

**PENGUKURAN KETINGGIAN AIR  
DENGAN ULTRASONIK  
BERBASIS MIKROKONTROLER M68HC11**

**SKRIPSI**

**Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik Elektro  
Program Studi Teknik Elektro**



Oleh :

**BERNADUS GUNAWAN SUDARSONO**

NIM : 945114008

NIRM : 940051123107120008

**PROGRAM STUDI TEKNIK ELEKTRO JURUSAN TEKNIK ELEKTRO  
FAKULTAS TEKNIK  
UNIVERSITAS SANATA DHARMA  
YOGYAKARTA  
1999**

SKRIPSI

PENGUKURAN KETINGGIAN AIR  
DENGAN ULTRASONIK  
BERBASIS MIKROKONTROLER M68HC11

Oleh :

Bernadus Gunawan Sudarsono

NIM : 945114008

NIRM : 9400051123107120008

Telah disetujui oleh :

Pembimbing I

( Ir. Bambang Sutopo , M.Phil )

tanggal : 24 FEB 1999

Pembimbing II

  
( Ir. Linggo Sumarno , MT )

tanggal : 24 FEB 1999

SKRIPSI  
PENGUKURAN KETINGGIAN AIR  
DENGAN ULTRASONIK  
BERBASIS MIKROKONTROLER M68HC11

Dipersiapkan dan ditulis oleh

Bernadus Gunawan Sudarsono

NIM : 945114008

NIRM : 940051123107120008

Telah dipertahankan di depan Panitia Penguji  
pada tanggal 8 Januari 1999  
dan dinyatakan memenuhi syarat.

Susunan Panitia Penguji

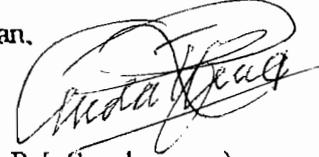
	Nama Lengkap	Tanda Tangan
Ketua	Ir. Bambang Sutopo, M.Phil	
Sekretaris	Ir. Linggo Sumarno, MT	
Anggota	Dr. Ir. Thomas Sriwidodo	
Anggota	Ir. Suryono, M.Phil	

Yogyakarta, 26 FEB 1999

Fakultas Teknik  
Universitas Sanata Dharma

Dekan,



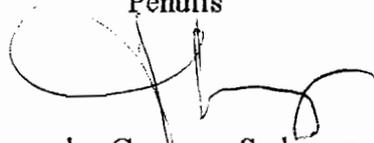
  
( Ir. P.J. Soedarjana )

## PERNYATAAN KEASLIAN KARYA

Saya menyatakan dengan sesungguhnya bahwa skripsi yang saya tulis ini tidak memuat karya atau bagian karya orang lain, kecuali yang telah saya sebutkan dalam kutipan dan daftar pustaka, sebagaimana layaknya karya ilmiah.

Yogyakarta, Desember 1998

Penulis

A handwritten signature in black ink, consisting of several loops and a long horizontal stroke, positioned below the word 'Penulis'.

( Bernadus Gunawan Sudarsono )

## ABSTRAK

Skripsi dengan judul “ Pengukuran Ketinggian Air dengan Ultrasonik Berbasis Mikrokontroler M68HC11 “ ini, membahas tentang cara mengukur tinggi air dalam bak dengan ketinggian 37 cm. Tinggi air dihitung dari dasar bak sampai permukaan air dalam bak. Transduser ultrasonik digunakan untuk memancarkan dan menerima pulsa. Transduser ini dipasang di atas bak air dengan jarak 1 cm dari bibir permukaan bak. Untuk selanjutnya tinggi bak dianggap 38 cm, yang merupakan penambahan dari 37 cm tinggi bak dengan 1 cm jarak ultrasonik dengan bibir permukaan bak.

Pulsa dipancarkan ke bawah oleh transduser ultrasonik sebagai pemancar dan dipantulkan oleh permukaan air, sehingga pantulan ini akhirnya diterima oleh transduser ultrasonik sebagai penerima. Waktu yang dibutuhkan untuk perjalanan pulsa ini , disebut dengan tunda waktu. Tunda waktu ini diproses di dalam mikrokontroler sehingga menjadi bilangan yang menandakan tinggi air dalam bak. Bilangan ini akhirnya ditampilkan dengan penampil tujuh segmen.

Dalam skripsi ini, judul tersebut diaplikasikan dalam bentuk alat ukur miniatur yang diletakkan di atas bak. Alat ukur tersebut terdiri dari perangkat keras dan perangkat lunak. Tinggi air dalam alat ukur ini dapat dilihat dalam tampilannya yang menggunakan penampil tujuh segmen. Alat ukur ini hanya digunakan untuk tinggi bak 37 cm. Jika kurang atau melebihi 37 cm ini, alat ukur ini akan menghasilkan hasil yang keliru.

## KATA PENGANTAR

Puji syukur kepada Tuhan Yang Maha Kuasa, karena telah menyertai, membantu dan memberi restu, sehingga tugas akhir dan skripsi ini dapat selesai dengan baik.

Memang benar, kesulitan merupakan tantangan dalam tugas akhir ini. Namun setelah kesulitan tersebut teratasi, muncul suatu kepuasan batin. Selain itu menambah kemampuan akademis. Hal-hal baru dalam tugas akhir ini memang banyak sekali, yang seringkali menimbulkan kesulitan baru. Namun kesulitan ini, bukan dihindari tetapi harus dihadapi dengan tekun dan sabar, karena dengan ketekunan dan kesabaranlah kesulitan tersebut akan teratasi. Terima kasih Tuhan, karena telah memberi semangat ketekunan dan kesabaran dalam mengerjakan Tugas Akhir dan Skripsi ini.

Skripsi ini dibuat sebagai syarat untuk lulus strata I jurusan Teknik Elektro, fakultas Teknik, Universitas Sanata Dharma. Tugas Akhir dan Skripsi ini, tentunya tidak akan berhasil tanpa bantuan pihak lain. Terima kasih kuucapkan bagi mereka yang telah membantu dalam Tugas Akhir ini, khususnya :

1. Ir. Bambang Sutopo, M.Phil selaku Ketua Jurusan dan Pembimbing I
2. Ir. Linggo Sumarno, MT selaku pembimbing II
3. B. Djoko Suwarno, MT yang telah banyak memberi saran dan masukan.
4. Dekan Fakultas Teknik
5. Laboran-laboran Teknik Elektro

6. Ayahku Almarhum Marito
7. Ibuku Endang Samsinah dan adik-adikku Beny dan Santi
8. Pihak manapun yang telah membantu dalam tugas akhir dan skripsi ini.

Tentunya masih banyak kekurangan dalam skripsi ini. Kritik dan saran untuk kesempurnaan skripsi ini sangat diharapkan.

Yogyakarta, Desember 1998

Penyusun

Bernadus Gunawan Sudarsono

## DAFTAR ISI

	Halaman
HALAMAN JUDUL.....	i
HALAMAN PERSETUJUAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN.....	iii
PERNYATAAN KEASLIAN KARYA.....	iv
ABSTRAK.....	v
KATA PENGANTAR.....	vi
DAFTAR ISI.....	viii
DAFTAR TABEL.....	x
DAFTAR GAMBAR.....	xi
BAB I PENDAHULUAN.....	1
I.1 LATAR BELAKANG.....	2
I.2 PERUMUSAN MASALAH.....	2
I.3 PEMBATASAN MASALAH.....	3
I.4 MANFAAT PENELITIAN.....	3
I.5 SISTEMATIKA PENULISAN.....	4
BAB II DASAR TEORI.....	5
II.1 ULTRASONIK.....	11
II.1.1 BUNYL.....	11
II.2 MIKROKONTROLER M68HC11.....	18
II.2.1. STRUKTUR MCU M68HC11.....	19
II.2.2. MODEL UNTUK PEMROGRAM.....	20
II.2.3. PETA MEMORI DAN I/O.....	21
II.2.4. MODE OPERASI.....	22
II.2.5. UNIT PENGOLAH UTAMA.....	22



II.2.6. RESET DAN INTERUPSI.....	29
II.2.7. PARALEL INPUT / OUTPUT.....	32
II.2.8. SISTEM PEWAKTUAN.....	35
II.3. PPI 8255.....	40
BAB III PERANGKAT KERAS.....	47
III.1. MODE EKSPANDED M68HC11.....	48
III.2. PEMANCAR DENGAN TIMER 17555.....	48
III.3. RANGKAIAN PENGONDISI ISYARAT.....	56
III.3.1. PENGUAT.....	58
III.3.2. PENYEERAH.....	65
III.3.3. PENYARING LALU RENDAH.....	66
III.3.4. BLOKING KAPASITOR.....	68
III.3.5 PEMBANDING, PENYEERAH KELUARAN PEMBANDING, DAN PEMBATAS TEGANGAN.....	72
III.4. PENAMPIL.....	73
BAB IV PERANGKAT LUNAK.....	77
BAB V DATA DAN PEMBAHASAN.....	104
BAB VI KESIMPULAN DAN SARAN.....	119
DAFTAR PUSTAKA.....	121
LAMPIRAN A.....	A-1
LAMPIRAN B.....	B 1
LAMPIRAN C.....	C-1

## DAFTAR TABEL

	Halaman
1.1. VEKTOR ALAMAT RESET.....	26
1.2. VEKTOR ALAMAT RESET DAN PENYEBAB RESET.....	30
1.3. URUTAN PENUMPUKAN ISI REGISTER SAAT INTERUPSI.....	32
1.4. PORT <u>INPUT</u> / <u>OUTPUT</u> .....	32
1.5. OPERASI CS, RD, WR, A0 DAN A1.....	42
2.1. HASIL PENGAMATAN DAN TEORITIS	
TUNDA WAKTU.....	105
2.2. DATA PERHITUNGAN REGRESI .....	111
2.3. DATA LINEARISASI DARI GRAFIK YANG ADA DI GAMBAR D.1.....	113

## DAFTAR GAMBAR

	Halaman
A.1. ILUSTRASI TUNDA WAKTU.....	1
A.2. PENEMPATAN TRANSDUSER ULTRASONIK DAN KOMPONEN YANG DIUKUR.....	7
A.3. BLOK DIAGRAM PENGUKURAN KETINGGIAN AIR SEDERHANA.....	9
A.4. GELOMBANG TRANSVERSAL DAN GELOMBANG LONGITUDINAL.....	11
A.5. GELOMBANG LONGITUDINAL SECARA GRAFIS.....	13
A.6. KONSTRUKSI DASAR ULTRASONIK SEBAGAI PENGIRIM.....	16
A.7. GEMA ULTRASONIK UNTUK PENGUKURAN LEVEL AIR.....	18
A.8. BLOK DIAGRAM M68HC11.....	19
A.9. REGISTER CPU M68HC11.....	20
A.10. PETA MEMORI DAN I/O M68HC11.....	21
A.11. DEMULTIPLEKS ALAMAT/ DATA.....	23
A.12. BLOK DIAGRAM CAPTURE / COMPARE.....	38
A.13. DIAGRAM BLOK PPI 8255.....	41
A.14. FORMAT KATA KENDALI.....	45
B.1. DIAGRAM BLOK PENGUKURAN KETINGGIAN AIR.....	47
B.2. MODE EKSPANDED M68HC11.....	49
B.3. IC TIMER 17555 DENGAN KOMPONEN TAMBAHANNYA.....	50
B.4. KELUARAN GELOMBANG KOTAK DAN SINYAL AKIBAT KAPASITOR.....	50
B.5. MENCARI TEGANGAN DI TITIK A DAN B.....	51
B.6. UNJUK KERJA PA6 PADA PIN RESER TIMER 17555.....	56

B.7.	RANGKAIAN PENGONDISI ISYARAT SECARA KESELURUHAN.....	57
B.8.	GELOMBANG KELUARAN TRANSDUSER ULTRASONIK SEBAGAI PENERIMA.....	57
B.9.	PENGUAT DENGAN UMPAN BALIK ARUS.....	58
B.10.	PENGUAT DENGAN ANALIS DC.....	59
B.11.	RANGKAIAN PENGUAT TANPA UMPAN BALIK.....	61
B.12.	MODEL TRANSISTOR UNTUK ANALISA AC.....	62
B.13.	MODEL RANGKAIAN UNTUK PENGUATAN PERTAMA ( AV1 ).....	62
B.14.	MODEL RANGKAIAN PENGUAT KEDUA ( AV2 ).....	63
B.15.	DIODA SEBAGAI PENYEARAH.....	66
B.16.	PENYARING LALU RENDAH.....	67
B.17.	TANGGAPAN FREKUENSI PENYARING LALU RENDAH 16 KHZ.....	67
B.18.	KAPASITOR DIPASANG PARALEL DENGAN PEMBANDING.....	69
B.19.	PERUBAHAN BENTUK SINYAL AKIBAT KAPASITOR LIAR DI OSILOSKOP.....	70
B.20.	BLOK DIAGRAM SISTEM PENYIMPANG VERTIKAL OSILOSKOP.....	70
B.21.	RANGKAIAN PENYUSUT ( ATENUATOR ) DENGAN KAPASITOR LIARNYA.....	70
B.22.	RANGKAIAN PEMBANDING, PENYEARAH DAN PEMBATAS TEGANGAN.....	73
B.23.	RANGKAIAN PENAMPIL SECARA KESELURUHAN.....	74
B.24.	RANGKAIAN TANDA KOMA.....	75
C.1.	DIAGRAM ALIR UTAMA.....	78
C.2.	DIAGRAM ALIR LAYANAN INTERUPSI INPUT CAPTURE 1.....	79

C.3.	REGISTER CCR.....	80
C.4.	DIAGRAM ALIR DELAY1 DAN DELAY2.....	81
C.5.	DIAGRAM ALIR BLOK 3.....	85
C.6.	DIAGRAM ALIR BLOK 4.....	86
C.7.	DIAGRAM ALIR DELAY.....	87
C.8.	DIAGRAM ALIR BLOK 5.....	90
C.9.	DIAGRAM ALIR BLOK 6.....	92
C.10.	DIAGRAM ALIR PERUBAHAN BINER MENJADI BCD.....	94
C.11.	CONTOH PENGGESERAN 5 BIT KE KIRI.....	95
C.12.	DIAGRAM ALIR BLOK 7.....	99
D.1	GRAFIK TINGGI AIR DALAM BAK DENGAN HASIL PENAMPIL.....	108
D.2	GRAFIK REGRESI ANTARA TINGGI AIR DALAM BAK DAN HASIL PENAMPIL.....	114

# BAB I

## PENDAHULUAN

### I.1. Latar Belakang

Kemajuan teknologi saat ini berkembang dengan pesat, sehingga masyarakat dunia merasa dimanja dalam berbagai kemudahan dalam kehidupan sehari-harinya. Teknologi elektronika seakan-akan sudah mendarah daging dalam kebutuhan masyarakat. Aplikasi teknologi elektronika sungguh dirasakan dalam kehidupan masyarakat, contoh dalam kehidupan industri.

Teknologi elektronika memegang peran penting dalam industri karena teknologi ini mempunyai banyak keunggulan diantaranya dalam kecepatan proses, kepastian kerja, dan kemampuannya sebagai pengendali alat ukur secara otomatis. Namun sayang, pesatnya kemajuan teknologi elektronika di industri tidak sepenuhnya bisa diikuti oleh dunia pendidikan, padahal industri membutuhkan hasil dunia pendidikan untuk mengembangkan industri. Hal ini terjadi diantaranya karena kurangnya dana untuk penelitian. Selain itu kurangnya minat untuk melakukan penelitian karena memang memakan waktu yang cukup lama dan sudah terbiasa sebagai pengguna yang mungkin memerlukan biaya yang lebih sedikit daripada melakukan penelitian sendiri. Masih ada sebab lain yaitu kurangnya buku-buku yang menunjang untuk penelitian.

Banyak sekali macam aplikasi teknologi elektronika di industri di antaranya adalah bidang pengukuran. Bidang pengukuran yang diminati industri adalah

pengukuran yang tidak memerlukan pemikiran lebih lanjut dalam penggunaannya. Bidang pengukuran dalam tugas akhir ini adalah pengukuran ketinggian air dengan menggunakan transduser ultrasonik dan dikendalikan mikrokontroler M68HC11.

## **I.2. Perumusan Masalah**

Masalah yang dibahas dalam tugas akhir ini adalah mengukur ketinggian air dalam suatu bak dengan ketinggian tertentu. Tinggi air dalam bak yang diukur adalah dari dasar bak sampai permukaan air. Transduser yang digunakan untuk mendeteksi perubahan ketinggian air adalah transduser ultrasonik. Transduser ini terdiri atas pemancar dan penerima. Untuk mengetahui besarnya ketinggian air dan perubahannya dibuat suatu penampil yang memunculkan angka-angka yang menampilkan hasil pengukuran ini.

Proses pendeteksian dan penampilan ini dilakukan dengan pengendalian secara otomatis. Pengendalian ini dilakukan oleh mikrokontroler M68HC11. Hasil akhir dari tugas akhir ini berupa suatu alat miniatur yang dapat mengukur ketinggian air dengan tinggi bak tertentu. Alat miniatur ini diletakkan di atas bak.

## **I.3. Pembatasan Masalah**

Pengukuran ketinggian air ini memang tidak sempurna, karena memang alat ini mempunyai pembatasan masalah di luar perancangan secara elektronis.

Contohnya : Suhu. Suhu dalam tugas akhir ini cukup berperan dalam menentukan hasil akhir. Tetapi dalam tugas akhir ini suhu dibatasi dengan nilai tertentu yang konstan. Selain suhu, hal yang perlu dibatasi adalah daerah pengukuran yaitu dengan menentukan tinggi bak dengan nilai tertentu, di luar tinggi bak tersebut, hasil pengukuran tidak sesuai yang diinginkan.

Hal lain yang perlu diatasi adalah kondisi air yaitu air yang diukur harus tenang dalam arti tidak berombak dan bergelembung, karena akan mempengaruhi hasil pengukuran. Hal ini terjadi karena berhubungan dengan kemampuan ultrasonik sebagai transduser.

#### **I.4. Tujuan Penelitian**

Tugas Akhir ini mempunyai tujuan yaitu mencoba mengaplikasikan suatu ide dan dituangkan dalam suatu perancangan awal yang berupa diagram blok dan perancangan akhir dalam bentuk rangkaian dan program. Perancangan akhir ini akhirnya dibuat suatu bentuk yang bisa dipakai dan dinikmati walaupun dalam tahap sederhana.

