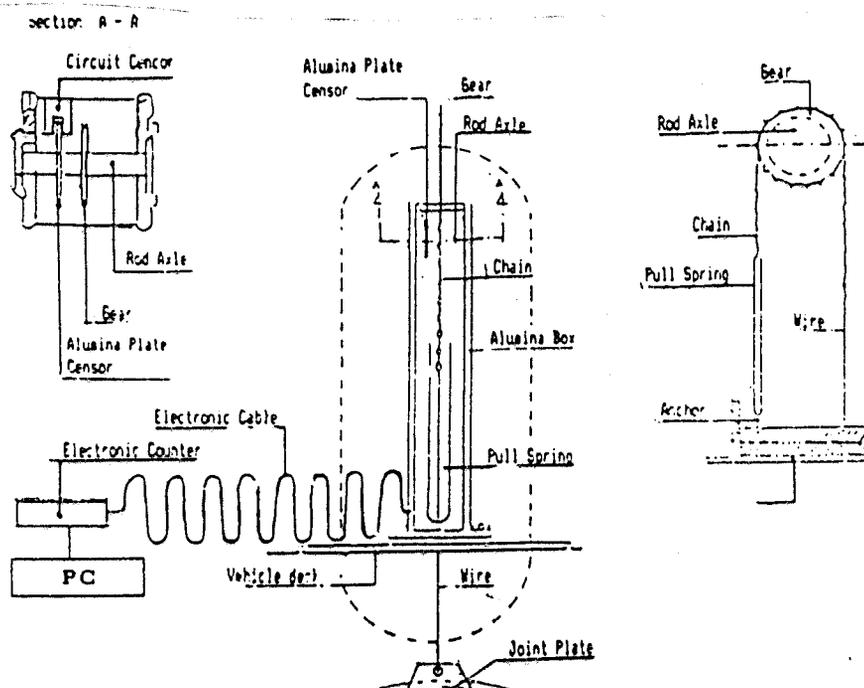


## BAB IV

# PERENCANAAN MEKANIK DAN INSTALASI

### 4.1 PERENCANAAN MEKANIK

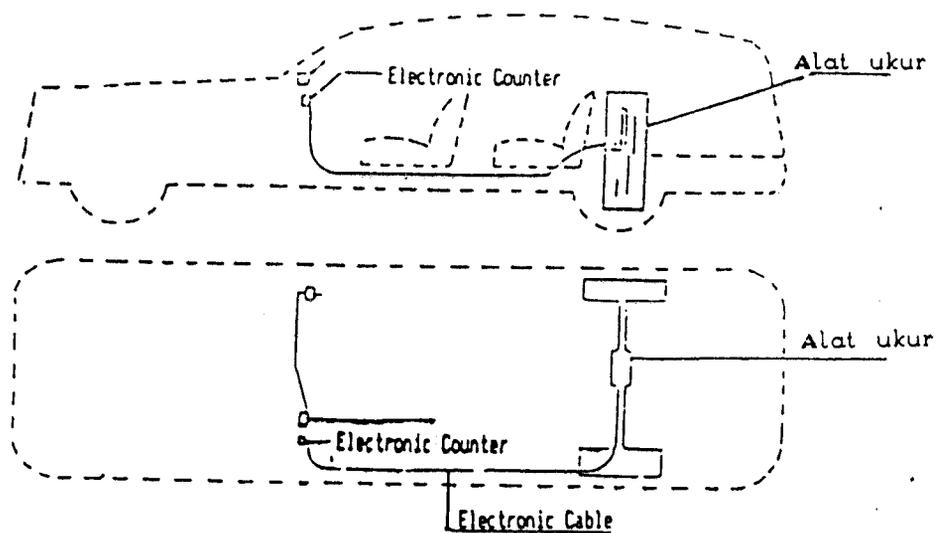
Mekanik untuk pengukuran kekasaran permukaan jalan dibuat sama dengan alat yang sudah ada (NAASRA Roughness Meter) dengan bentuk dan ukuran dapat dilihat pada gambar 4.1. Piringan kisi diletakkan berhimpit dengan gir dan menyatu dengan poros. Pada bagian kiri dan kanan poros diberi bantalan satu arah yang bekerja saling berlawanan untuk memutar kisi searah. Sensor Infra merah diklam pada plat atas untuk mendeteksi kisi-kisi tersebut.



Gambar 4.1 Mekanik Pengukuran Kekasaran Permukaan Jalan

#### 4.2 INSTALASI ALAT PENGUKUR KEKASARAN JALAN

Pada dasarnya pengukuran terhadap kekasaran permukaan jalan melibatkan kendaraan sebagai satu kesatuan sistem. Untuk itu dipakai kendaraan khusus untuk melakukan tugas pengukuran ini. Instalasi pemasangan alat pengukur ini dapat dilihat pada gambar 4.2. Peralatan ukur diletakkan diatas lantai kendaraan bagian belakang. Kawat baja pada alat diklam pada gardan melalui lubang yang dibuat pada lantai kendaraan. Sehingga gerakan naik turunnya gardan menyebabkan terjadinya simpang putaran pada kisi alat. Perubahan jumlah kisi inilah yang nantinya diubah kebesaran listrik dan ditampilkan berupa data kekasaran jalan.



Gambar 4.2 Instalasi Alat Pengukur Kekasaran Jalan

# BAB V

## KALIBRASI DAN PENGUKURAN

Untuk mengetahui hasil dari pembuatan alat ukur ini, telah dilakukan langkah-langkah agar didapatkan hasil yang paling sesuai dan benar mendekati dengan keadaan yang sesungguhnya. Karena itu dilakukan kalibrasi dan uji ukur, yang dibahas pada bab ini.