#### **I.5. Manfaat Penelitian**

Manfaat penelitian ini besar sekali, khususnya dalam hal kemandirian untuk berpikir, bertindak dan memutuskan sesuatu. Selain itu menambah kemampuan akademis di antaranya :

- a. Mengetahui sensor ultrasonik

- b. Mengetahui Mikrokontroller M68HC11
- c. Mengetahui PPI 8255
- d. Dapat membuat alat ukur ketinggian air sendiri

#### **L6. Sistematika Penulisan**

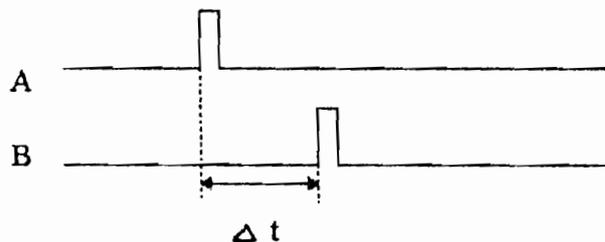
Dalam penulisan ini mempunyai sistematika sebagai berikut :

1. **BAB I    PENDAHULUAN**
2. **BAB II    DASAR TEORI**
3. **BAB III    PERANGKAT KERAS**
4. **BAB IV    PERANGKAT LUNAK**
5. **BAB V    DATA DAN PEMBAHASAN**
6. **BAB VI    KESIMPULAN DAN SARAN**

## BAB II DASAR TEORI

Tugas akhir dengan judul "Pengukuran Ketinggian Air dengan Ultrasonik Berbasis Mikrokontroler M68HC11 " ini, bertujuan mengukur ketinggian air dalam bak yang mempunyai ketinggian 38 cm. Tranduser yang digunakan untuk mengukur ketinggian air ini adalah tranduser ultrasonik. Tranduser ultrasonik adalah tranduser yang memancarkan dan menerima bunyi dengan frekuensi di atas 20 Khz. Tranduser ini diletakkan di atas permukaan bak dengan arah menghadap ke bawah, agar bunyi yang dikeluarkan mengarah ke permukaan air yang ada di dalam bak.

Sebenarnya fungsi tranduser ultrasonik adalah memancarkan pulsa dan menerima pulsa kembali setelah pemantulan pulsa oleh air. Waktu yang dibutuhkan saat permulaan pulsa yang dipancarkan oleh tranduser ultrasonik sebagai pemancar sampai diterimanya pulsa kembali setelah pemantulan di permukaan air oleh tranduser ultrasonik sebagai penerima disebut dengan tunda waktu. Ilustrasi mengenai tunda waktu ini dapat dilihat pada gambar di bawah ini :



Gambar A.1 ilustrasi tunda waktu

Keterangan : A = pulsa yang keluar dari tranduser ultrasonik sebagai pemancar untuk mengaktifkan pencacah tunda waktu.

B = pulsa yang diterima oleh transduser ultrasonik sebagai penerima untuk menghentikan pencacah tunda waktu.

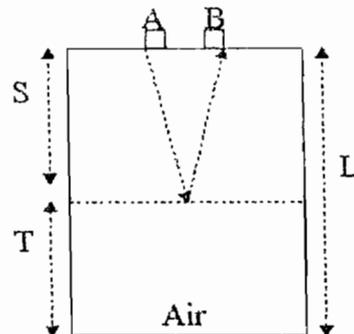
$\Delta t$  = tunda waktu

Dalam kenyataannya, tunda waktu diukur dari keluaran pin PA6 Port A M68HC11 sampai masukan pin Input Capture 1 M68HC11. Keluaran transduser ultrasonik sebagai penerima berupa gelombang AM termodulasi. Hal ini dapat dilihat pada bab III.

Alat yang digunakan untuk mengukur tunda waktu adalah mikrokontroler M68HC11. Tunda waktu ini merupakan hal yang sangat penting sebagai penentu dalam mengukur jarak, yaitu dari transduser ultrasonik sampai dengan permukaan air. Dalam prakteknya nanti, tunda waktu ini dibagi dengan dua karena waktu sesungguhnya adalah waktu yang digunakan pulsa untuk berjalan dari transduser ultrasonik sampai permukaan air atau dari permukaan air sampai transduser ultrasonik. Persamaan yang digunakan untuk mengukur jarak dari transduser ultrasonik sampai permukaan air adalah  $S = V \cdot t$ . S adalah jarak antara transduser ultrasonik dengan permukaan air yang satuannya yaitu meter. V adalah kecepatan bunyi di udara. Kecepatan bunyi ini besarnya tergantung dari suhu sekitaran dan frekuensi. Persamaan dari kecepatan bunyi yang tergantung dari suhu adalah  $V = 331 + (0,6 \cdot T)$  meter/detik. 331 adalah kecepatan bunyi di udara pada suhu  $0^\circ$  Celcius dan T adalah suhu sekitaran ( $^\circ$ Celcius). Kecepatan bunyi yang tergantung frekuensi, mengikuti persamaan  $V = \lambda \cdot f$ ,  $\lambda$  adalah panjang gelombang dan f adalah frekuensi. Namun dalam tugas akhir ini nilai kecepatan tergantung oleh suhu, tanpa memperhatikan frekuensi karena setelah

diadakan percobaan pengaruh frekuensi terhadap hasil ukur ketinggian air, sangatlah kecil.  $t$  adalah waktu sesungguhnya, dapat diperoleh dengan membagi dua nilai tunda waktu yang didapat ( $t = \Delta t / 2$  detik).

Tinggi air dalam bak diukur dari dasar bak sampai permukaan air dalam bak. Ilustrasi mengenai tinggi air dalam bak dan penempatan transduser ultrasonik dapat dilihat pada gambar A.2 di bawah ini.



**Gambar A.2 Penempatan transduser ultrasonik dan komponen yang diukur**

Keterangan gambar : A = transduser ultrasonik sebagai pemancar  
 B = transduser ultrasonik sebagai penerima  
 S = jarak ultrasonik dengan permukaan air  
 T = tinggi air dalam bak  
 L = tinggi bak air

Untuk memperoleh tinggi air dalam bak ( $T$ ) ini, tinggi bak air ( $L$ ) harus dikurangkan dengan jarak transduser dengan permukaan air ( $S$ ), sehingga dapat diperoleh persamaan  $T = L - S$ .

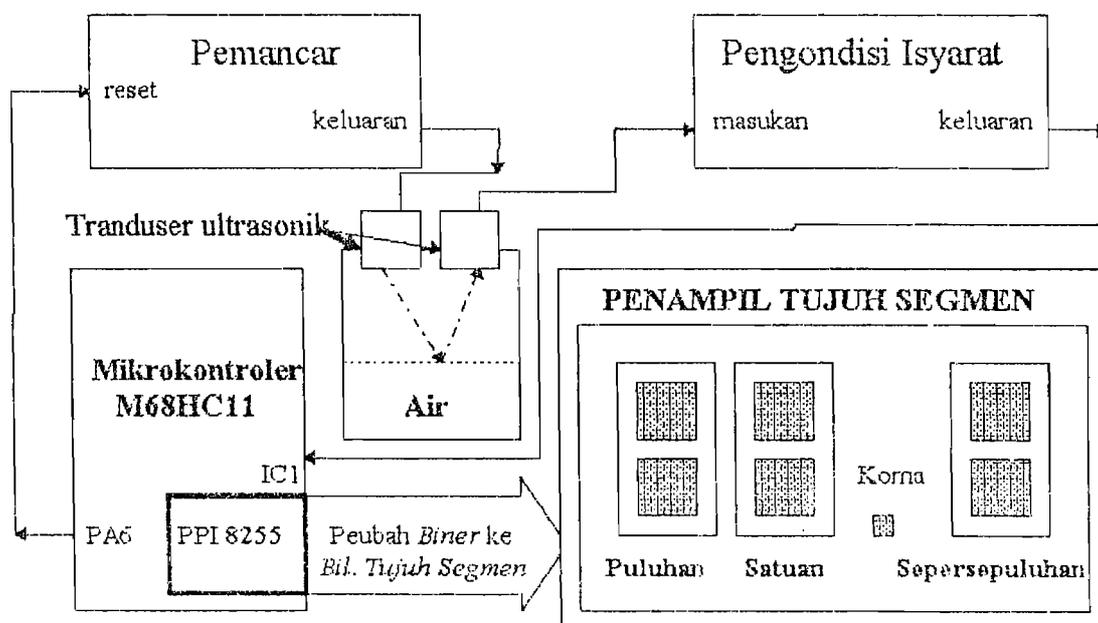
Proses yang menggunakan persamaan-persamaan di atas, dalam tugas akhir ini dilakukan oleh mikrokontroler M68HC11. Jadi tugas mikrokontroler bukan hanya

mengukur tunda waktu saja. Tunda waktu diukur dengan proses yang cukup panjang. Proses tersebut diawali dengan pembacaan pinggirannya naik dari pulsa yang dipancarkan oleh pin PA6 dari Port A M68HC11. Pin PA6 ini dihubungkan dengan pin reset dari timer 17555. Timer 17555 digunakan untuk memancarkan gelombang kotak dengan frekuensi 40 Khz. Sinyal keluaran timer 17555 ini dikendalikan oleh pin PA6 Port A M68HC11. Jika pin PA6 dalam aras tinggi, timer 17555 akan memancarkan pulsa dengan periode sebesar lebar pulsa PA6. Jika pin PA6 dalam keadaan aras rendah, timer 17555 tidak memancar ( dalam keadaan mati ). Fungsi PA6 dapat diringkas sebagai berikut :

- a) Sebagai port output yang membangkitkan pulsa, yang digunakan sebagai tanda awal pengukuran tunda waktu. Dalam tugas akhir ini, tanda awal dibaca dari pinggirannya naik pulsa keluaran pin PA6. Pembacaan pinggirannya naik ini digunakan untuk mengaktifkan pencacah yang ada di mikrokontroler M68HC11.
- b) Mengendalikan keluaran timer 17555 yang terhubung ke transduser ultrasonik sebagai pemancar. Pin PA6 tidak langsung dihubungkan ke transduser ultrasonik sebagai pemancar, tetapi harus melalui timer 17555. Hal ini dilakukan karena timer 17555 mempunyai daya pancar yang lebih besar daripada keluaran pin PA6.
- c) Memberikan pengaturan waktu antar pengukuran. Hal ini dilakukan dengan memberikan keluaran aras rendah dengan waktu yang sesuai keinginan.

Transduser ultrasonik sebagai pemancar akan memancarkan pulsa yang keluar dari timer 17555. Pulsa ini dipancarkan sampai ke permukaan air dan dipantulkan oleh

permukaan air sampai diterima oleh transduser ultrasonik sebagai penerima. Sinyal yang diterima oleh transduser ultrasonik sebagai penerima bukan lagi berupa pulsa tetapi berupa gelombang AM termodulasi yang amplitudonya sangat kecil. Gelombang AM ini perlu dikuatkan dan diubah menjadi pulsa kembali. Perubahan menjadi pulsa ini dilakukan agar mikrokontroler dapat menghentikan pencacahan tunda waktu pada pinggiran naik dari gelombang kotak yang dikenakan padanya. Dengan terhentinya pencacahan ini dapat diukur tunda waktunya. Perubahan dan penguatan gelombang AM termodulasi menjadi gelombang kotak dilakukan dalam diagram blok di bawah ini, yaitu pada kotak pengondisi isyarat. Kotak pemancar dari diagram blok ini berisi unjuk kerja dari timer 17555 yang dikendalikan oleh PA6 M68HC11. Gelombang kotak pengondisi isyarat akan masuk pada Input Capture 1 ( IC1 ) M68HC11. Diagram blok secara keseluruhan dapat dilihat pada gambar A.3 di bawah ini.



Gambar A.3 Diagram blok Pengukuran Ketinggian Air Sederhana

Untuk melihat hasil pengukuran ketinggian air dalam bak ( T ) dapat dilihat pada penampil yang terdiri dari 3 digit yang terdiri dari 2 digit di sebelah kiri koma digunakan untuk menampilkan angka puluhan dan satuan, sedangkan 1 digit di sebelah kanan koma digunakan untuk menampilkan angka sepersepuluh. Untuk mengeluarkan data ke penampil digunakan PPI 8255. Hubungan PPI 8255 dengan mikrokontroler M68HC11 dapat dilihat pada bab III. Sebelum penampil ada pengubah bilangan BCD menjadi bilangan tujuh segmen. Bilangan tujuh segmen inilah yang akan digunakan penampil tujuh segmen untuk menghidupkan LED- LED yang ada didalamnya.

Dalam tugas akhir ini, fungsi mikrokontroler M68HC11 dapat diringkas seperti berikut :

- a) Mengeluarkan pulsa melalui pin PA6 yang fungsinya telah dijelaskan di atas.
- b) Melakukan pencacahan
- c) Mengukur tunda waktu
- d) Melaksanakan proses aritmatika dengan persamaan-persamaan yang telah dijelaskan di atas.
- e) Mengubah bilangan Biner hasil pengukuran ketinggian air menjadi bilangan BCD yang siap ditampilkan di PPI 8255.
- f) Mengatur selang waktu keluaran hasil pengukuran awal dengan pengukuran berikutnya yang akan ditampilkan di penampil tujuh segmen.

## IL1 ULTRASONIK

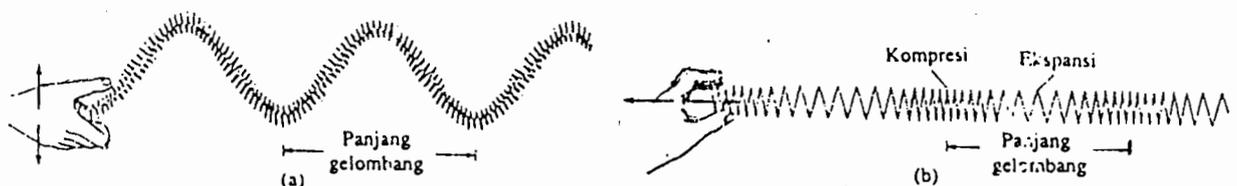
Kata “ ultrasonik “ berasal dari kata *ultra* yang berarti sesuatu yang berada di atas rata-rata dan dari kata *sound* yang berarti bunyi. Dapat diringkas artinya yaitu bunyi yang berada di atas rata-rata pendengaran manusia yang hanya mampu menangkap bunyi dengan frekuensi antara 20 Hz sampai 20 KHz. Ultrasonik merupakan bunyi di atas 20 KHz. Pembahasan akan dimulai terlebih dahulu dengan bunyi, karena merupakan unsur utama dari ultrasonik ini.

### IL.1.1 BUNYI

Bunyi diasosiasikan dengan rasa pendengaran dan oleh karena itu melalui fisiologi telinga dan fisiologi otak, kita dapat menterjemahkan perasaan tersebut sampai pada telinga. Pengertian “ bunyi “ juga mengacu pada perasaan fisik yang merangsang telinga yaitu gelombang longitudinal. Gelombang longitudinal adalah suatu tipe gelombang. Ada tipe gelombang yang lain yaitu gelombang transversal .

#### II.1.1.1 Gelombang Transversal

Jika membayangkan gelombang transversal yaitu dapat diilustrasikan dengan cara menggerakkan sebuah tali seperti gambar dibawah ini :



Gambar A.4 Gelombang Transversal ( a ) dan Gelombang Longitudinal ( b )

Bila gelombang berjalan sepanjang tali, katakan dari kiri ke kanan, partikel tali bergetar naik dan turun dalam arah lintang ( atau tegak lurus ) pada gerak gelombang itu sendiri. Gelombang seperti ini disebut gelombang lintang atau gelombang transversal. Pada gelombang transversal partikel media hanya bergetar pada daerah ruang yang terbatas.

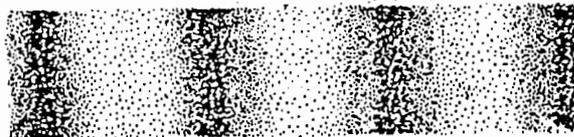
#### II.1.1.2 Gelombang Longitudinal

Dalam sebuah gelombang longitudinal getaran partikel media adalah sama arahnya dengan arah gerak gelombang. Gelombang longitudinal adalah siap dibentuk pada pegas yang ditarik atau diketatkan secara bergantian menekan dan mengembang pada satu ujungnya. Ini ditunjukkan dalam gambar A.5 dibawah ini dan bandingkan dengan gambar gelombang transversal.

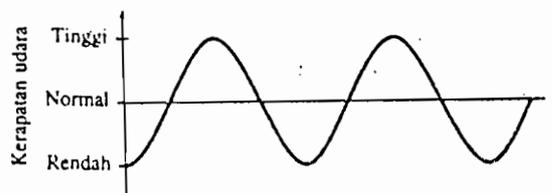
Serangkaian penekanan dan pengembangan perambatan sepanjang pegas. Kompresi adalah daerah dimana gulungan secara sementara saling menjauh. Kompresi dan ekspansi berhubungan dengan puncak dan lembah gelombang transversal.

Panjang gelombang, frekuensi, dan kecepatan gelombang semua mempunyai arti untuk gelombang longitudinal. Panjang gelombang adalah jarak antara kompresi yang berturutan ( atau antara ekspansi yang berturutan ), dan frekuensi adalah jumlah kompresi yang melewati titik tertentu perdetik. Kecepatan gelombang adalah kecepatan dengan mana setiap nampak bergerak dan sama dengan perkalian panjang gelombang dan frekuensi.

Gelombang longitudinal dapat digambarkan secara grafik dengan menggambarkan kerapatan molekul udara ( atau gulungan ketat ) terhadap posisi seperti ditunjukkan gambar A.5 dibawah ini :



(a)



**Gambar A.5 Gelombang longitudinal secara grafis**

Bunyi dapat dibedakan dalam tiga aspek. Pertama, harus ada sumber bunyi, dan seperti dalam beberapa gelombang, sumber gelombang bunyi adalah suatu obyek yang bergetar. Kedua, energi yang dipindahkan dari sumber dalam bentuk gelombang longitudinal. Dan ketiga, bunyi dideteksi oleh alat penerima bunyi.