### 5.1 KALIBRASI

Sebelum peralatan ini digunakan untuk mendapatkan data yang baik maka dilakukan kalibrasi alat terlebih dahulu. Bagian-bagian yang perlu dilakukan kalibrasi adalah:

1. Output Comparator LM324 Dengan mengukur tegangan level 'high' dan level 'low' dari output comparator LM324 pada rangkaian pengukur kekasaran maupun rangkaian odometer dan speedometer. Harus didapatkan level tegangan yang pas antara saat buka dan saat tutupnya.
2. Memantau output counter dari tiap rangkaian counter terutama pada rangkaian speedometer. Dengan mengukur waktu dengan stopwatch untuk mendapatkan kecepatan putar piringan kemudian menyesuaikannya dengan output logic yang diharapkan dari teori. Dimana pada kecepatan 30 km/jam didapatkan logika output

00001111b. Dengan memutar VR-multiturn pada rangkaian astable untuk menyesuaikannya.

## **5.2 PENGUKURAN**

### **5.2.1 Pengukuran pada Rangkaian Odometer**

Pengukuran pada rangkaian odometer dilakukan dengan memutar piringan atau melewati celah sebanyak 10, pada display tercatat nilai satu. Artinya jarak yang ditempuh sepanjang 100 m

### **5.2.2 Pengukuran pada Rangkaian Pengukur Kekasaran**

Pengukuran jarak antar kisi dengan menggunakan jangka sorong adalah 1,1 mm, sehingga secara teoritis satu counter sama dengan 1,1 mm pada simpangan kawat baja (sama dengan simpangan antara lantai dengan gardan), sedangkan untuk 10 counter akan memberikan simpangan sebesar  $1,1 \times 10$  sama dengan 11 mm. Dari hasil pengukuran dengan memberikan counter dari 10 sampai 100 diperoleh tabel seperti dibawah:

Tabel 5.1  
Hasil pengukuran simpangan

n	Counter	Simpangan		Error (mm)
		Pengamatan (mm)	Perhitungan (mm)	
1	10	10	11	1
2	20	23	22	1
3	30	33	33	0
4	40	45	44	1
5	50	56	55	1
6	60	67	55	1
7	70	78	77	1
8	80	90	88	2
9	90	100	99	1
10	100	110	110	0

$\Sigma Error = 9mm$

dimana :

n = banyaknya pengukuran

Pengamatan = pengamatan simpangan kawat baja (mm)

Perhitungan =  $1,1 \times \text{Counter}$  (mm)

Error = (Pengamatan - Perhitungan) (mm)

$$\begin{aligned} \text{Error rata-rata} &= \frac{\Sigma Error}{n} \\ &= \frac{9}{10} \end{aligned}$$

$$= 0,9 \text{ mm}$$

Dari data pada tabel 5.1 diperoleh kesalahan pengukuran untuk 10 kali pengukuran pada alat adalah sebesar 0,9 mm.

**Korelasi antara alat yang dibuat dengan alat NAASRA Roughness Meter.**

Perbandingan antara pengukuran simpangan antara alat yang direncanakan dengan alat yang ada saat ini yaitu NAASRA Roughness meter dapat dilihat pada tabel 5.2. Dengan memberikan simpangan tertentu sampai didapat hasil counter antara 10 sampai 100 akan diperoleh persamaan korelasi dan error dari alat yang direncanakan.

Tabel 5.2

Korelasi antara alat yang dibuat dengan alat NAASRA Roughness Meter

N	Count	Alat(mm)	NAASRA(mm) X	X <sup>2</sup>	X.Y
1	10	10	20	100	200
2	20	23	42	529	966
3	30	33	63	1089	2145
4	40	45	85	2025	3825
5	50	56	105	3136	5880
6	60	67	126	4489	8442
7	70	78	147	6084	11466
8	80	90	170	8100	15300
9	90	100	190	10000	19000
10	100	110	210	12100	23100
		$\Sigma X = 612$	$\Sigma Y = 1160$	$\Sigma x^2 = 47652$	$\Sigma XY = 90324$

Persamaan linier :

$$Y = mX + n$$

Dihitung dengan rumus regresi linier :

$$m = \frac{N \cdot \sum (X \cdot Y) - \sum X \cdot \sum Y}{N \cdot \sum X^2 - (\sum X)^2}$$

$$n = \frac{\sum Y \cdot \sum X^2 - \sum X \cdot \sum (X \cdot Y)}{N \cdot \sum X^2 - (\sum X)^2}$$

diperoleh :

$$m = 1,896$$

$$n = -0,019$$

sehingga didapat persamaan :

$$Y = 1,896 X - 0,019$$

Tabel dibawah ini menunjukkan simpangan pengukuran dengan alat NAASRA Roughness Meter.

Tabel 5.3  
Tabel Simpangan antara alat yang dibuat  
dengan alat NAASRA Roughness Meter

N	$Y_2$	$Y_1$	$Y_2 - Y_1$
1	18,94	20	1,059
2	43,58	42	1,58
3	62,54	63	0,41
4	85,28	85	0,28
5	106,14	105	1,14
6	127,0	126	1,0
7	147,8	147	0,8
8	170,6	170	0,6
9	189,6	190	0,4
10	208,5	210	1,5

$$\begin{aligned} \Sigma (Y_2 - Y_1) \\ = 8,769 \end{aligned}$$

Rata-rata simpangan pengukuran :

$$\frac{\Sigma (Y_2 - Y_1)}{N} = \frac{8,769}{10} = 0,8769$$

**Konversi Nilai Kekasaran (NR) Ke Standard Kekasaran (IRI)**

Dari rumus konversi Nilai Kekasaran (Count/km) ke Standard Kekasaran (IRI mm/km) yang telah disepakati secara Internasional seperti dibawah ini

$$SR(IRI) = 472 + \frac{1473}{b} - \frac{1473 \times a}{b} - \frac{450 \times a \times NR}{b^2} + \frac{225 \times a^2}{b^2} + \frac{225 \times NR}{b^2}$$

data parameter a dan b yang diperoleh dari Bina Marga Dati I Jatim yaitu sebesar a = -43,7 dan b = 118,5, maka dapat dicari korelasi antara alat yang dibuat terhadap standard kekasaran (IRI) dimana NR diperoleh dari persamaan linier diatas, yaitu :

$$NR = Y = 1,896 X - 0,019$$

dimana X adalah data pengukuran dari alat yang dibuat.

#### **Korelasi Terhadap Nilai Indek Kondisi Kekasaran (RCI)**

Telah ditetapkan secara Nasional oleh standard Bina Marga bahwa untuk wilayah Indonesia korelasi antara Standard Kekasaran (IRI) terhadap Indek kondisi Kekasaran (RCI) menggunakan rumus sebagai berikut :

$$RCI = 10EXep(-0,0501.IRI^{1,220920})$$

maka untuk menentukan Indek Kondisi Kekasaran dari data yang diperoleh dari pengukuran terhadap alat yang dibuat pada Tugas Akhir ini dengan memasukkan nilai IRI yang diperoleh dari rumus diatas.

Untuk mendapatkan data yang akurat (memenuhi standard) telah ditetapkan memakai kendaraan khusus untuk survai dengan spesifikasi sebagai berikut:

- Memakai kendaraan khusus (lihat pendahuluan).
- Tekanan ban 27 psi
- Heavy duty shock Absorber
- Ban tidak gundul

# BAB VI

## PENUTUP

### 6.1 KESIMPULAN

Dari pembuatan Tugas Akhir ini didapatkan kesimpulan sebagai berikut :

1. Alat yang direncanakan pada tugas akhir ini dilengkapi dengan komputer, sehingga pendataan dapat dilakukan secara otomatis, sehingga kelalaian pendataan oleh surveyor tidak akan terjadi.
2. Evaluasi terhadap data-data kekasaran hasil survai dapat dilihat dengan cermat dan teliti dan kondisi jalan dapat dilihat secara visual melalui tampilan grafik pada layar monitor.
3. Berdasarkan korelasi terhadap alat pengukur kekasaran permukaan jalan yang telah ada (NAASRA Roughness Meter) dengan alat yang dibuat pada Tugas Akhir ini, maka alat yang direncanakan pada Tugas Akhir ini dapat digunakan sebagai alat pengukur kekasaran permukaan jalan.
4. Pengambilan data kekasaran tiap 1m tempuh dari kendaraan, menyebabkan pengambilan terhadap data kekasaran jalan menjadi lebih teliti.

### 6.2. SARAN-SARAN

Alat pengukur kekasaran permukaan jalan yang dibuat pada tugas akhir ini masih jauh dari sempurna, dibawah ini diberikan beberapa saran-saran atas kekurangan

tersebut dan diharapkan dapat berguna bagi peneliti dan pengembangan yang lebih lanjut.

1. Agar didapat nilai pengukuran yang lebih akurat digunakan kisi dengan celah yang lebih halus dan didukung dengan sensor infra merah yang mempunyai waktu tanggap yang sangat cepat.
2. Poros dibuat pas tanpa goyah.
3. Bantalan cengkeram yang terdapat pada poros harus kuat dan tidak longgar.
4. Batang penyangga dibuat dari bahan yang sangat kuat untuk menghindari gerakan akibat getaran kendaraan saat pengukuran.

Demikian kesimpulan dan saran dari penulis Tugas Akhir yang berjudul **Alat Pengukur Kekasaran Permukaan Jalan (NAASRA Roughness Meter) yang Terhubung Serial ke IBM PC.**