Kecepatan bunyi berbeda-beda pada bahan-bahan yang berbeda-beda pula. Di udara pada temperatur 0 ( nol ) derajat celcius dan tekanan 1 atmosfer, kecepatan bunyi 331 m/s. Nilai kecepatan bunyi di udara bergantung pada temperatur, ini sangat penting sekali terutama pada gas. Sebagai contoh di udara, kecepatan akan bertambah sebesar 0,60 m/s untuk setiap derajat celcius penambahan temperatur.

$$V = [ 331 + (0.60 \cdot T) ] \text{ m/s}$$

dengan T adalah temperatur dalam derajat celcius. 331 adalah kecepatan bunyi di udara pada suhu nol derajat celcius. Maka jika mengasumsikan  $T = 20^\circ \text{C}$  maka :

$$V = [ 331 + 0.6 \cdot 20 ] \text{ m/s} = 343 \text{ m/s.}$$

Telinga manusia dapat menerima frekuensi pada jangkauan 20 Hz sampai 20.000 Hz . Daerah ini disebut jangkauan pendengaran ( daerah yang dapat didengar). Jangkauan ini berbeda untuk satu individu dengan individu lainnya. Satu kecenderungan umum adalah umur orang, mereka kurang mampu mendengar frekuensi tinggi, sehingga batas frekuensi tinggi sekitar 10.000 Hz atau kurang.

Gelombang bunyi dengan frekuensi diluar daerah yang dapat didengar mungkin mencapai telinga, tapi tidak sadar akan frekuensi tersebut. Frekuensi di atas 20.000 Hz disebut ultrasonik ( Bunyi yang berada di atas rata-rata frekuensi jangkauan pendengaran ). Hewan-hewan dapat mendengar bunyi pada frekuensi ultrasonik contohnya anjing, dapat mendengar bunyi-bunyi dengan frekuensi 50 Khz dan kelelawar dapat mendeteksi frekuensi sekitar 100 Khz. Gelombang bunyi dengan frekuensi di bawah daerah yang dapat didengar ( di bawah 20 Hz ) disebut infrasonik. Sumber-sumber gelombang infrasonik adalah gempa bumi, halilintar, gunung meletus.

Gelombang ultrasonik mempunyai banyak aplikasi misalnya dalam bidang kedokteran, industri, dan lain-lain. Dalam banyak hal, gelombang ultrasonik digunakan terutama dalam hal pantulannya. Bila gelombang ultrasonik mengenai suatu rintangan atau datang pada ujung media di mana gelombang tersebut berjalan, paling tidak sebagian gelombang akan dipantulkan. Contoh pantulan adalah jika mendengar teriakan

yang memantul dari jurang yang jauh yang disebut “ gema “. Tidak semua energi yang dipancarkan akan sempurna dipantulkan, tetapi ada sebagian energi yang diserap oleh bahan pemantul.

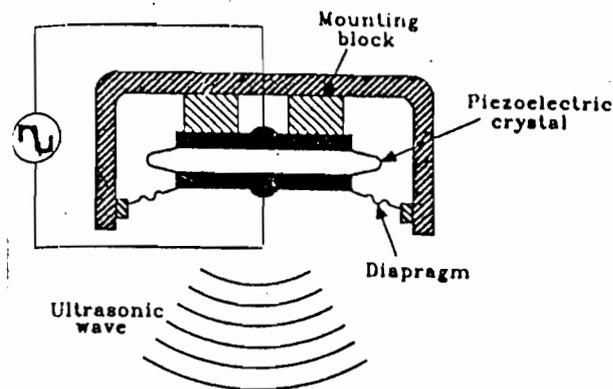
Pantulan bunyi digunakan dalam beberapa aplikasi untuk menentukan jarak. Teknik pulsa gema dalam tugas akhir ini sangat berperanan penting . Alat pengirim ( transmitter ) mengirimkan pulsa bunyi melalui udara dan dipantulkan oleh air. Pantulannya atau gema, diterima detektor beberapa saat kemudian. Interval waktu ini diukur, dan dari sini jarak pantulan benda dapat dihitung karena kecepatan bunyi dalam udara mudah diketahui jika suhu udara diketahui.

Satu alasan penggunaan gelombang ultrasonik lain daripada kenyataan bahwa tidak dapat didengar, bahwa untuk panjang gelombang yang pendek dengan sedikit difraksi, sehingga berkas menyebar lebih sedikit, sehingga walaupun media pemantul kecil masih bisa digunakan sebagai media pemantul. Kalimat di atas ada kata difraksi atau lenturan. Akan bahas apa itu difraksi, karena merupakan hal penting dalam bunyi. Gelombang menyebar ketika dia berjalan, dan ketika menemui rintangan, mereka membelok sedikit. Jadi jika difraksi besar, maka akan terjadi penyebaran gelombang yang besar dan akan terbentuk suatu interferensi. Jika difraksi kecil, maka penyebaran gelombang akan sedikit sehingga gelombang dapat berjalan tegak lurus.

Jadi difraksi dapat didefinisikan bahwa peristiwa dimana gelombang dilenturkan atau melebar di sekitar penghalang.

Jika ingin menggunakan gelombang ultrasonik ini, ada suatu alat yang disebut dengan transduser ultrasonik. Transduser adalah suatu alat yang digunakan untuk mengubah bentuk sinyal yang satu menjadi suatu sinyal yang lain. Misalnya transduser ultrasonik ini digunakan untuk mengubah sinyal listrik menjadi sinyal bunyi ultrasonik (jika transduser ini digunakan sebagai pemancar) dan sinyal bunyi ultrasonik menjadi sinyal listrik (jika transduser ini digunakan sebagai penerima).

Transduser ultrasonik menggunakan osilator frekuensi tinggi untuk membangkitkan frekuensi di atas 20 KHZ. Gambar A.6 di bawah ini merupakan konstruksi dasar dari transduser ultrasonik sebagai pengirim :

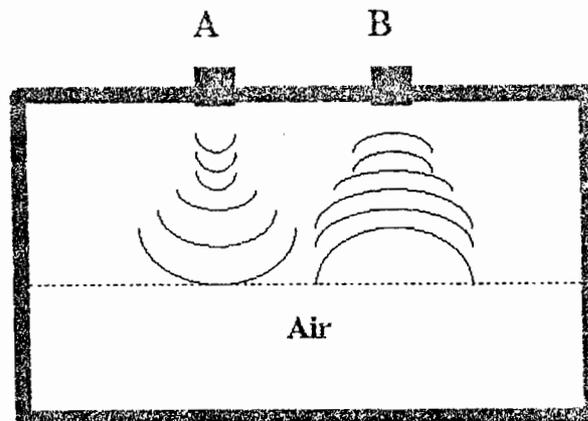


**Gambar A.6 Konstruksi dasar ultrasonik sebagai pengirim**

Terlihat bahwa transduser ini terdiri dari sebuah material kristal piezoelektrik yang terletak di antara dua plat metal. Salah satu metal plat ini diletakkan pada posisi tetap dan plat metal yang lain dihubungkan dengan sebuah diaphragma yang dapat bergetar. Sifat piezoelektrik ditunjukkan oleh material kristal yang nyata. Kristal ini ketika dihubungkan dengan tekanan mekanikal menunjukkan sebuah perbedaan potensial listrik pada titik-titik nyata dari struktur mereka. Efek piezoelektrik didasarkan pada

fakta bahwa sebuah tegangan diaplikasikan melalui kristal, sebuah polaritas menyebabkan bahwa kristal mengembang dan polaritas yang lain menyebabkan kristal mengerut. Gambar A.6 di atas menunjukkan getaran mekanikal frekuensi tinggi menyebabkan diapragma bergetar melawan udara disekitarnya menghasilkan gelombang ultrasonik.

Sebuah penerima ultrasonik didasarkan pada konstruksi yang sama seperti sebuah transmitter ( pemancar ). Perbedaan utama adalah bahwa sumber AC diletakkan dengan sebuah penguat tegangan. Adanya gelombang ultrasonik menyebabkan diapragma bergetar di mana menghasilkan efek tekanan / regangan pada kristal. Efek piezoelektrik sekarang bekerja berlawanan dan getaran frekuensi tinggi yang diaplikasikan ke kristal memproduksi tegangan AC frekuensi tinggi antara 2 buah plat. Transduser ultrasonik umumnya terdiri dari peralatan pemancar dan penerima dan didasarkan pada prinsip bahwa ketika suara dipancarkan melalui udara dan ada sebagian yang dipantulkan. Sinyal pantulan ini disebut gema. Seandainya pulsa sonik dikirim dari transduser memancar sampai permukaan cairan, waktu yang diinginkan untuk gema kembali ke transduser diukur yang merupakan kedalaman udara pada sebuah tempat di atas permukaan air. Hal ini dapat digambarkan seperti gambar A.7 di bawah. Ketika bak penuh, gema yang dipantulkan kembali cepat sehubungan dengan jarak yang pendek pada perjalanan gelombang. Ketika bak kosong, perjalanan gelombang membutuhkan waktu yang lebih panjang dari gelombang pemancar yang dipantulkan kembali ke penerima.



**Gambar A.7** Gema ultrasonik untuk pengukuran level air

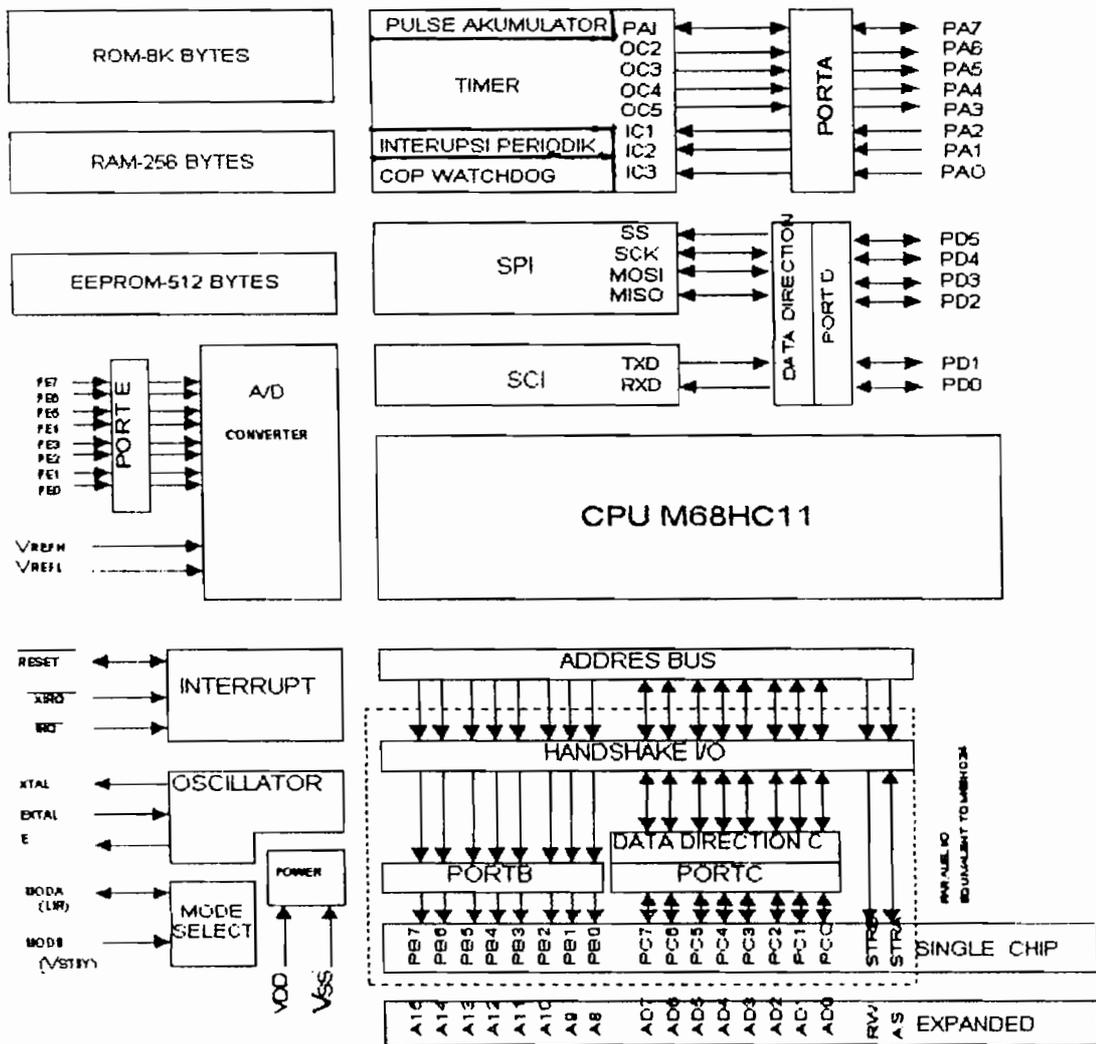
Keterangan : A = transduser ultrasonik sebagai pemancar.  
B = transduser ultrasonik sebagai penerima.

## II.2 Mikrokontroler M68HC11

MC68HC11 adalah sebuah *microcontroller* (MCU) 8 bit yang mempunyai kemampuan periferal dalam satu chip. Satu chip MCU mempunyai memori yang berupa *read only memory* (ROM), *electrically erasable programable ROM* (EEPROM), dan *random access memory* (RAM), konverter analog ke digital 8 kanal dengan resolusi 8 bit, *asynchronous serial communication interface* (SCI), *synchronous peripheral interface* (SPI), *free-running timer system* 16 bit, dan *pulse accumulator* 8 bit. MCU MC68HC11 juga memiliki sistem penanggulangan kesalahan dalam menjalankan program. Jika ditemui kode operasi yang tidak sah maka akan dibangkitkan *nonmaskable interrupt* oleh sistem.

### II.2.1 Struktur MCU M68HC11

Blok diagram MCU 68HC11 dapat dilihat pada gambar A.8 dibawah ini. Blok diagram tersebut menggambarkan bagian-bagian utama dari MCU yaitu CPU, osilator , memori, timer, SPI, SCI, konverter A/D, interupsi, pemilih mode, catu daya, dan paralel I/O.

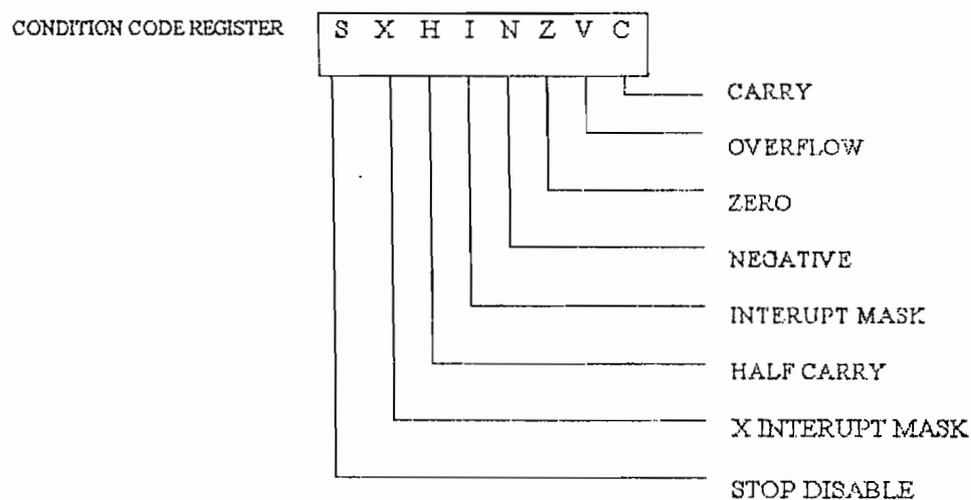


Gambar A.8 Blok diagram M68HC11

## II.2.2 Model untuk Pemrogram

Register CPU yang tersedia bagi pemrogram dapat dilihat pada gambar A.9. CPU MC68HC11 mempunyai dua buah register akumulator yaitu A dan B, dua buah register indeks yaitu X dan Y, Stack pointer (SP), program counter (PC), dan condition code register (CCR). Register A,B dan CCR adalah register 8 bit. Register X, Y, SP dan PC adalah register 16 bit. Dua buah register akumulator 8 bit A dan B dapat digunakan sebagai register akumulator 16-bit yang disebut register D pada beberapa instruksi.