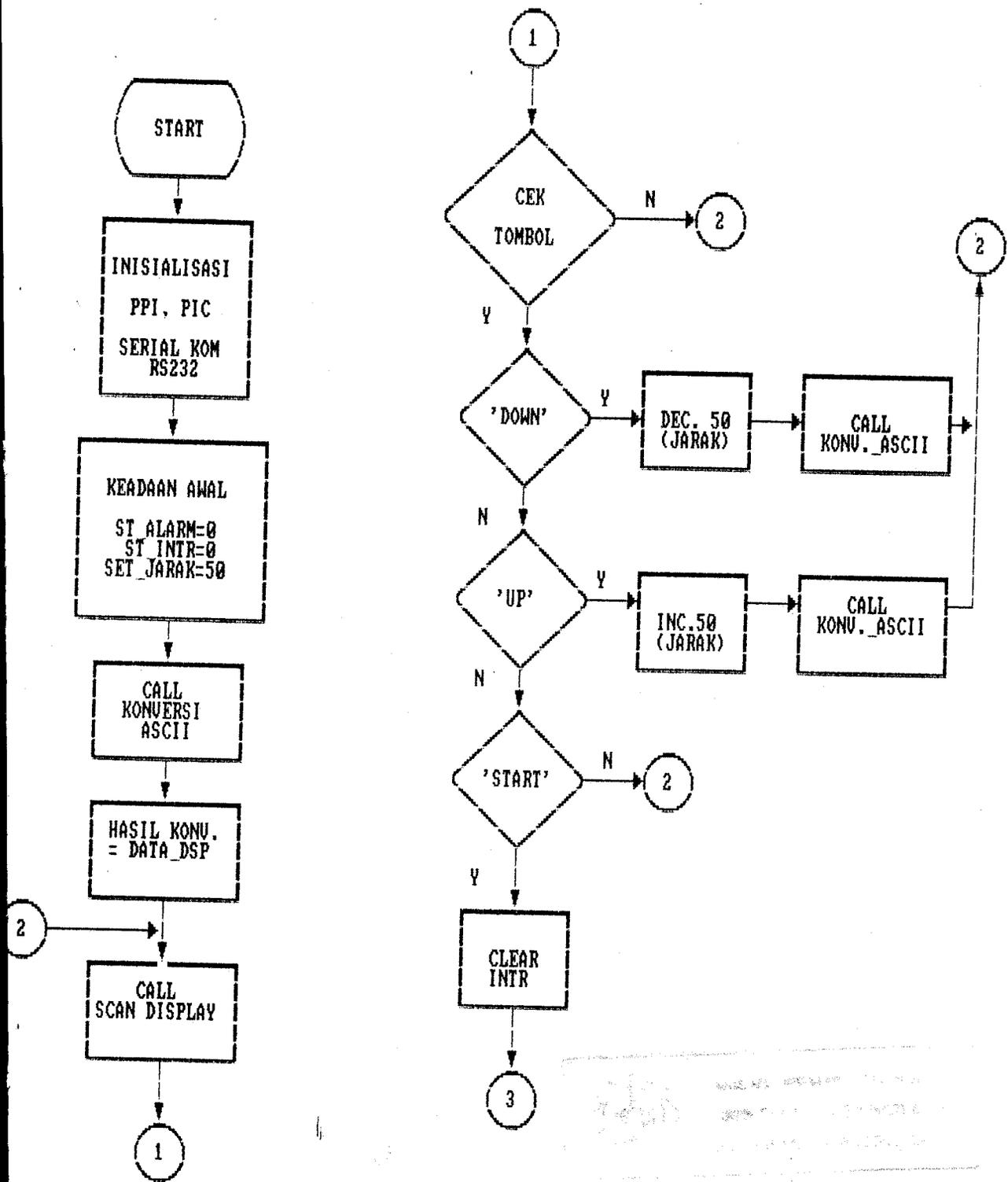
## DAFTAR PUSTAKA

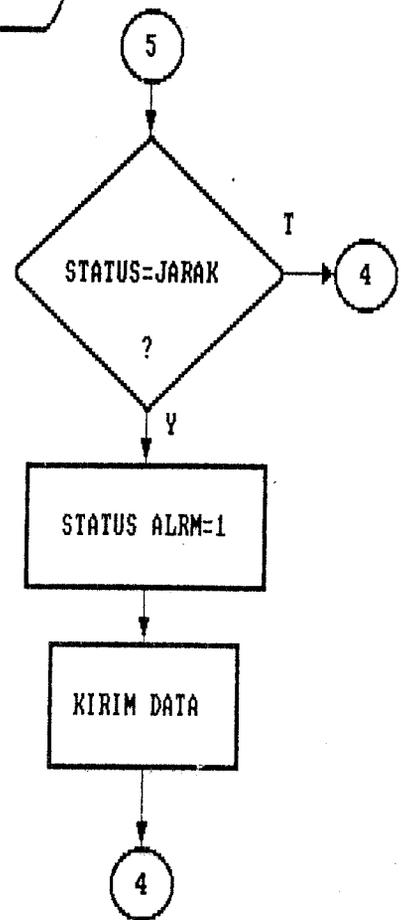
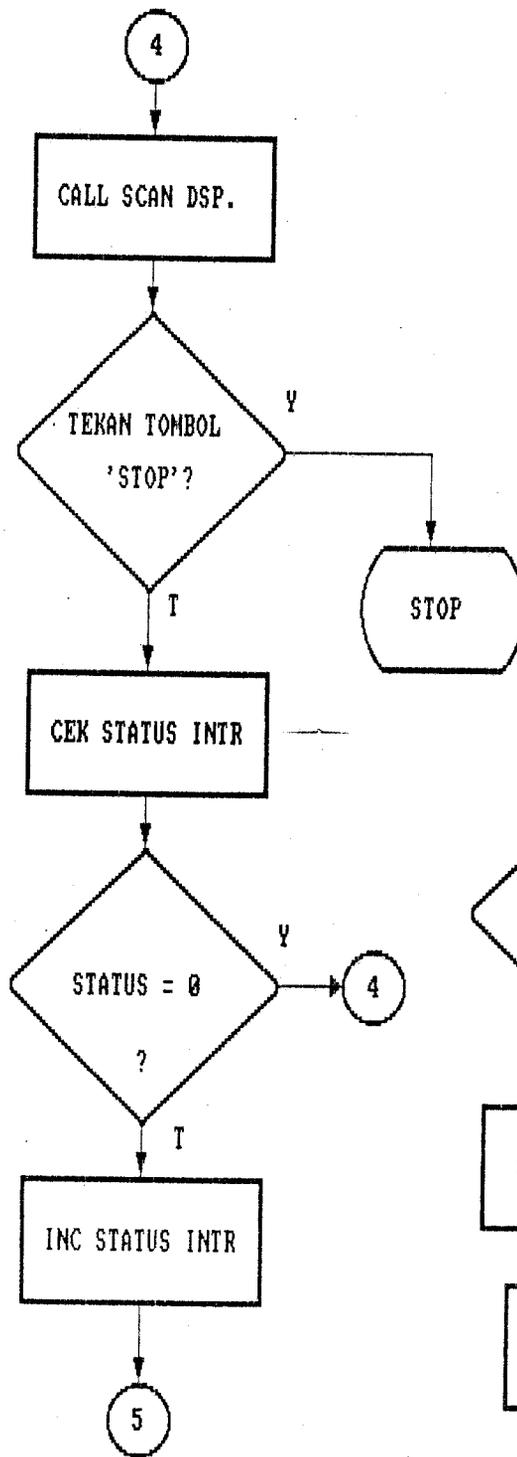
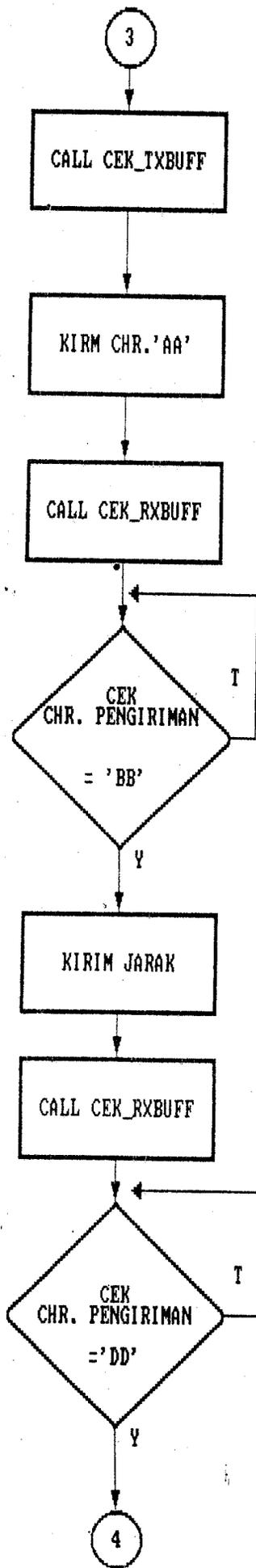
1. Cox. J.B., **MEASUREMENT OF ROAD CONDITION AND THE INTRODUCTION OF NAASRA METER INDONESIA**, DPUP Management Assistant Project for Indonesia, 1983
2. Enggebrechth, Lewis C., **INTERFACING TO IBM PC**, Howard W Sam & Co. Inc. 1985
3. Hall, Douglas V. **MICROPROCESSOR AND INTERFACING : Programming and Hardware**. MacGraw Hill Book, Co. Singapore, 1986
4. Sukirman, Silvia, **PERKERASAN LENTUR JALAN RAYA**, Nova, Bandung, 1992.
5. Uffenbeck, John, **THE 8088/8086 FAMILY : DESIGN, PROGRAMMING AND INTERFACING**, Prentice-Hall Inc., Englewood Cliffs, 1988.
6. -----, **MANUAL PEMELIHARAAN JALAN NO.03/MN/B/1983**, Direktorat Jendral Bina Marga.

7. -----, MICROPROCESSOR AND PERIPHERAL

HANDBOOK Vol. I. Microprocessor, Intel Corporation, 1988.

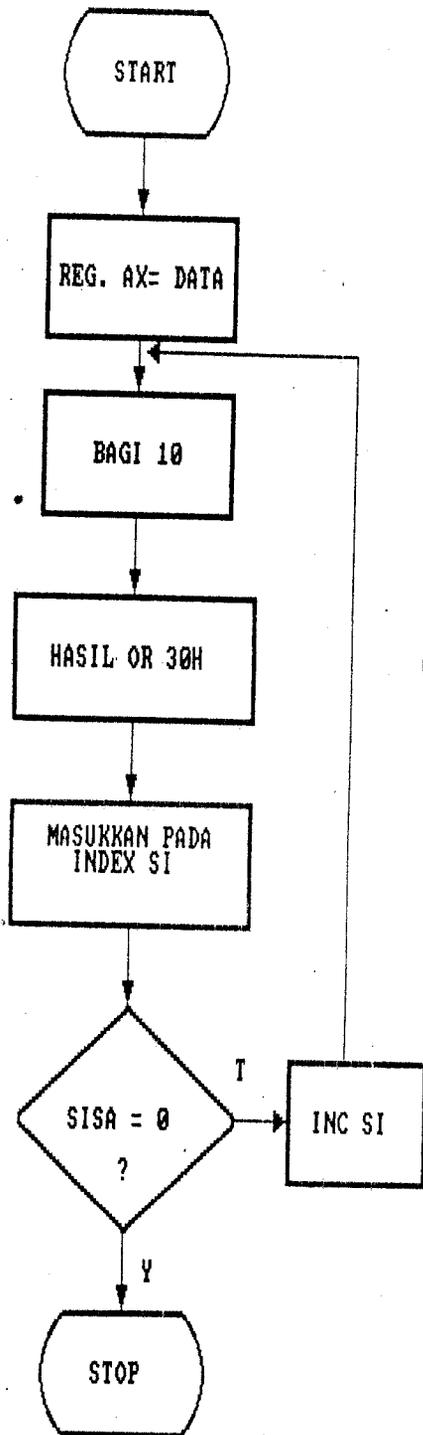
# MAIN PROGRAM



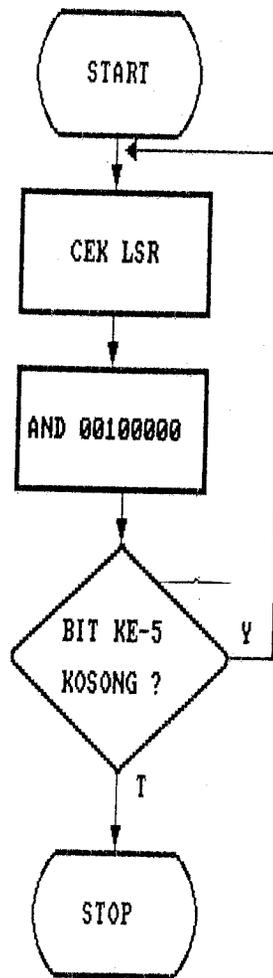


# PROSEDUR PROGRAM

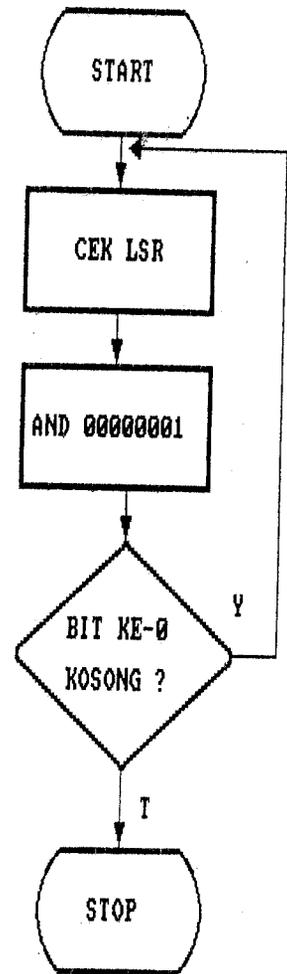
## PROS. KONVERSI\_ASCII

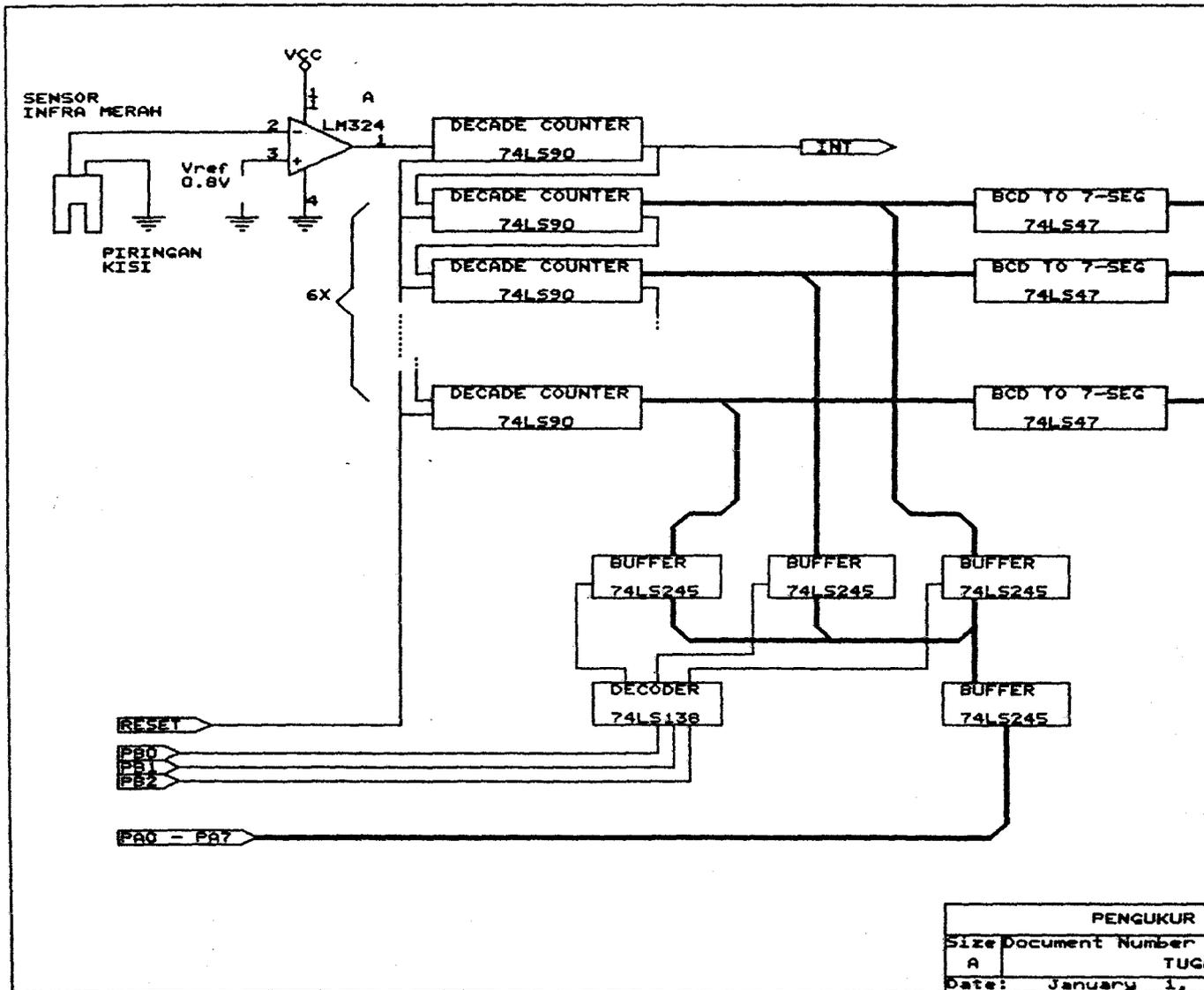


## PROS. TX\_BUFF

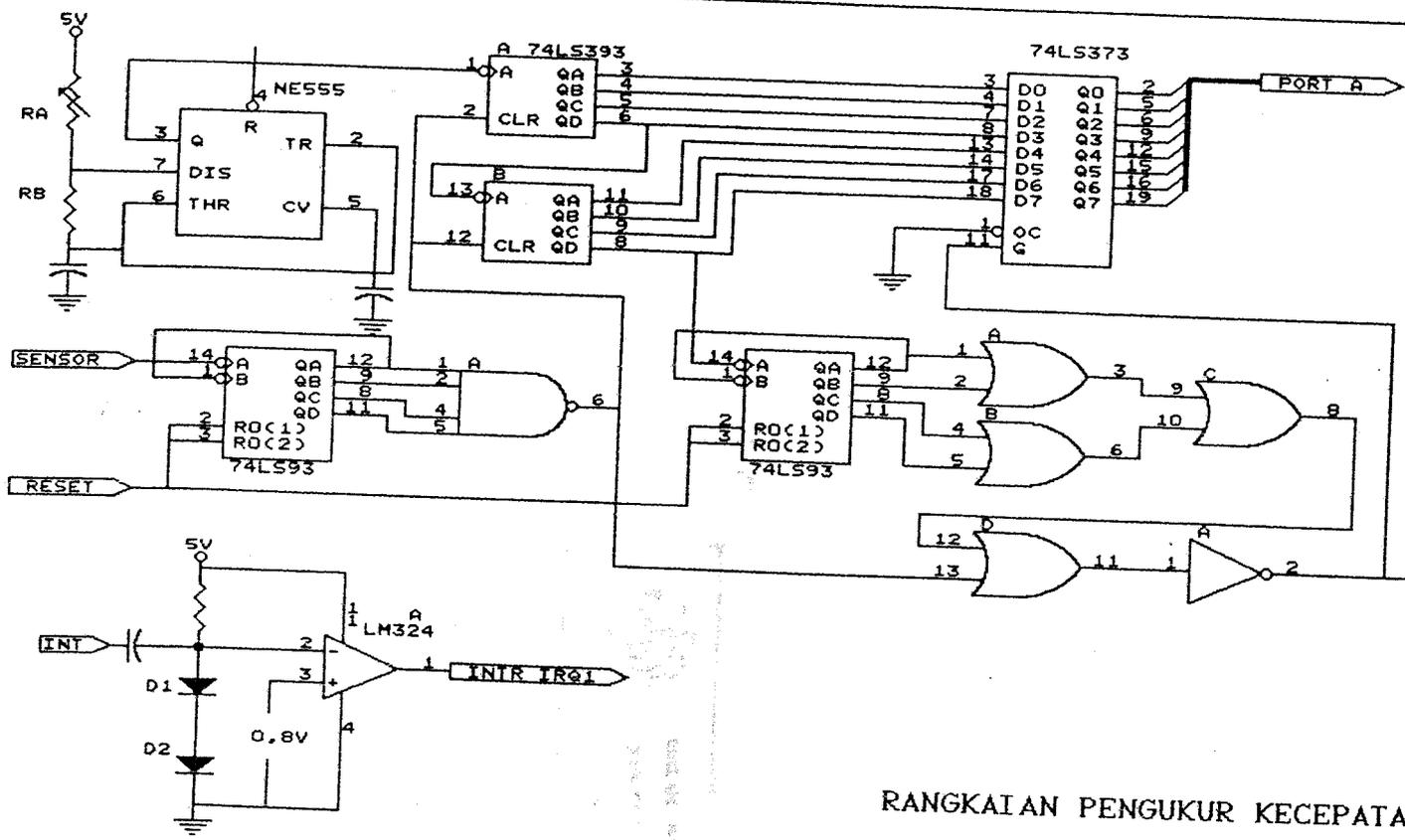


## PROS. RX\_BUFF









RANGKAIAN PENGUKUR KECEPATAN

```
TX      EQU 03F8H
RX      EQU 03F8H
BDL     EQU 03F8H
BDM     EQU 03F9H
IER     EQU 03F9H
IIR     EQU 03FAH
LCR     EQU 03FBH
MCR     EQU 03FCH
LSR     EQU 03FDH
MSR     EQU 03FEH
PA1     EQU 0000H
PB1     EQU 0001H
PC1     EQU 0002H
PW1     EQU 0003H
PA2     EQU 0224H
PB2     EQU 0225H
PC2     EQU 0226H
PW2     EQU 0227H
pica    equ 0c4h
picb    equ 0c5h
icw1    equ 13h
icw2    equ 8h
icw4    equ 1h
ocw1    equ 0e0h
eoi     equ 20h
```

```
org 00h
```

```
jmp start
```

```
data dsp      db 4 dup (0)
old data     dw 0064h
count intr   dw 0
stat alarm   db 0
```

```
:-----:
:inialisasi segment register:
:-----:
```

```
start:
```

```
cli
xor ax,ax
mov ds,ax
mov es,ax
mov ss,ax
mov bp,ax
mov si,ax
mov di,ax
mov ax,07ffh
mov sp,ax
```

```
:-----:
:vektor interrupt PIC
:-----:
```

```
lea bx,IRO0
mov es:[020h],bx
mov es:[022h],cs
```

```
mov al,icw1
mov dx,pica
out dx,al
```

```
mov al,icw2
mov dx,picb
out dx,al
```

```
mov al,icw4
out dx,al
mov al,ocw1
out dx,al
```

```
:-----:
:inisialisasi PPI 8255
:-----:
```

```
mov dx,PW1
mov al,10010001b      :PA:input  PCup:output
out dx,al             :PB:output  PClo:input
```

```
mov dx,PW2