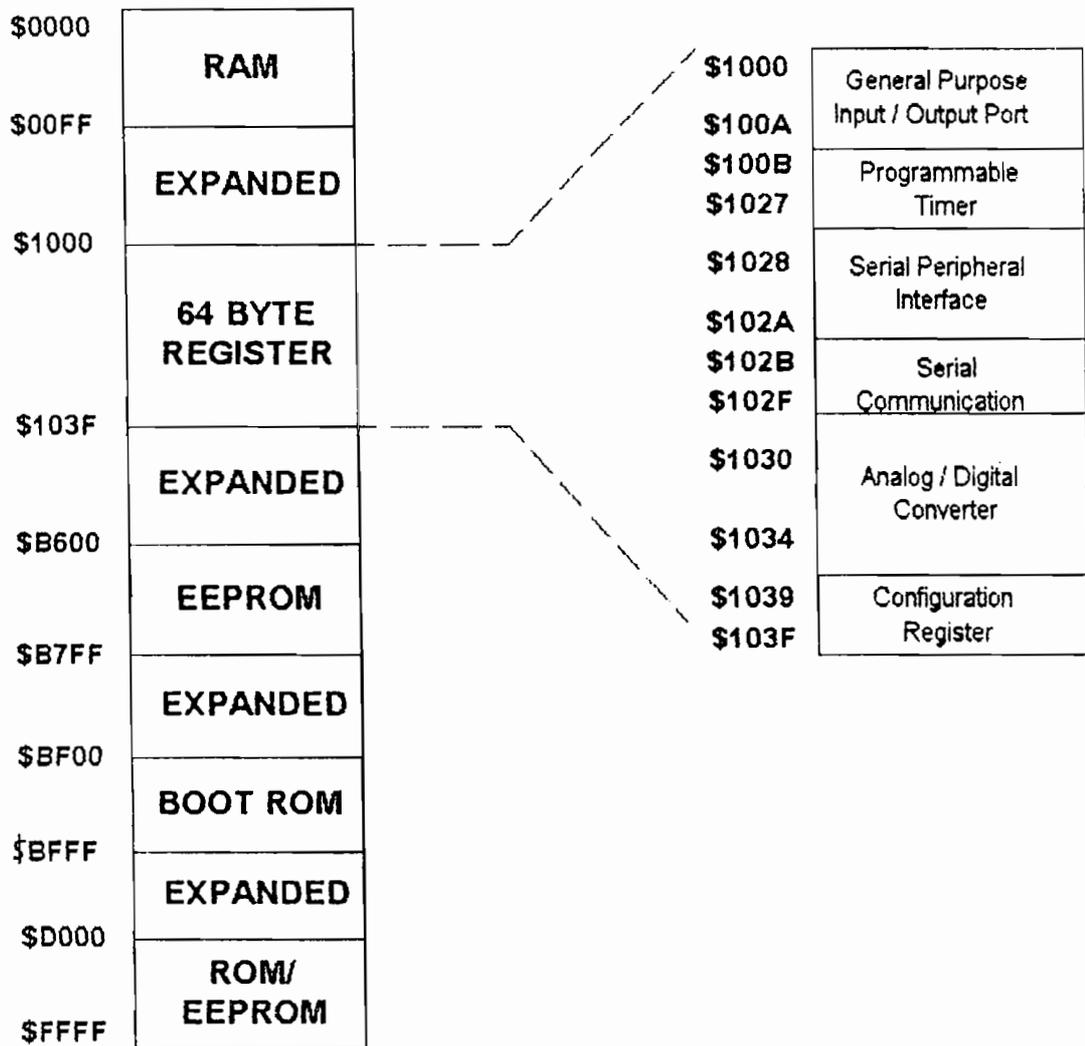
7	ACCUMULATORA	0	7	ACCUMULATORB	0
15	DOUBBLE ACCUMULATOR D				0
15	INDEX REGISTER X				0
15	INDEX REGISTER Y				0
15	STACK POINTER				0
15	PROGRAM COUNTER				0



Gambar A.9 Register CPU M68HC11

### II.2.3 Peta Memori dan I/O

Pada MCU MC68HC11 , ruang alamat memori menjadi satu dengan ruang alamat I/O. Instruksi yang digunakan untuk mengakses memori baik baca maupun tulis sama dengan instruksi yang digunakan untuk mengakses I/O. Peta memori dan I/O untuk MC68HC11 dapat dilihat pada gambar A.10.



Gambar A.10 Peta Memori dan I/O M68HC11

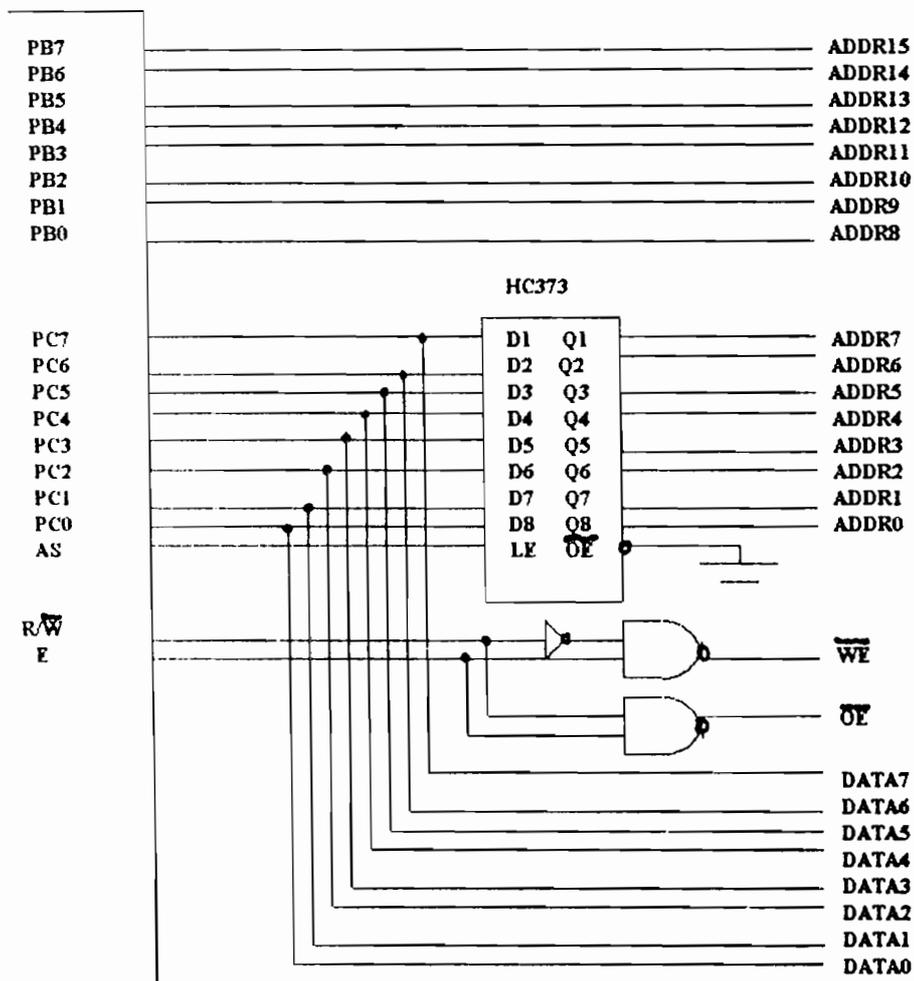
M68HC11 mempunyai 64 byte register kendali yang diletakkan pada alamat 1000H hingga 103FH . Register-register tersebut digunakan untuk mengendalikan dan mengakses unit-unit periferal yang terdapat dalam MCU 68HC11 yaitu I/O paralel , timer I/O serial, Komunikasi serial, dan konverter A/D. Selain itu register tersebut juga digunakan untuk mengatur konfigurasi MCU. Lokasi RAM dan Register kendali dapat diatur kembali dengan mengubah isi register konfigurasi.

#### **II.2.4 Mode Operasi**

MCU 68HC11 mempunyai 4 mode operasi yaitu single chip mode, expanded, test mode, dan bootstrap mode. Pada mode single chip, semua kaki port dapat digunakan sebagai port. Program yang dijalankan disimpan di dalam MCU dan hanya memanfaatkan sumber daya yang ada di dalam MCU. Pada mode expanded, port B dan port C digunakan sebagai bus external. Port B sebagai bus alamat orde tinggi. Port C secara multipleks digunakan sebagai bus alamat orde rendah dan bus data. Kaki AS dan R/W digunakan untuk membantu proses multipleks pada port C. Pada setengah siklus bus pertama port C berfungsi sebagai bus alamat dan kaki AS aktif. Pada setengah siklus bus berikutnya port C berfungsi sebagai bus data dan kaki R/W aktif. Rangkaian perluasan bus dapat dilihat pada gambar A.11.

#### **II.2.5. Unit Pengolah Utama**

Arsitektur CPU 68HC11 menggunakan teknik memory-mapped I/O sehingga semua unit periferal, I/O, dan memori diperlakukan sama di dalam peta memori 64 Kbyte. Jadi tidak ada perintah khusus untuk I/O yang terpisah dengan memori.



Gambar A.11 Demultipleks Alamat/Data

### II.2.5.1. Register

Register-register dalam CPU 68HC11 dapat dikelompokkan menjadi akumulator, register indeks, penunjuk tumpukan (*stack pointer*), pencacah program (*program counter*), dan register kondisi (*condition code register*). Register-register tersebut dapat dilihat pada gambar A.9.

#### II.2.5.1.1. Akumulator

68HC11 mempunyai dua buah akumulator serba guna yaitu A dan B yang mempunyai lebar 8 bit. Dua buah akumulator ini dapat diperlakukan sebagai sebuah akumulator 16 bit yang disebut D pada beberapa instruksi. Akumulator digunakan untuk menyimpan operand dan hasil pada operasi aritmatika atau manipulasi data.

#### II.2.5.1.2. Register indeks

68HC11 mempunyai dua buah register indeks yaitu X dan Y yang masing-masing mempunyai lebar 16 bit. Dalam modus pengalamatan berindeks (*indexed addressing mode*) isi register indeks akan ditambah dengan suatu nilai offset 8 bit yang terdapat dalam instruksi untuk menghasilkan alamat efektif operand yang digunakan dalam instruksi tersebut.

Salah satu register indeks pada umumnya diisi dengan alamat awal register internal (umumnya \$1000) sehingga untuk mengakses I/O dan register kendali dapat menggunakan modus pengalamatan berindeks yang kode operasinya membutuhkan jumlah byte lebih sedikit dari modus pengalamatan *extended*.

#### II.2.5.1.3. Penunjuk tumpukan

Penunjuk tumpukan (*stack pointer, SP*) adalah register 16 bit yang menunjukkan lokasi memori untuk menyimpan data secara *last in first out*. Isi register ini biasa diisi pada awal program untuk menyediakan lokasi tumpukan untuk program tersebut.

Tumpukan digunakan dalam pemanggilan subrutin, interupsi, dan penyimpanan data sementara. Jika sebuah data diambil dari tumpukan maka isi SP akan dinaikkan.

Ketika sebuah subrutin dipanggil oleh perintah JSR atau BSR maka isi pencacah program yang menunjuk lokasi setelah perintah JSR atau BSR tersebut disimpan pada tumpukan. Kemudian pencacah program diisi dengan alamat subrutin yang dipanggil. Setelah subrutin selesai dijalankan yang ditandai dengan perintah RTS maka alamat yang disimpan dalam tumpukan akan diambil dan dikembalikan pada pencacah program sehingga program kembali menjalankan program utama.

Bila terjadi interupsi pada program yang sedang berjalan maka setelah perintah yang sedang dijalankan CPU selesai dikerjakan maka isi pencacah program dan semua isi register CPU disimpan ke dalam tumpukan. Pencacah program diisi dengan alamat subrutin pelayanan interupsi. CPU mulai menjalankan perintah untuk menanggapi interupsi yang terjadi. Setelah selesai isi semua register dan pencacah program diambil kembali dari tumpukan.

#### II.2.5.1.4. Pencacah Program

Pencacah program adalah register 16 bit yang menyimpan alamat perintah berikutnya yang akan dikerjakan CPU. Setelah reset, isi pencacah program akan diisi dengan alamat tergantung pada mode operasi dan penyebab reset seperti pada tabel berikut ini :



Tabel 1.1 Vektor Alamat Reset

Mode	POR atau pin RESET	Clock Monitor	COP Watchdog
Normal	\$FFFE,F	\$FFFC,D	\$FFFA,B
Test atau Boot	\$BFFE,F	\$BFFC,D	\$BFFA,B

#### II.2.5.1.5. Register Kondisi

Register kondisi (*Condition Code Register, CCR*) sering disebut juga *Flag Register* adalah register 8 bit yang berisi 5 bit tanda (C, V, Z, N, dan H), 2 bit *interrupt masking* (IRQ dan XIRQ), dan bit *stop disable* (S). Isi register ini hampir selalu diperbarui setelah mengerjakan suatu perintah. Bit-bit tanda yang berubah untuk suatu perintah dapat dilihat pada lampiran tentang instruksi-instruksi MCU 68HC11.

Bit C (*carry*) menunjukkan adanya carry atau borrow pada instruksi aritmatika. Bit C juga digunakan sebagai tanda kesalahan pada operasi perkalian dan pembagian. Instruksi penggeseran dan rotasi menggunakan bit C untuk operasi penggeseran dan rotasi lebih dari satu word.

Bit V (*overflow*) menunjukkan terjadinya luapan pada operasi aritmatika.

Bit Z (*zero*) menunjukkan hasil nol pada operasi aritmatika, logika dan manipulasi data.

Bit N (*negative*) menunjukkan hasil negatif pada operasi aritmatika, logika, dan manipulasi data.

Bit I (*IRQ mask bit*) digunakan untuk menghambat semua sumber interupsi yang dapat dihambat (*maskable interrupt*).

Bit H ( *half carry* ) menunjukkan terjadinya carry antara bit3 dan 4 pada ALU selama operasi ADD, ABA, atau ADC. Half carry digunakan pada operasi aritmatika BCD. Bit X ( *XIRQ mask bit* ) digunakan untuk menghambat interupsi dari pin XIRQ.

Pemasangan bit S ( *Stop disable* ) akan mencegah instruksi STOP membuat CPU M68HC11 ke dalam kondisi berhenti daya rendah ( *low-power stop condition* ). Jika terdapat perintah STOP saat bit S terpasang maka perintah tersebut dianggap sebagai perintah NOP ( *no operation* ) dan proses berlanjut pada perintah berikutnya. Setelah reset bit S selalu terpasang.

#### II.2.5.2. Mode Pengalamatan

Ada enam buah mode pengalamatan yang digunakan untuk mengakses memori yaitu : *immediate*, *direct*, *extended*, *indexed*, *inherent* dan *relative*. Semua mode kecuali *inherent* menggunakan alamat efektif untuk mengakses memori. Alamat efektif adalah lokasi memori tempat *operand* suatu instruksi diambil atau disimpan.

#### II.2.5.3. Instruksi-Instruksi M68HC11

##### II.2.5.3.1. Instruksi pengisian, penyimpanan, dan pemindahan

Instruksi-instruksi ini digunakan untuk mengisi, menyimpan dan memindahkan isi akumulator. Instruksi ini melibatkan pemindahan data antar register dan memori. Instruksi-instruksi tersebut dapat dilihat di tabel pada lampiran A.

##### II.2.5.3.2. Instruksi aritmatika

Kelompok ini meliputi operasi aritmatika 8 bit dan 16 bit. Operasi aritmatika BCD dilakukan dengan operasi aritmatika biasa dan diakhiri dengan instruksi DAA

untuk menyesuaikan keluaran dalam format BCD. Instruksi CMP digunakan untuk melakukan operasi pengurangan tanpa mengubah isi akumulator dan hanya mengubah isi register kode kondisi. Instruksi MUL digunakan untuk mengalikan bilangan 8 bit dengan bilangan 8 bit dan menghasilkan bilangan 16 bit. Instruksi IDIV digunakan untuk membagi bilangan 16 bit dengan bilangan 16 bit dan menghasilkan hasil 16 bit dan sisa 16 bit. Instruksi selengkapnya dapat dilihat di tabel pada lampiran A.

#### **II.2.5.3.3. Instruksi logika, geser, dan rotasi**

Instruksi logika adalah instruksi yang mengoperasikan bit-bit dalam akumulator secara logika. Instruksi geser digunakan untuk menggeser bit-bit dalam akumulator ke kiri atau ke kanan dengan membuang bit pada ujung tujuan penggeseran. Instruksi rotasi digunakan untuk menggeser bit-bit akumulator ke kiri atau ke kanan dengan memasukkan kembali isi bit pada ujung penggeseran ke pangkal penggeseran. Instruksi-instruksi logika, geser, dan rotasi selengkapnya dapat dilihat di tabel pada lampiran A.

#### **II.2.5.3.4. Instruksi indeks dan tumpukan**

Instruksi indeks dan tumpukan adalah instruksi yang melibatkan register index IX maupun IY dan register penunjuk tumpukan SP yaitu instruksi pengisian, dan pemindahan isi register. Instruksi-instruksi ini menjadi satu kelompok karena register IX, IY dan SP mempunyai lebar 16 bit. Instruksi-instruksi indeks dan tumpukan dapat dilihat di tabel pada lampiran A.

#### **II.2.5.3.5. Instruksi lompatan, percabangan dan interupsi**

Instruksi lompatan ( *jump* ) digunakan untuk menentukan aliran program. Instruksi percabangan (*branch*) digunakan untuk menentukan urutan eksekusi program oleh CPU dengan melihat kondisi tertentu pada register kode kondisi ( *CCR*). Instruksi-instruksi selengkapnya dapat dilihat di tabel pada lampiran.

#### **II.2.5.3.6. Instruksi operasi bit dan register kondisi**

Instruksi-instruksi ini digunakan untuk mengubah isi bit tertentu dari suatu register atau memori. Instruksi-instruksi selengkapnya dapat dilihat di tabel pada lampiran A.

#### **II.2.5.3.7. Instruksi lain**

Instruksi-instruksi ini tidak melibatkan register CPU 68HC11. Instruksi tersebut dapat dilihat di tabel lampiran A.

### **II.2.6. RESET DAN INTERUPSI**

#### **II.2.6.1 Reset**

M68HC11 mempunyai empat sumber reset yaitu *Power-On Reset (POR)*, *reset eksternal (RESET)*, *Computer Operating Properly (COP)*, dan *Clock Monitor Reset*.

POR disebabkan oleh transisi naik tegangan sumber VDD dari keadaan mati ke keadaan hidup. CPU akan mulai bekerja setelah osilator bergetar selama 4064 siklus. Hal ini dimaksudkan agar clock generator stabil sebelum CPU bekerja.

COP adalah sistem untuk mencegah kesalahan perangkat lunak MCU. Jika COP aktif maka suatu kesalahan opcode akan menyebabkan reset.

Reset eksternal dilakukan dengan memberikan logika rendah pada kaki RESET.

Fasilitas ini memungkinkan pengguna untuk mereset MCU setelah bekerja.

*Clock Monitor Reset* digunakan untuk cadangan atau sebagai *backup* sistem COP. Jika clock MCU berhenti maka akan dibangkitkan sinyal reset.

#### II.2.6.2. Akibat Reset

Setelah CPU kembali dari keadaan reset maka semua register internal dan register kontrol akan diinisialisasi ulang. CPU akan mengambil instruksi yang ditunjuk oleh vektor alamat reset. Vektor alamat reset bergantung pada mode operasi MCU dan penyebab reset. Vektor alamat ini dapat dilihat pada tabel dibawah ini :

**Tabel 1.2 Vektor Alamat Reset dan Penyebabnya**

Penyebab Reset	Mode Normal	Special test atau bootstrap
POR atau Reset	\$FFFE,FFFF	\$BFFFE,BFFFF
Kesalahan Clock Monitor	\$FFFC,FFFD	\$BFFFC,BFFD
COP	\$FFFA,FFFB	\$BFFA,BFFB

Isi penunjuk stack dan register CPU yang lain tidak tentu. Bit X dan I dalam CCR akan diset untuk menghalangi mode STOP. Register INIT akan berisi \$00. Hal ini akan memetakan RAM ke alamat \$00 dan register kontrol ke alamat \$1000. Semua subsistem yang ada dalam MCU seperti Pulse Accumulator, COP, Serial Communication Interface, dan Analog to Digital Converter akan dinonaktifkan.

#### II.2.6.3. INTERUPSI

MCU mempunyai 18 Vektor interupsi yang mana 15 sumber interupsi merupakan interupsi yang dapat dihambat ( maskable interupt ) dan 3 sumber interupsi yang tidak

dapat dihambat ( non maskable interrupt ). Interupsi adalah suatu permintaan di tengah-tengah suatu program yang sedang berjalan dan layanan ini biasanya juga terdiri dari beberapa perintah dan harus dilayani. Hal yang harus diperhatikan saat ingin menginterupsi adalah memperhatikan register kode kondisi ( CCR ). Salah satu perhatian terhadap CCR ini adalah memperhatikan bit I dimana dengan mengisi bit I dengan bit '0' ( clearing ) berarti diijinkan untuk menginterupsi dan jika mengisi bit I dengan bit '1' ( setting ) berarti tidak diijinkan untuk interupsi. Pada EVBU vektor interupsi asli terdapat dalam ROM dan digunakan oleh program monitor EVBU yang disebut BUFFALO. Jika pemakai ingin membuat subrutin pelayanan interupsi sendiri maka pada alamat pseudovektor pada tabel di atas diisi dengan perintah :

JMP XX XX

dengan opcode : \$7E XX XX. XX XX adalah alamat dari awal subrutin pelayanan interupsi yang dibuat. Jika terjadi interupsi maka program pelayanan interupsi dari BUFFALO akan melihat isi alamat awal pseudovektor. Jika berisi \$7E maka program akan melompat ke alamat yang ditunjukkan oleh pseudovektor tersebut. Jika tidak maka BUFFALO akan menjalankan program pelayanan interupsinya

#### II.2.6.4. Proses Interupsi

Sebuah interupsi dapat dikenali bila global mask pada CCR dan local mask untuk interupsi tersebut diaktifkan ( enable ). Jika sebuah interupsi dikenali maka CPU akan menanggapi interupsi tersebut setelah menyelesaikan instruksi yang sedang

dikerjakan. Proses pelayanan interupsi dimulai dengan menyimpan isi semua register CPU ke dalam stack dengan urutan seperti tabel di bawah ini :

**Tabel 1.3 Urutan Penumpukan Isi Register Saat Interupsi**

Lokasi Memori	Register CPU
SP	PCL
SP-1	PCH
SP-2	IYL
SP-3	IYH
SP-4	IXL
SP-5	IXH
SP-6	ACCA
SP-7	ACCB
SP-8	CCR

Vektor interupsi dimasukkan ke register PC dan CPU mulai menjalankan instruksi yang ditunjuk oleh vektor tersebut. Setelah menjumpai instruksi RTI ( return from interrupt ) maka isi semua register akan diambil kembali dari stack dan CPU melanjutkan program semula.

## II.2.7. PARALEL INPUT/OUTPUT

M68HC11 mempunyai lima buah port I/O paralel yang masing-masing mempunyai fungsi lain disamping sebagai port I/O paralel. Tabel dibawah ini ringkasan port I/O yang dimiliki M68HC11.

**Tabel 1.4 Port Input / Output**

Port	Pin Input	Pin Output	Pin dua arah	Fungsi lain
Port A	3	3	2	Timer
Port B	-	8	-	Alamat Orde Tinggi
Port C	-	-	8	Alamat orde Rendah dan Bus Data
Port D	-	-	6	SCI dan SPI
Port E	8	-	-	Konverter A/D

Pengaksesan data pada port paralel dilakukan dengan operasi baca/ tulis register internal untuk port tersebut. Fungsi sebuah port ditentukan oleh isi register kontrol untuk port tersebut. Register internal dan register kontrol terletak pada alamat \$1000 hingga \$10FF.

#### II.2.7.1. Port A

Port A mempunyai tiga pin masukan, tiga pin keluaran, dan dua pin dua arah. Port A juga berfungsi sebagai sistem timer.

Data pada port dapat diakses melalui register Port A pada alamat \$1000.

Susunan bit register Port A jika berfungsi sebagai port I/O adalah :

PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0
bit 7	6	5	4	3	2	1	0

Setelah reset Port A akan berisi %10001111. PA0, PA1, dan PA2 adalah masukan. PA4, PA5, dan PA6 adalah keluaran. PA3 dan PA7 adalah dua arah. Susunan bit register Port A bila berfungsi sebagai sistem timer adalah :

PAI	OC2	OC3	OC4	IC4/OC5	IC1	IC2	IC3
Bit 7	6	5	4	3	2	1	0

Arah data pada pin dua arah ditentukan oleh isi register kontrol PACTL.

Register ini juga berfungsi sebagai register kontrol untuk Pulse Accumulator. Susunan bit register ini adalah :

DDRA7	PAEN	PAMOD	PEDGE	DDRA3	I4/O5	RTR1	RTR0
Bit 7	6	5	4	3	2	1	0

Bit yang menentukan arah data pada PA3 dan PA7 adalah DDRA3 dan DDRA7. Jika berisi '0' berarti masuk, bila '1' berarti keluar.

### II.2.7.2. Port B

Pada mode single-chip dan bootstrap, port B berfungsi sebagai port output serbaguna ( general purpose output ). Pada mode expanded , port B berfungsi sebagai jalur keluar alamat orde tinggi ( high order address).

Data pada port B dapat diakses melalui register Port B dengan alamat \$1004.

Susunan bit Port B sebagai port output adalah :

PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Bit7	6	5	4	3	2	1	0

Susunan Port B sebagai jalur alamat orde tinggi adalah :

ADDR15	ADDR14	ADDR13	ADDR12	ADDR11	ADDR10	ADDR9	ADDR8
--------	--------	--------	--------	--------	--------	-------	-------

### II.2.7.3. Port C

Pada mode single-chip, Port C berfungsi sebagai port Input/Output ( dua arah, bidirectional ). Pada mode expanded, port C berfungsi sebagai jalur data yang dimultiplekskan dengan jalur alamat orde rendah.

Data pada port C dapat diakses melalui register PORTC. Susunan bit register PORT C bila berfungsi sebagai port input/output adalah :

PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Bit 7	6	5	4	3	2	1	0

Susunan bit register Port C bila berfungsi sebagai jalur alamat orde rendah adalah :

ADDR7	ADDR6	ADDR5	ADDR4	ADDR3	ADDR2	ADDR1	ADDR0
Bit 7	6	5	4	3	2	1	0

Arah data untuk masing-masing bit pada port C ditentukan oleh isi register DDRC dengan alamat \$1007. Jika pada suatu bit berisi '0' maka bit tersebut berfungsi sebagai input. Jika berisi '1' maka bit tersebut berfungsi sebagai output. Susunan bit register DDRC adalah :

DDC7	DDC6	DDC5	DDC4	DDC3	DDC2	DDC1	DDC0
Bit 7	6	5	4	3	2	1	0

Untuk Port D dan Port E tidak dibahas disini karena dalam tugas akhir, port yang digunakan hanya Port A, Port B, dan Port C

## II.2.8 . SISTEM PEWAKTUAN

### II.2.8.1. Free-Running Counter

Free-Running Counter merupakan elemen utama dari sistem M68HC11, bahkan bisa dikatakan jantung pewaktuan dari M68HC11 ini. Kapasitasnya adalah 16 bit. Pencacah ini mulai mencacah dari \$0000 ( cacahan setelah keluar dari reset ) dan terus mencacah hingga pada batas maksimum cacahan yaitu \$FFFF, pencacah akan mencacah balik ( roll over ) dari \$FFFF ke \$0000. Hal ini ditandai dengan flag overflow dalam keadaan set. Sepanjang power ada, Free-Running Counter akan terus mencacah. Selama MCU bekerja dalam mode operasi normal, tidak ada cara untuk mereset, mengubah atau interupsi dari pencacah ini. Isi pencacah ini dapat dibaca

setiap saat untuk mengetahui waktunya. Isi pencacah ini biasanya dapat dibaca dengan register TCNT.

#### II.2.8.2. TCNT ( *Timer Counter Register* )

TCNT ini digunakan untuk mengakses aliran *free-running counter*. TCNT ini dapat dibaca menggunakan instruksi pembacaan 2 byte seperti load D ( LDD ) atau load X ( LDX). Susunan bit register TCNT sebagai pengakses *free-running counter* adalah sebagai berikut:

bit 15	14	13	12	11	10	9	bit 8
bit 7	6	5	4	3	2	1	bit 0

Jika TCNT dibaca maka TCNT ( Hi) akan dibaca terlebih dahulu sedangkan TCNT (low) akan disimpan dalam buffer dan dibaca pada siklus berikutnya. Dengan demikian yang dibaca adalah isi TCNT saat pembacaan TCNT (Hi). Setelah reset TCNT akan berisi \$0000.

#### II.2.8.3. Struktur Waktu

Gambar dibawah ini menunjukkan diagram blok dari sistem capture ( penangkapan) dan compare ( perbandingan ). Pin-pin yang ada di port A dapat digunakan untuk fungsi timer dan serbaguna input/output. Untuk pin PA3, PA2, PA1, PA0 dari blok ini terdiri dari logika pendeteksian dan pengendalian logika yang mengijinkan terseleksiinya pemuncuan pinggiran dari sebuah input capture. Pin PA6, PA5,PA4,PA3 digunakan baik untuk serbaguna input/ output atau sebagai pin output

compare. PA7 dapat digunakan untuk masukan pulse accumulator, serbaguna input/output dan output compare. Blok diagramnya dapat dilihat pada gambar A.12

Dalam tugas akhir ini hanya digunakan satu input capture yaitu IC1 ( pin PA2) dan pin PA6 yang digunakan untuk mengeluarkan pulsa ( bukan sebagai output compare tetapi sebagai port serba guna keluaran (output )).

### II.2.8.3. Input Capture

Input capture digunakan untuk mencatat isi TCNT saat terjadi perubahan level tegangan pada masukan input capture baik perubahan level tegangan naik, turun, ataupun keduanya. Perubahan level yang dikehendaki dapat dipilih dengan mengatur isi register TCTL2 ( Timer Control Register 2 ). Susunan bit register ini adalah :

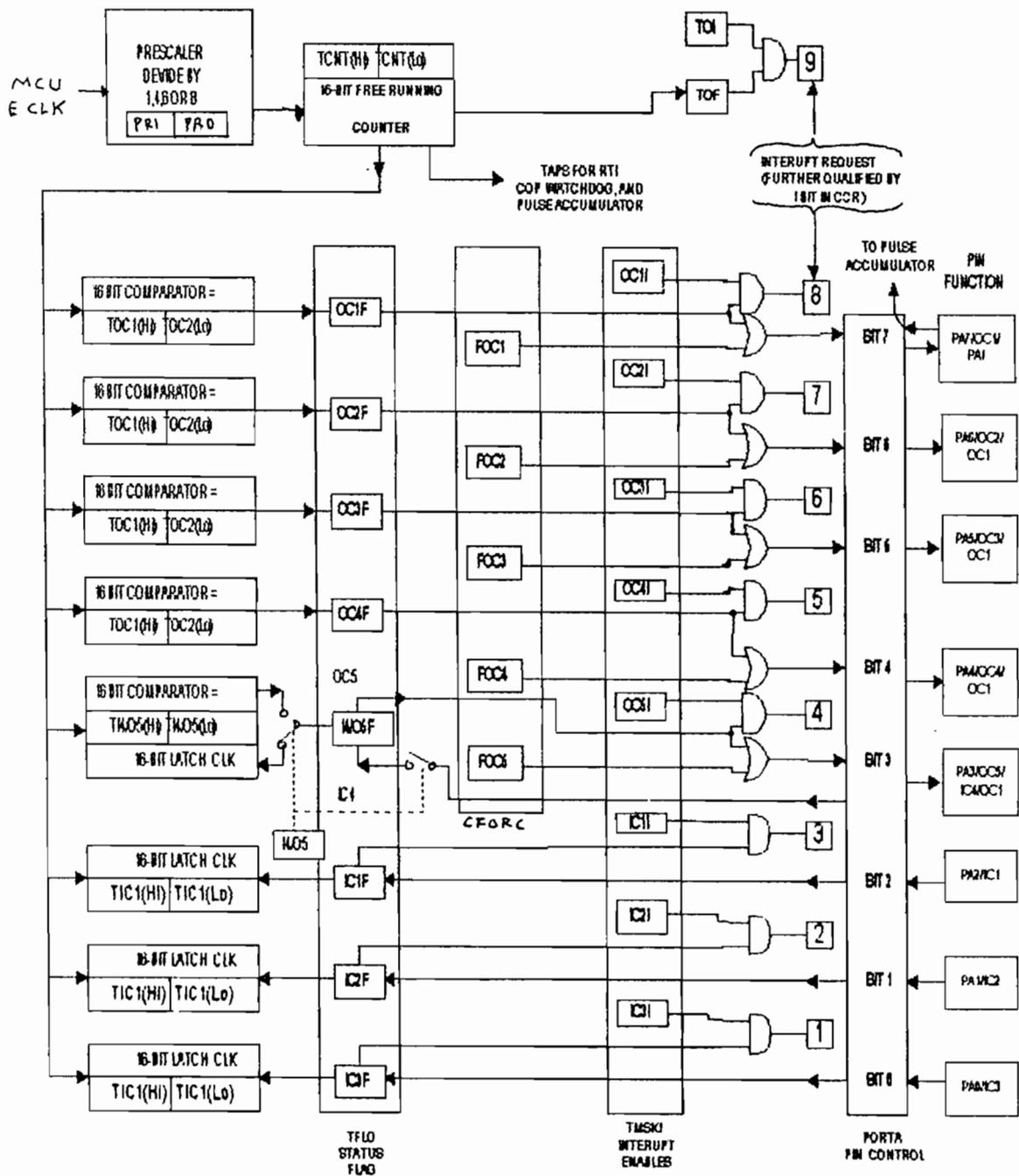
TCTL2 \$1021

<b>EDG4B</b>	<b>EDG4A</b>	<b>EDG1B</b>	<b>EDG1A</b>	<b>EDG2B</b>	<b>EDG2A</b>	<b>EDG3B</b>	<b>EDG3A</b>
Bit 7	6	5	4	3	2	1	0

Terdapat 4 pasang bit EDGxB dan EDGxA yang masing-masing menentukan perubahan level tegangan yang dideteksi oleh masukan yang bersesuaian. Input Capture mempunyai empat buah masukan yang terpisah satu dengan yang lain yaitu IC1, IC2, IC3, dan IC4. Perubahan level tegangan yang dideteksi untuk setiap kombinasi EDGxB dan EDGxA dapat dilihat pada tabel di bawah ini :

**Tabel 1.5 Konfigurasi Timer Control**

EDGxA	EDGxB	Konfigurasi
0	0	tidak mendeteksi
0	1	perubahan naik
1	0	perubahan turun
1	1	perubahan naik dan turun



Gambar A.12 Blok diagram Capture / Compare

Setelah reset TCTL2 akan berisi \$00 sehingga semua masukan input capture tidak aktif. Jika terdeteksi perubahan pada masukan input capture maka isi TCNT pada saat itu disimpan pada register timer input capture yaitu TIC1 hingga TIC3. Register ini mempunyai lebar bit dan masing-masing menempati dua lokasi alamat. Susunan bit dan alamat register TIC1 hingga TIC3 adalah sebagai berikut :

\$1000	bit 15	14	13	12	11	10	9	8	TIC1 (Hi)
\$1011	bit 7	6	5	4	3	2	1	0	TIC1 (Lo)
\$1012	bit 15	14	13	12	11	10	9	8	TIC2 (Hi)
\$1013	bit 7	6	5	4	3	2	1	0	TIC2 (Lo)
\$1014	bit 15	14	13	12	11	10	9	8	TIC3 (Hi)
\$1015	bit 7	6	5	4	3	2	1	0	TIC3 (Lo)

Masukan IC4 juga digunakan sebagai keluaran output compare. Jika ingin digunakan sebagai masukan input capture maka bit I4/05 pada PACTL harus diset '1'. Hasil dari IC4 disimpan pada register TI4/05. Susunan bit dan alamat register ini adalah :

\$101E	15	14	13	12	11	10	9	8	TI4/05 (Hi)
\$101F	7	6	5	4	3	2	1	0	TI4/05 (Low)

Input capture dapat digunakan untuk menghitung tunda waktu dari suatu sistem pengukuran ketinggian air yaitu dimana waktu diukur dari saat pulsa pemancaran dipancarkan sampai waktu penerimaan dari sistem. Saat pulsa dipancarkan , free-running counter ditangkap oleh TCNT dan saat pulsa diterima oleh penerima , input

capture menangkap free-running counter. Untuk lebih jelasnya baca BAB III dan BAB IV karena disini akan dibahas lebih mendalam.

Bila terdeteksi perubahan tegangan input yang dikehendaki maka input capture flag akan set. Input Capture flag terdapat pada register TFLG1.

TFLG1 \$1023

OC1F	OC2F	OC3F	OC4F	I4/05F	IC1F	IC2F	IC3F
------	------	------	------	--------	------	------	------

Jika bit input capture interrupt enable pada register TMSK1 set maka bersama dengan input capture flag set juga dibangkitkan sinyal interupsi input capture.