mov al,10011011b     :PA PB PC : input
out dx,al
```

```
:-----:
:inisialisasi serial RS-232
:-----:
```

```
mov dx,LCR
mov al,10000000b
out dx,al
```

```
mov dx,BDM
mov al,0
out dx,al
mov dx,BDL
mov al,0Ch           :baud rate 9600
out dx,al
```

```
mov dx,LCR
mov al,00011011b    :8bit, 1 stop bit,
out dx,al           :parity disable
```

```
mov dx,IER
mov al,00
```

```

    out dx,al

    mov dx,MCR
    mov al,0
    out dx,al

    mov cx,64
zero RX:
    mov dx,RX
    in al,dx
    loop zero Rx

```

```

:-----:
:          PROGRAM UTAMA          :
:-----:

```

```

    mov dx,PC1
    mov al,00010000b           :on alarm
    out dx,al                 :on lamp-set

    push cx
    mov cx,0ffffh
    loop $
    pop cx

    mov dx,PC1
    mov al,01010000b           :off alarm & lamp-stop
    out dx,al                 :on lamp-set

    mov es:stat alarm,0        :awal isi memory
    mov es:count intr,0
    mov es:old data,100

    call konv ASCII
put data:
    lea si,data dsp
    call bouncing

wait pres:
    call scan dsp
    mov dx,PC1
    in al,dx
    and al,00001110b
    cmp al,00000000b
    je wait pres
    cmp al,00000010b
    je @down
    cmp al,00000100b
    je @up
    cmp al,00001000b
    je @set
    jmp wait pres

@down:
    cmp es:old data,100
    jle wait pres

```

```
sub es:old data.100
call konv ASCII
jmp put data
```

@up:

```
cmp es:old data.1000
jge wait pres
add es:old data.100
call konv ASCII
jmp put data
```

@set:

```
mov dx,PC1
mov al,01000000b :off lamp-test
out dx,al
```

```
call cek TxBuff
mov dx,TX
mov al,0AAh :char. kontrol kom.serial out
out dx,al
```