TMSK1 \$1022

OC1I	OC2I	OC3I	OC4I	I4/05I	IC1I	IC2I	IC3I
------	------	------	------	--------	------	------	------

### IL3 PPI (Programmable Peripheral Interface) 8255

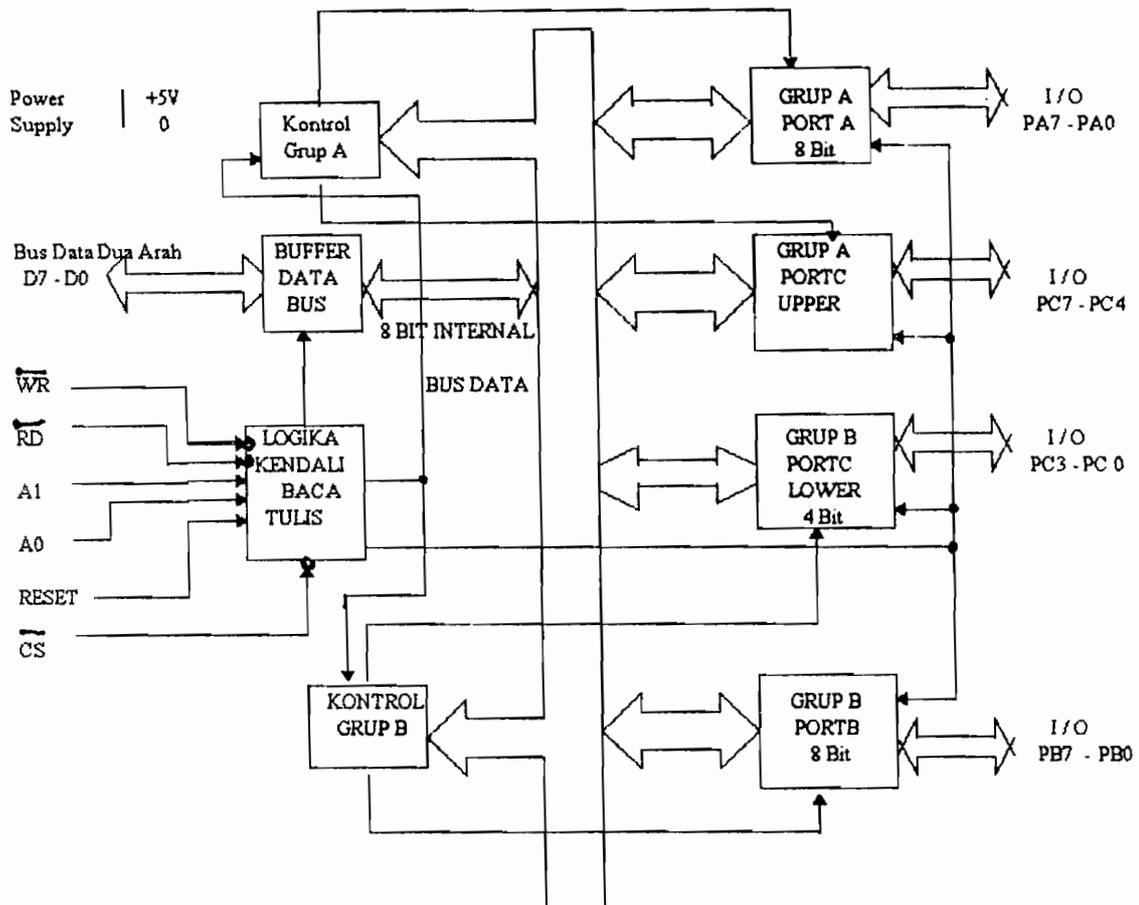
PPI 8255 adalah peralatan masukan/keluaran serbaguna yang dapat diprogram. Chip ini mempunyai 24 pin I/O yang diprogram dalam 2 grup yang masing-masing grup ini mempunyai 12 pin dan digunakan dalam 3 mode operasi yaitu mode 0, mode 1, dan mode 3. Blok diagram PPI 8255 dapat dilihat pada gambar A.13.

Dari blok diagram di atas dapat dijelaskan tiap bloknnya sebagai berikut :

#### 1. Data Bus Buffer

Tiga keadaan buffer 8 bit bidirectional ini digunakan untuk mengantarai (interface) 8255 dengan sistem data bus. Data dikirimkan atau diterima oleh buffer atas

perintah eksekusi masukan atau keluaran oleh CPU. Kata kendali ( Control Words ) dan status informasi juga ditransfer melalui data bus ini.



**Gambar A.13 Diagram blok PPI 8255**

## 2. Logika Baca/ Tulis dan logika kendali

Fungsi dari blok ini untuk mengatur semua transfer internal atau eksternal baik itu data maupun kata ( words ) kendali atau status. Blok ini menerima masukan dari alamat CPU dan bus kendali dan akhirnya, menyebar untuk memerintah 2 grup yaitu grup A dan B.

\*) CS ( Chip Select )

Keadaan rendah dari pin ini memungkinkan terjadinya komunikasi antara 8255 dengan CPU.

\*) RD ( Read )

Keadaan rendah dari pin ini memungkinkan 8255 mengirimkan data atau informasi status ke CPU melalui data bus. Ringkasnya, pin ini mengizinkan CPU untuk membaca dari 8255.

\*) WR ( Write )

Keadaan rendah dari pin ini memungkinkan CPU untuk menulis data atau kata kendali ( control words ) ke dalam 8255.

\*) A0 dan A1 ( Seleksi Port 0 dan Port 1 )

Masukan sinyal ini dalam hubungannya dengan masukan RD dan WR, mengendalikan pemilihan satu dari 3 ports atau register kata kendali ( control words ). A1 dan A0 ini umumnya dihubungkan ke LSB (least significant bit ) dari bus alamat. Dasar operasi dari pin-pin CS, RD, WR, dan A0, A1 dapat dijelaskan dari tabel di bawah ini :

**Tabel 1.6 Tabel Operasi CS, RD, WR, A0 dan A1**

A1	A0	RD	WR	CS	Operasi Masukan ( Baca )
0	0	0	1	0	PORT A → DATA BUS
0	1	0	1	0	PORT B → DATA BUS
1	0	0	1	0	PORT C → DATA BUS
					OPERASI KELUARAN ( Tulis )

0	0	1	0	0	DATA BUS → PORTA
0	1	1	0	0	DATA BUS → PORTB
1	0	1	0	0	DATA BUS → PORTC
1	1	1	0	0	DATA BUS → CONTROL
					FUNGSI YANG TAK DIINGINKAN
x	x	x	x	1	DATA BUS → 3 - STATE
1	1	0	1	0	KONDISI TAK SAH
x	x	1	1	0	DATA BUS → 3- STATE

#### \*) Reset

Keadaan tinggi dari pin ini akan menghapus register kendali dan semua port di set untuk mode masukan.

#### \*) Pengendalian Grup A dan Grup B

Pembentukan fungsional dari masing-masing port diprogram dengan sistem perangkat lunak. Ringkasnya CPU mengeluarkan kata kendali ( control words ) ke 8255. Kata kendali mengandung informasi seperti mode, bit set, bit reset, dan lain-lain; yang mengawali pembentukan fungsional dari PPI 8255.

Masing-masing blok kendali ( Grup A dan grup B ) menerima perintah dari logika kendali baca/tulis, menerima kata kendali dari internal bus data dan menyebarkan perintah-perintah ini ke kumpulan port-port .

Kendali grup A → PORT A dan PORT C ( bit atas yaitu C7-C4)

Kendali grup B  $\implies$  PORT B dan PORT C ( bit bawah yaitu C3-C0)

Register kata kendali hanya dapat ditulis ke blok kontrol grup ini. Tak ada operasi baca register kendali yang diijinkan.

### Port A, B, dan C

8255 mengandung 3 port 8 bit ( A, B dan C ). Semua port dapat dibentuk dengan variasi yang luas dalam fungsinya dengan sistem perangkat lunak, tetapi masing-masing port mempunyai kegunaan yang khusus dalam fungsinya.

*Port A* terdiri dari 8 bit data untuk keluaran latch/buffer dan 8 bit data input latch

*Port B* terdiri dari 8 bit data keluaran/masukan latch/buffer dan 8 bit data masukan buffer.

*Port C* terdiri dari 8 bit data keluaran latch/buffer dan 8 bit data masukan buffer ( tak ada latch untuk input ). Port ini dapat dibagi dalam 4 bit port tergantung kendali modenya. Masing-masing 4 bit port mengandung 4 bit latch dan dapat digunakan untuk sinyal kendali keluaran dan sinyal status masukan dalam hubungannya dengan Port A dan port B.

### Seleksi Mode Operasi

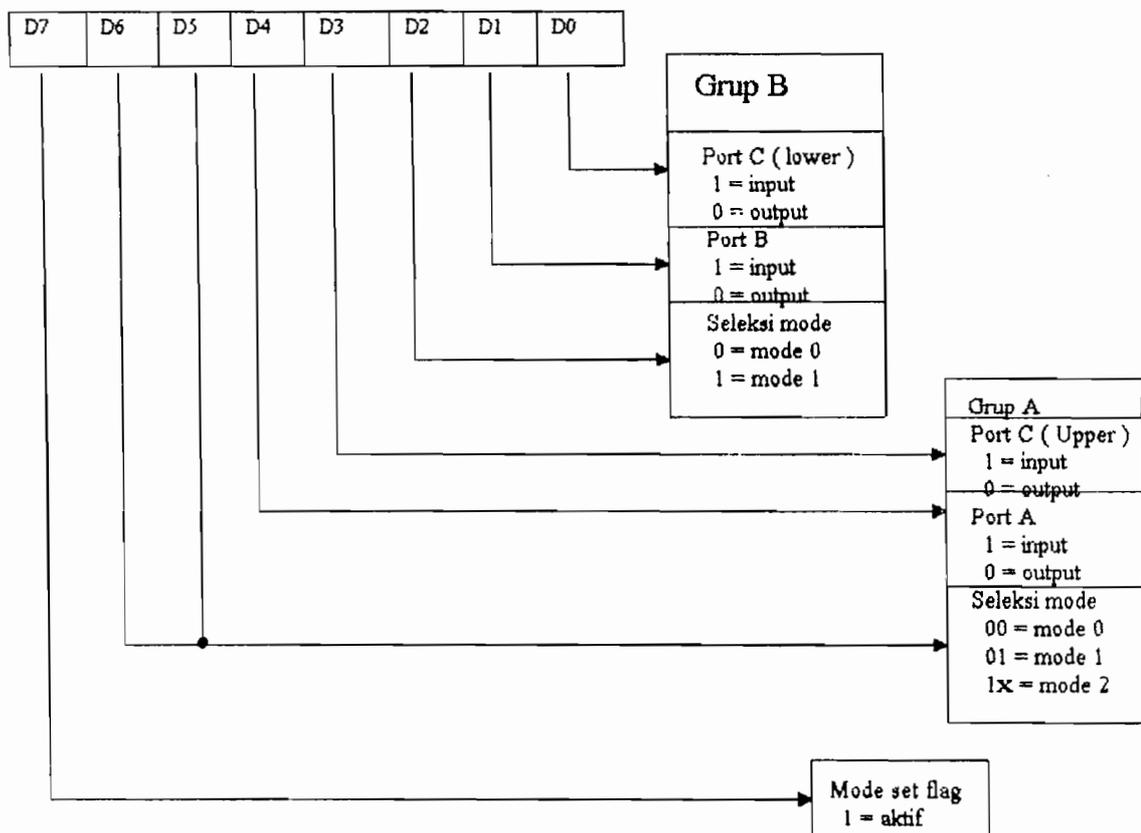
Ada 3 dasar mode operasi yang dapat diseleksi dengan perangkat lunak yaitu :

1. Mode 0  $\implies$  Dasar masukan / keluaran
2. Mode 1  $\implies$  Strobed masukan/ keluaran ( single handshake I/O )
3. Mode 2  $\implies$  Double handshake I/O

Perbedaan dari ketiga mode di atas yaitu terletak pada tata cara ( protokol ) komunikasi dalam transfer datanya. Dalam pemilihan salah satu mode dari ketiga mode ini, CPU harus mengirimkan kata kendali ( control words ) ke 8255. Kata kendali ini mengandung pemilihan mode operasi dan pemilihan port. Format kata kendali ini dapat dilihat pada gambar A.14.

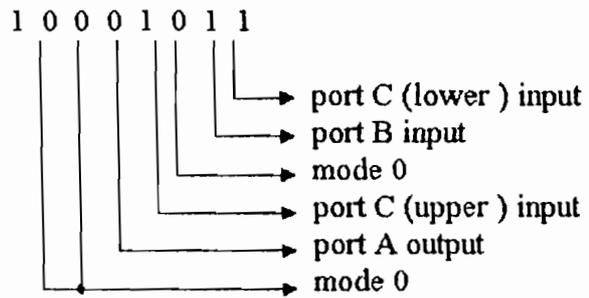
Dalam kesempatan ini, hanya akan dibicarakan mode 0, karena dalam tugas akhir ini yang digunakan hanyalah mode 0 untuk mengeluarkan data ke penampil seven segmen.

Penggunaan format kata kendali di atas dapat dibuat contoh sebagai berikut .



**Gambar A.14 Format Kata Kendali**

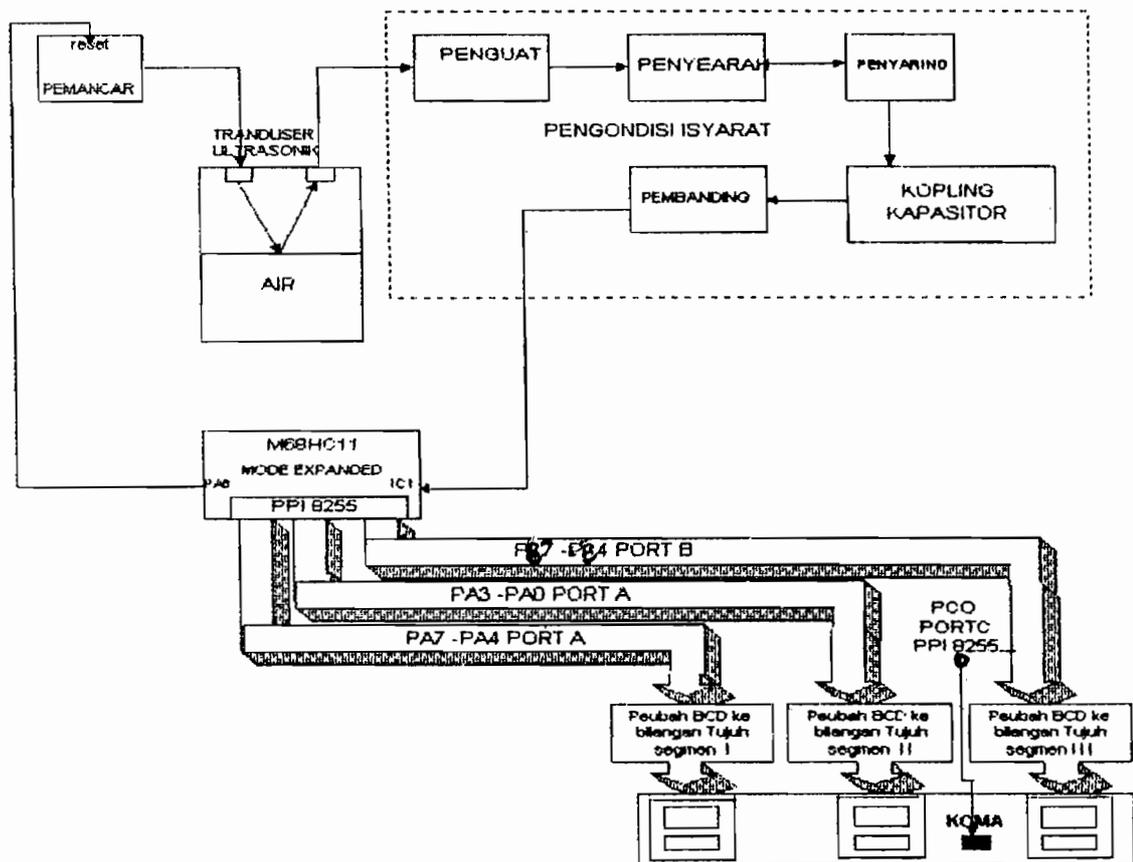
Misalnya kita ingin port A berlaku sebagai output dalam mode 0 dan port B serta port C sebagai input mode 0. Kata kendali untuk itu adalah :



## BAB III

### PERANGKAT KERAS

Dalam bab ini akan dibahas mengenai mode expanded dan rangkain-rangkaian pemancar, penerima yang berupa pengondisi isyarat dan penampil tujuh segmen. Namun sebelumnya akan diberikan diagram blok dari sistem pengukuran ketinggian air dengan ultrasonik berbasis mikrokontroler M68HC11 secara keseluruhan yaitu sebagai berikut :



Gambar B.1 Diagram blok pengukuran ketinggian air

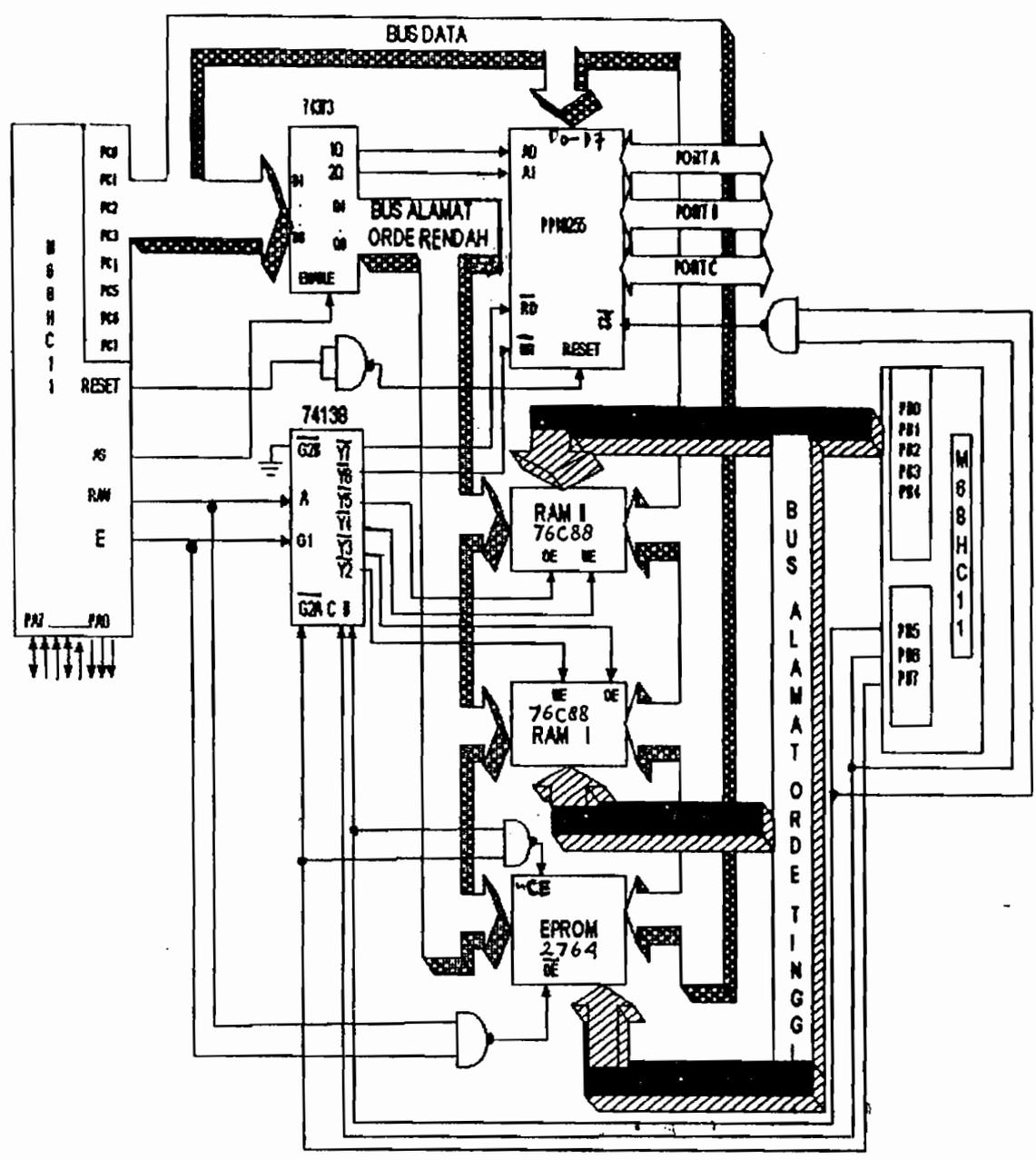
### III.1. MODE EKSPANDED M68HC11

Dalam mode ini, MCU dapat mengakses ruang alamat 64 Kbyte secara penuh. Mode expanded ini digunakan untuk mengatasi keterbatasan memori dan port I/O yang ada pada mode single chip. RAM internal mempunyai memori 512 byte, tetapi karena memori pada alamat \$0047 - \$00FF dari RAM ini digunakan untuk monitor program BUFFALO, maka memori RAM yang bisa digunakan oleh user sebesar 312 byte yaitu ( \$0000 - \$0047 ) + ( \$0100 - \$01FF ). Oleh sebab itu, program yang membutuhkan memori lebih dari 312 byte tidak bisa ditangani oleh mode single chip. Maka mode expanded ini bisa mengatasi keterbatasan memori ini.

Dengan adanya mode expanded maka memori yang terbatas dapat ditambah memorinya dengan adanya tambahan EPROM 2764 ( \$E000H - \$FFFFH ), RAM 76C88 ( \$2000H - \$5FFFH ) dan tambahan expanded I/O dengan alamat \$6000H - \$6003H. Jadi mode expanded ini memorinya terdiri atas memori pada single chip dan memori tambahan yang telah disebutkan di atas. Dalam tugas akhir ini, mode operasi yang digunakan adalah mode expanded di mana hubungan jalur-jalurnya dapat dilihat pada gambar B.2 di bawah ini.

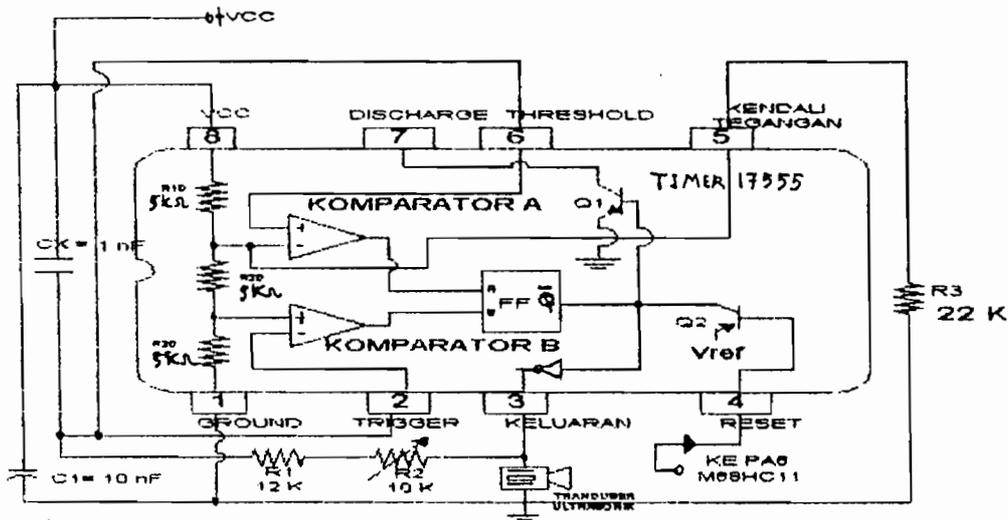
### III. 2. PEMANCAR DENGAN TIMER 17555

Telah dijelaskan pada Bab II bahwa ultrasonik membutuhkan suatu osilator frekuensi tinggi di atas 20 Khz. Osilator yang digunakan adalah osilator kotak dengan frekuensi 40 Khz. Timer yang digunakan dalam tugas akhir ini adalah timer 17555. Timer 555 dengan timer 17555 mempunyai kaki-kaki yang sama. Untuk menghasilkan frekuensi sebesar 40 Khz, IC timer 17555 ini memerlukan komponen tambahan, yang



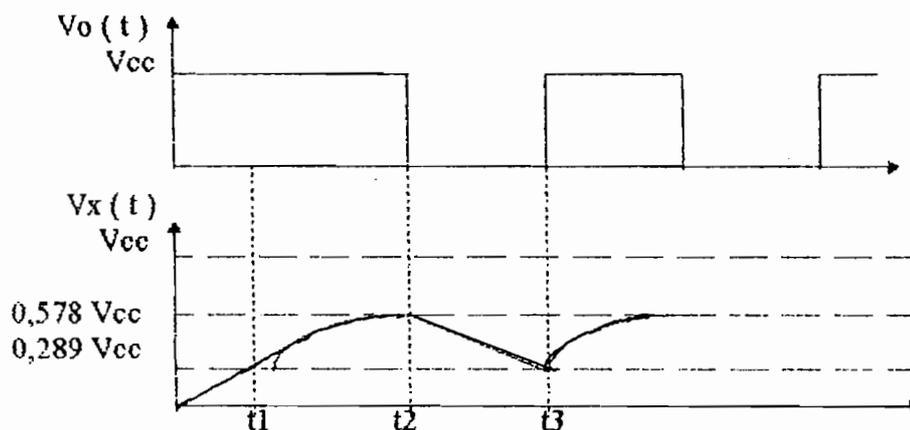
Gambar B.2 Mode expanded M68HC11

dapat dilihat rangkaian dalam IC timer 17555 dengan rangkaian tambahannya pada gambar B.3 di bawah ini. Rangkaian ini diambil dari buku rangkaian *Tulus Kid* dan merupakan rangkaian yang tidak umum, sehingga perlu untuk dianalisis sendiri.



Gambar B.3 IC timer 17555 dengan komponen tambahannya

Analisa rangkaian pada gambar B.3 adalah sebagai berikut. Untuk tahap awal diberikan ilustrasi sinyal keluaran dari timer 17555 serta sinyal pelepasan dan pengisian muatan kapasitor seperti di bawah ini :



Gambar B.4 Keluaran gelombang kotak dan Sinyal akibat kapasitor



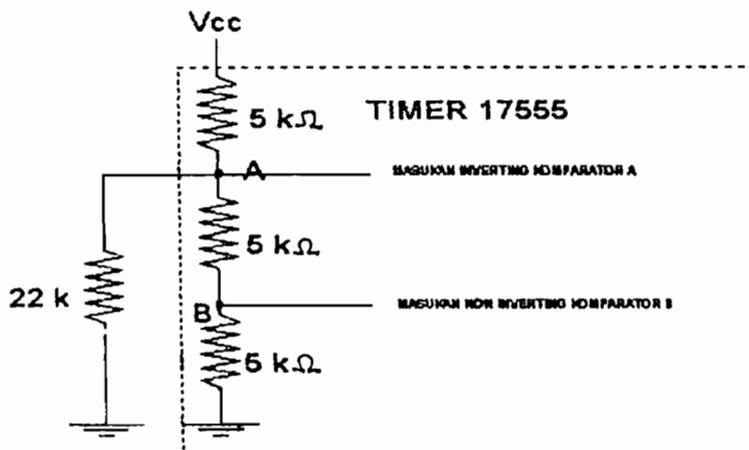
Keterangan gambar B.4 :

$t_1$  = waktu awal pengisian muatan kapasitor yang menyebabkan keluaran flip-flop  $Q=1$ .

$t_2$  = waktu dimulainya pelepasan muatan kapasitor yang menyebabkan keluaran flip-flop  $Q = 0$ .

$t_3$  = waktu dimulainya kembali pengisian kapasitor menyebabkan  $Q = 1$ .

Masukan inverting komparator A dan masukan non inverting komparator B internal timer 1755 pada gambar B.3 mempunyai tegangan yang dapat diperoleh dari pembagi tegangan, dari ketiga resistor internal timer 17555 masing-masing sebesar  $5\text{ K}\Omega$  dan resistor eksternal  $22\text{ K}\Omega$  seperti gambar di bawah ini :



Gambar B.5 Mencari tegangan di titik A dan B

Besar tegangan masukan pin inverting komparator A ( titik A pada gambar ) yaitu :

$$\text{Tegangan di titik A} = \frac{22\text{ K}\Omega // (5\text{ K}\Omega + 5\text{ K}\Omega)}{5\text{ K}\Omega + \{22\text{ K}\Omega // (5\text{ K}\Omega + 5\text{ K}\Omega)\}} \times V_{cc}$$

$$\begin{aligned}
 &= \frac{22 \text{ K}\Omega // 10 \text{ K}\Omega}{5 \text{ K}\Omega + (22 \text{ K}\Omega // 10 \text{ K}\Omega)} \times V_{cc} \\
 &= \frac{6875}{11875} \times V_{cc} \\
 &= 0,5789 V_{cc}
 \end{aligned}$$

Sedangkan tegangan yang ada di pin masukan non inverting komparator ( titik B pada gambar di atas ) adalah :

$$\begin{aligned}
 \text{Tegangan di titik B} &= \frac{5 \text{ K}\Omega}{10 \text{ K}\Omega} \times 0,5789 V_{cc} \\
 &= 0,289 V_{cc}
 \end{aligned}$$

Untuk tujuan analisis, digunakan kondisi awal  $Q=1$  dan  $C_x$  tidak isi. Karena  $V_x(0) = 0$  , flip-flop masukan diawali  $(S,R) = (1,0)$ , maka  $C_x$  mulai mengisi dengan waktu konstan  $\tau = (R_1 + R_2) \cdot C_x$  .....(1)

$V_x$  bertambah secara eksponensial mengikuti :

$$V_x(t) = V_{cc} \cdot (1 - e^{-t/\tau})$$

Keluaran dari komparator B berubah saat  $t = t_1$ , disebabkan kapasitor mengisi kembali sebesar :

$$0,289 V_{cc} = V_{cc} (1 - e^{-t_1/\tau}) \dots\dots\dots(2)$$

dan keadaan flip-flop mulai berubah, sehingga membuat  $(S,R) = (0,0)$ , menyebabkan  $Q$  tetap tinggi ( $Q=1$ ).

Saat  $t = t_2$ , didefinisikan :

$$0,5789 V_{cc} = V_{cc}(1 - e^{-t_2/\tau_1}) \dots\dots\dots(3)$$

keluaran dari komparator A berubah dan masukan flip-flop menjadi (S,R) = (0,1). Sekarang flip-flop merubah keadaan dan menyebabkan  $Q = 0$ . Gambar A.4 menunjukkan bentuk gelombang dari  $V_x(t)$  dan  $V_o(t)$ .

Saat keadaan flip-flop berubah, terjadi pelepasan muatan  $C_x$  dengan waktu konstan  $\tau_2 = (R_1 + R_2) \cdot C_x$ . Ketika  $V_x(t)$  turun dibawah  $0,5789 V_{cc}$ , flip-flop input menjadi (S,R) = (0,0) dan menyebabkan  $Q$  tetap rendah ( $Q = 0$ ). Ketika  $V_x(t)$  turun ke  $0,289 V_{cc}$ , pada  $t = t_3$  didefinisikan dengan :

$$0,5789 V_{cc} e^{-(t_3-t_2)/\tau_2} = 0,289 V_{cc} \dots\dots\dots(4)$$

Saat keluaran dari komparator B berubah, memberikan (S,R) = (1,0) dan menyebabkan  $Q = 1$  ( keluaran timer dalam keadaan tinggi ).

Untuk menemukan periode, kita catat bahwa  $T = (t_3-t_2) + (t_2-t_1)$ . Dari persamaan 4 didapat :

$$0,5789 V_{cc} e^{-(t_3-t_2)/\tau_2} = 0,289 V_{cc}$$

$$e^{-(t_3-t_2)/\tau_2} = \frac{0,289}{0,5789}$$

$$\text{Log}_e e^{-(t_3-t_2)/\tau_2} = \text{Log}_e \left( \frac{0,289}{0,5789} \right)$$

$$-\left( \frac{t_3-t_2}{\tau_2} \right) = \text{Log}_e \left( \frac{0,289}{0,5789} \right)$$

$$t_3 - t_2 = -\tau_2 \cdot \text{Log}_e \left( \frac{0,289}{0,5789} \right)$$

$$t_3 - t_2 = -\tau_2 \cdot \text{Log}_e(1/2)$$

$$t_3 - t_2 = -\tau_2 \cdot \ln(1/2) \dots\dots\dots(5)$$

diketahui bahwa  $\tau_2 = (R_1 + R_2) \cdot C_x$

$$\tau_2 = (12000 + 10000) \cdot 10^{-9}$$

$$\tau_2 = 22000 \cdot 10^{-9} = 2,2 \cdot 10^{-5}$$

Jika kita masukkan ke persamaan 5 maka

$$t_3 - t_2 = -\tau_2 \cdot \ln(1/2)$$

$$t_3 - t_2 = -2,2 \cdot 10^{-5} \cdot \ln(1/2)$$

$$t_3 - t_2 = 0,0000152 \text{ (nilai ini merupakan spasi pulsa)}$$

Dari persamaan 2 didapat :

$$0,289 V_{cc} = V_{cc} (1 - e^{-t_1/\tau_1})$$

$$0,289 = (1 - e^{-t_1/\tau_1})$$

$$1 - 0,289 = e^{-t_1/\tau_1}$$

$$\text{Log}_e(1 - 0,289) = \text{Log}_e(e^{-t_1/\tau_1})$$

$$\text{Log}_e(1 - 0,289) = -t_1 / \tau_1$$

$$t_1 = -\tau_1 \cdot \log_e(1 - 0,289)$$

$$t_1 = \tau_1 \cdot \log_e \left( \frac{1}{1 - 0,289} \right)$$

$$t_1 = \tau_1 \cdot \ln \left( \frac{1}{1 - 0,289} \right) \text{ detik} \dots\dots\dots(6)$$

Dari persamaan 3 didapat  $t_2$  sebagai berikut :

$$0,5789 V_{cc} = V_{cc} (1 - e^{-t_2/\tau_1})$$

$$0,5789 = 1 - e^{-t_2/\tau_1}$$

$$t_2 = \tau_1 \cdot \ln \left( \frac{1}{1 - 0,5789} \right) \text{ detik}$$

Sehingga untuk mengetahui besarnya lebar pulsa maka :

$$t_2 - t_1 = \tau_1 \left( \ln \left( \frac{1}{1 - 0,5789} \right) - \ln \left( \frac{1}{1 - 0,289} \right) \right)$$

$$t_2 - t_1 = \tau_1 (0,865 - 0,341)$$

$$t_2 - t_1 = \tau_1 \cdot 0,524$$

Diketahui bahwa  $\tau_1 = \tau_2$  maka :

$$t_2 - t_1 = 22000 \cdot 10^{-9} \cdot 0,524$$

$$t_2 - t_1 = 0,0000115 \text{ detik}$$

Dengan mengatur R2 pada resistans 12 K $\Omega$  maka diperoleh periode T sebesar

$$T = (t_3 - t_2) + (t_2 - t_1)$$

$$T = 0,0000152 + 0,0000115$$

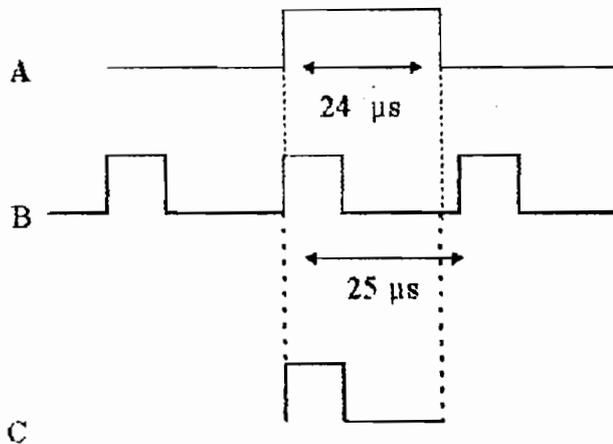
$$T = 0,0000267 \text{ detik}$$

Sehingga dapat diperoleh frekuensi sebesar :  $f = \frac{1}{0,0000267} = 37.453 \text{ Hz}$

Dengan mengatur R2, maka frekuensi sebesar 40 KHz mudah didapat. Resistans R2 semakin kecil maka frekuensi keluaran semakin tinggi.

Pada aplikasinya nanti ternyata frekuensi pemancar sangat menentukan hasil pengukuran. Hal ini akan dijelaskan pada bab V. Pada pin reset dari timer 17555 ini dihubungkan dengan pin PA6 dari port A M68HC11, untuk membatasi keluaran timer 17555. Timer 17555 dapat dikendalikan oleh mikrokontroller melalui pin reset dari timer ini. Saat pulsa ini pada keadaan aras tinggi, maka keluaran timer 17555 akan mengeluarkan suatu gelombang kotak, yang jumlahnya tergantung besarnya lebar pulsa dari pin PA6 ini. Saat pulsa dari pin PA6 ini dalam keadaan aras rendah, maka keluaran timer 17555 tidak akan mengeluarkan gelombang kotak. Jadi besarnya pulsa dari pin PA6 ini, membatasi banyaknya gelombang kotak yang dihasilkan oleh timer 17555. Untuk mengeluarkan pulsa dari pin PA6 maka harus dibuat program khusus yang dapat dilihat pada bab IV mengenai delay1. Besarnya pulsa dari pin PA6 adalah 24  $\mu\text{s}$ . Maka dari lebar pulsa 24  $\mu\text{s}$  ini akan didapat 1 gelombang kotak yang dapat dikeluarkan oleh timer 17555 dengan periode 24  $\mu\text{s}$ , sehingga keluaran timer 17555 terpotong 1  $\mu\text{s}$  dari frekuensi sesungguhnya sebesar 25  $\mu\text{s}$ .