@lookRx:

```
call cek RxBuff
mov dx,RX
in al,dx
cmp al,0BBh :char.kontrol kom.serial in
jne @lookRx
```

```
mov dx,PC1
mov al,00000000b :on alarm
out dx,al
```

```
push cx
mov cx,0ffffh
@lop: loop @lop
pop cx
```

```
mov dx,PC1
mov al,11000000b :off alarm
out dx,al
```

```
:-----:
: kirim info jarak :
:-----:
:
```

```
call cek TxBuff
mov dx,TX
mov ax,es:old data
out dx,al :kirim low byte
xchg al,ah
```

```
push cx
mov cx,0ffffh
delayTX: loop delayTX
pop cx
```

```
call cek TxBuff
```

```
mov dx, TX ; kirim high byte
out dx, al
```

```
-----:
```

```
cek trans:
```

```
call cek RxBuff
mov dx, Rx
in al, dx
cmp al, 0DDh
jne cek trans
```

```
sti
```

```
-----:
```

```
ulang:
```

```
call scan dsp
cmp es:stat alarm.1 ; status alarm -----> ?
je Time alarm
mov dx, PC1
in al, dx
and al, 00000001b ; lihat tombol STOP
jnz stop pros
jmp ulang
```

```
-----:
```

```
time alarm:
```

```
mov dx, PC1
mov al, 0 ; on alarm & lacth data
out dx, al
```

```
push cx
mov cx, 10
```

```
time delay:
```

```
call scan dsp
mov dx, PC1
in al, dx
and al, 00000001b
jnz stop pros
loop time delay
pop cx
```

```
mov dx, PC1
mov al, 11000000b ; off alarm
out dx, al
mov es:stat alarm, 0 ; clear status alarm
jmp ulang
```

```
stop pros:
```

```
Cli
mov dx, PC1
mov al, 00000000b ; lacth data
out dx, al
```

```
call cek TxBuff
mov al, 0CCh
```

```

        mov dx,Tx
        out dx,al

stop:
        mov dx,PC1
        mov al,00110000b           :on alarm & lacth data
        out dx,al

@mat:
        mov cx,0ffffh

        call scan dsp
        loop @mat

        mov dx,PC1
        mov al,01110000b         :off alarm
        out dx,al

@hidup:
        mov cx,10

        push cx
        mov cx,0ffffh

@hid:
        call scan dsp
        loop @hid
        pop cx
        call scan dsp
        loop @hidup

        jmp stop

```

```

:-----:
: SUB-PROGRAM :
:-----:

```

```

:-----:
: anti bouncing :
:-----:

```

```

bouncing proc
wait open:
        call scan dsp
        mov dx,PC1
        in al,dx
        and al,00001110b
        cmp al,00000000b
        jne wait open
        ret
bouncing endp

```

```
:-----:
:scan display
:-----:
```

```
scan dsp proc
    push bx
    push cx
    xor bx,bx
    xor cx,cx
    mov cx,4
```

```
lop1:
```

```
    xor ax,ax
    mov dx,PB1
    mov ah,cl
    sub ah,1
    push cx
    mov cl,4
    rol ah,cl
    and ah,00110000b
    pop cx
    mov al,[bx+si]
    and al,00001111b
    or al,ah
    out dx,al
    inc bx
```

```
    push cx
    mov cx,1000
    loop $
    pop cx
```

```
    loop lop1
```

```
    pop cx
    pop bx
    ret
```

```
scan dsp endp
```

```
:-----:
:konversi desimal ke ASCII
:-----:
```

```
konv ASCII proc
    xor dx,dx
    mov data dsp[0],0
    mov dx,es:old data
    mov si,offset data dsp+3
```

```
konversi:
```

```
    mov ax,dx
    cwd
    mov bx,10
    div bx
    xchg dx,ax
```

```
    or al.30h
    mov byte ptr [si],al
    dec si
    cmp dx.0
    jnz konversi
```

```
    ret
konv ASCII endp
```

```
:-----:
:cek Tx buffer register:
:-----:
```

```
cek TxBuff proc
    push ax
```

```
check thre:
    mov dx,LSR
    in al,dx
    and al.00100000b           :1 --> Tx Buff empty
    jz check thre
```

```
check tsre:
    mov dx,LSR
    in al,dx
    and al.01000000b         :1 --> Tx Shift empty
    jz check tsre
    pop ax
    ret
```

```
cek TxBuff endp
```

```
:-----:
:cek Rx buffer register:
:-----:
```

```
cek RxBuff proc
```

```
@check thre:
    mov dx,LSR
    in al,dx
    and al.00000001b         :1 --> Tx Buff empty
    jz @check thre
    ret
```

```
cek RxBuff endp
```

```
:-----:
:procedure interrupt PIC 8259 IROO:
:-----:
```

```
IROO proc near
```

```
    pushf
    push es
    push ds
    push si
    push di
    push ax
```

```
push bx
push cx
push dx
cli
```

```
add es:count intr.1
mov ax,es:old data
cmp ax,es:count intr
jg @lanjut
mov es:stat alarm.1
mov es:count intr.0      :clear counter intr
```

```
@lanjut:
```

```
mov dx,PA2                :ambil data pada port A
in al,dx
```

```
call cek TxBuff
mov dx,TX
out dx,al
call delay TX
```

```
mov dx,PB2                :ambil data pada port B
in al,dx
```

```
call cek TxBuff
mov dx,TX
out dx,al
call delay TX
```

```
mov dx,PC2                :ambil data pada port C
in al,dx
```

```
call cek TxBuff
mov dx,TX
out dx,al
```

```
mov dx,pica
mov al,eoi
out dx,al
```

```
pop dx
pop cx
pop bx
pop ax
pop di
pop si
pop ds
pop es
popf
sti
iret
```

```
IROO endp
```

```
:-----  
Delay TX proc  
    push cx  
    mov cx.0ffffh  
lop TX:  
    loop lop TX  
    pop cx  
    ret  
Delay TX endp
```

```
:-----  
full:  
    fille db (@reset-$) dup (0ffh)  
    org 3ff0h  
@reset:  
    db 0eah.00h.00h.00h.0fch
```