Untuk ilustrasi hal ini dapat dilihat pada gambar B.6 dibawah ini.



**Gambar B.6 Unjuk kerja PA6 pada pin reset timer 17555**

Keterangan gambar B. 6:

A = keluaran PA6 Port A M68HC11

B = keluaran timer jika pin reset dihubungkan dengan aras tinggi terus.

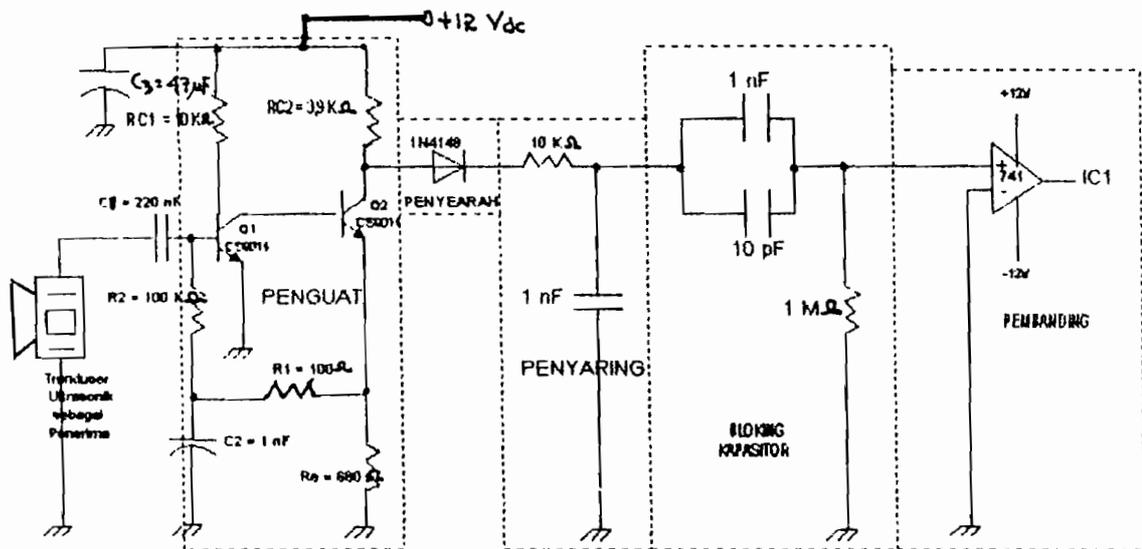
C = keluaran timer, saat PA6 dalam keadaan aras tinggi

Frekuensi pemancar yang keluar dari timer 17555 sangat menentukan dalam pengukuran. Hal ini akan dijelaskan pada bab V mengenai data dan pembahasannya.

### **III.3. RANGKAIAN PENGONDISI ISYARAT**

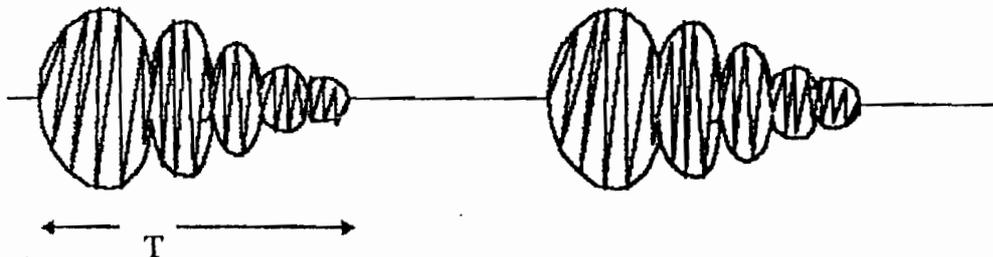
Pada gambar B.1 terlihat bahwa rangkaian pengondisi isyarat terdiri dari penguat, penyearah, penyaring, dan bloking kapasitor. Rangkaian penerima secara utuh dapat dilihat pada gambar B.7.

Sebelum memasuki rangkaian penerima secara mendalam, maka terlebih dahulu akan dibahas keluaran dari transduser ultrasonik sebagai penerima. Diketahui bahwa transduser ultrasonik sebagai pemancar yang dipancarkan berupa pulsa yang diperoleh



**Gambar B.7 Rangkaian pengondisi isyarat secara keseluruhan**

dari osilator dengan timer 17555. Bentuk gelombang keluaran transduser ultrasonik sebagai penerima berupa gelombang AM termodulasi yang jumlahnya banyak menjadi satu kesatuan untuk setiap pulsa yang dipancarkan. Gambar AM termodulasi ini dapat dilihat pada gambar di bawah ini :

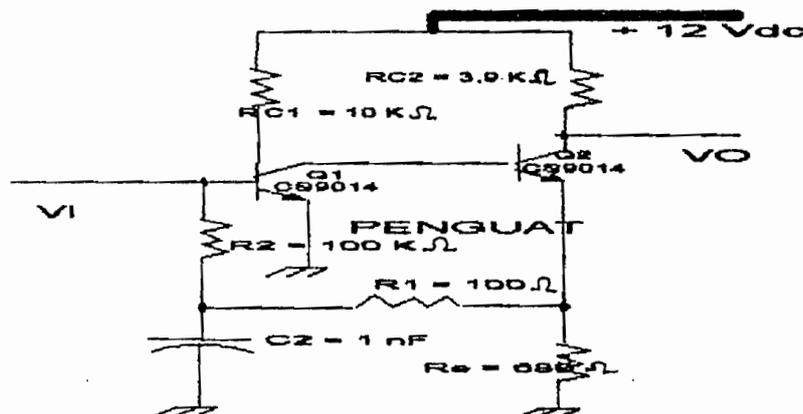


**Gambar B. 8 Gelombang keluaran transduser ultrasonik sebagai penerima**

Setiap pulsa yang dipancarkan melalui transduser ultrasonik sebagai pemancar akan menghasilkan gelombang AM termodulasi seperti gambar B.8. Amplitudo dari gelombang ini sangat kecil, bahkan tidak terbaca oleh osiloskop. Gelombang seperti gambar di atas dapat terbaca setelah melalui sebuah penguatan yang akan dibahas setelah pembahasan ini. Periode gelombang ini besarnya tergantung dari ketinggian air dalam bak. Semakin tinggi air, periode T besarnya semakin kecil. Semakin rendah air, periode T semakin besar dimana muncul gelombang- gelombang AM termodulasi yang baru, yang amplitudonya kecil. Besarnya T berkisar dari 6 ms sampai 8 ms.

### III.3.1. Penguat

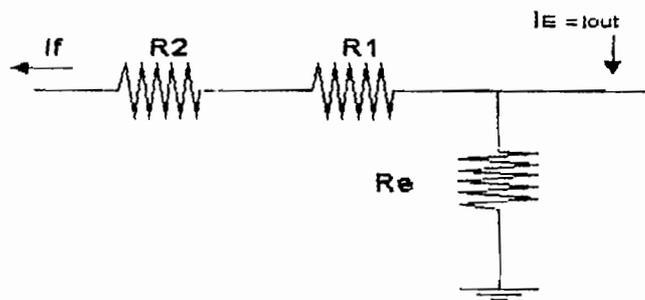
Alasan pemberian penguat karena gelombang keluaran transduser ultrasonik amplitudonya sangat kecil sehingga untuk pengukuran jarak yang cukup jauh tidak mampu. Gambar B.9 dibawah ini merupakan penguat di mana arus bias basis Q1-nya diperoleh dengan umpan balik dari emitor Q2.



Gambar B.9 Penguat dengan umpan balik arus

Pada gambar B.7 terlihat  $C_1$  yang digunakan sebagai blok kapasitor agar sinyal yang ada sebelum kapasitor dapat masuk ke basis transistor  $Q_1$  tanpa adanya tegangan dc yang berasal dari transduser ultrasonik.  $C_3$  digunakan agar sinyal ac yang terbawa  $V_{cc}$  tidak terbawa dalam proses penguatan.

Analisis rangkaian pada gambar B.9 adalah sebagai berikut.  $C_2$  digunakan agar sinyal ac dari emitor transistor  $Q_2$  tidak ikut masuk / diumpanbalikkan ke basis transistor  $Q_1$ .  $R_2$ ,  $R_1$  dan  $R_e$  digunakan sebagai umpan balik sehingga arus basis pada transistor  $Q_1$  dapat tersuplai. Jika berhubungan dengan sinyal dc, maka  $C_2$  akan terhubung buka, sehingga rangkaiannya dapat berbentuk sebagai berikut :



Gambar B.10 Penguat dengan analisis dc

Untuk menganalisa penguat pada gambar B.9 diperlukan dua analisa yaitu analisa dc dan analisa ac. Sekarang akan dibahas penguatan ini dengan analisa dc. Dc disini sangat penting untuk mensuplai arus bias pada basis transistor yang dapat diilustrasikan pada gambar B.10. Diketahui bahwa arus keluaran transistor  $Q_2$  ( $I_{out}$ )  $\approx I_{E_2}$ . Jika diasumsikan  $I_{E_2} = 1 \text{ mA}$ , maka besarnya  $I_f$  ( arus umpan balik ) adalah :

$$I_f = - ( 10^{-3} ) \frac{R_e}{R_1 + R_2 + R_e}$$

$$= -(10^{-3}) \frac{680}{100.000 + 100 + 680}$$

$$= 0,0000067 \text{ A} = 6,7 \mu\text{A}.$$

Sehingga diperoleh penguatan arus yang besarnya :

$$A_{IF} = \frac{I_{out}}{I_F} = \frac{10^{-3} \text{ A}}{6,7 \mu\text{A}} = 149$$

Jadi diperoleh penguatan arus umpan balik sebesar 149.

Sekarang akan dianalisa rangkaian B.9 dengan analisa ac untuk memperoleh penguatan tegangan ac. Menurut Millman, Sutanto ( 1992), rangkaian umpan balik dapat dianalisa dengan membuat rangkaian tanpa umpan balik. Hal ini digunakan untuk memudahkan perhitungan. Konfigurasi penguat dasar tanpa umpan balik, tapi dengan memasukkan pembebanan rangkaian umpan balik ke dalam perhitungan akan diperoleh, dengan penerapan aturan sebagai berikut :

\*) Untuk mendapatkan rangkaian masuk:

(a) Aturilah  $V_0 = 0$  untuk pencuplikan tegangan. Dengan kata lain, hubungkan singkat simpul keluaran.

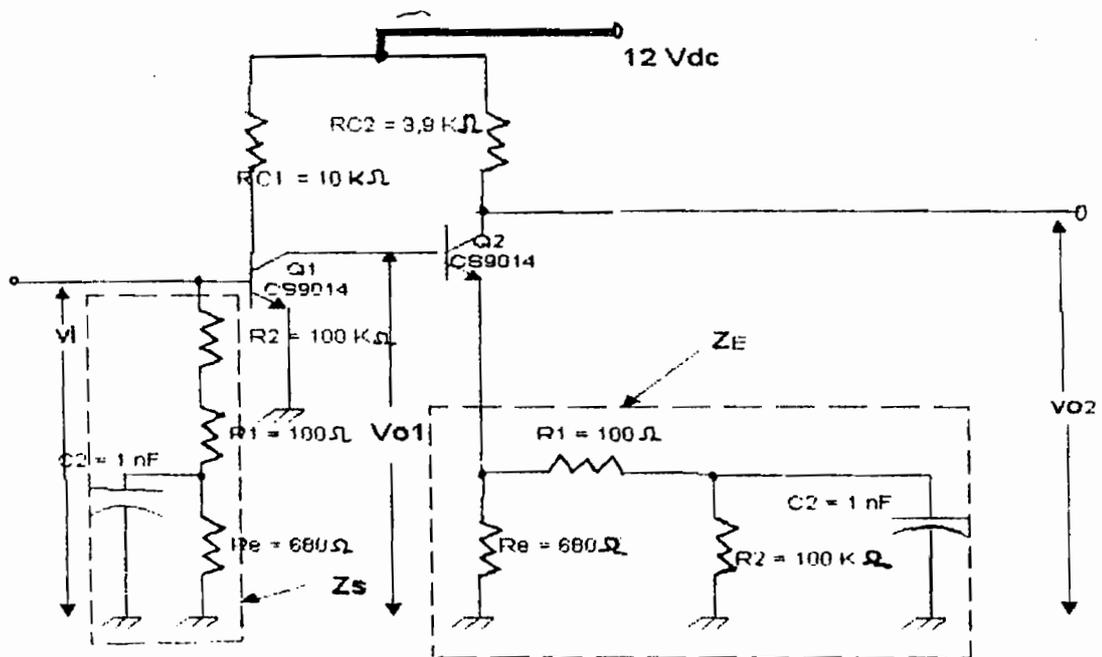
(b) Aturilah  $I_0 = 0$  untuk pencuplik arus. Dengan kata lain, hubungkan terbuka lingkaran keluaran.

\*) Untuk mendapatkan rangkaian keluar :

(a) Aturilah  $V_i = 0$  untuk pembandingan shunt. Dengan kata lain, hubungkan singkat simpul masuk ( Jadi tidak ada satu pun dari arus umpan balik masuk ke masukan penguat )

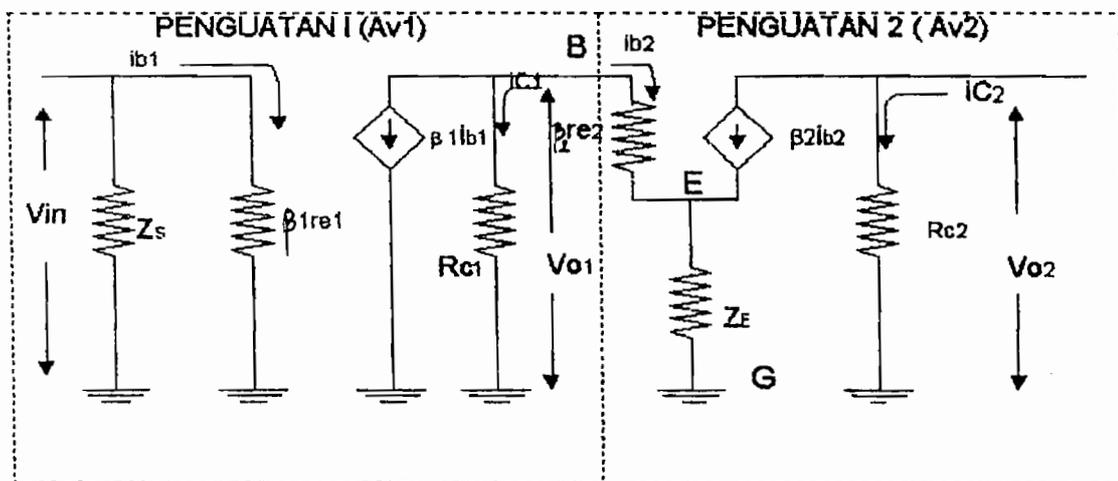
- (b) Aturlah  $V_i = 0$  untuk pembandingan seri. Dengan kata lain, rangkaian terbuka untuk lingkaran masuk ( Jadi tidak ada satupun dari tegangan umpan balik mencapai ke masukan penguat ).

Maka rangkaian umpan balik pada gambar B.9 dapat dibuat rangkaian tanpa umpan balik dengan aturan-aturan di atas. Dengan membuat  $I_0 = 0$  dan mengabaikan arus basis  $Q_2$ , emiter  $Q_2$  dibuka rangkaiannya ( $I_{e2} \approx 0$ ). Konsekuensinya,  $R_1$  dan  $R_e$  ditempatkan seri dengan dengan  $R_2$  dari basis ke emiter  $Q_1$ , dengan  $C_2$  dipasang paralel dengan  $R_1$  dan  $R_e$ . Rangkaian keluar dicari dengan menghubungsingkat simpul masuk ( basis  $Q_1$  ), sehingga menyebabkan penempatan  $R_2$  dan  $R_1$  ini paralel dengan  $R_e$ .  $C_2$  dipasang paralel dengan  $R_2$ . Sehingga rangkaian penguat tanpa umpan balik dapat dibuat seperti pada gambar B.11 di bawah ini.



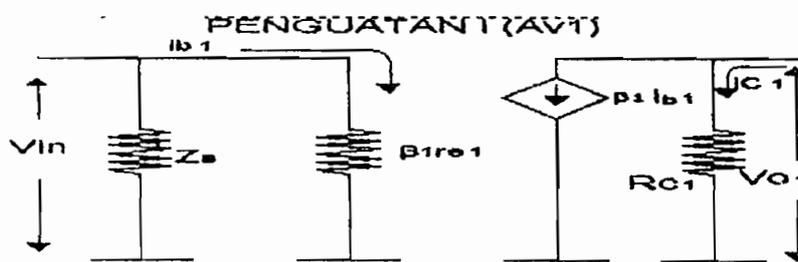
**B.11 Rangkaian penguat tanpa umpan balik**

Untuk analisa ini rangkaian yang ada di gambar B.11 dibuat model ac. Karena menggunakan frekuensi menengah, maka kapasitor yang ada di  $Z_s$  dan  $Z_E$  dihubungkan singkat. Rangkaian ini terdiri dari 2 buah penguat yang dipasang cascade yaitu penguatan pertama dengan transistor  $Q_1$  dengan  $\beta_1 = 60$  dan penguatan kedua dengan  $\beta_2 = 60$ . Model untuk rangkaian yang ada di gambar B.11 dapat dilihat pada gambar B.12 di bawah ini.



Gambar B.12 Model transistor untuk analisa ac

Akan dianalisis terlebih dahulu penguatan pertama dengan model transistornya digambar kembali seperti di bawah ini :



Gambar B.13 Model rangkaian untuk penguatan pertama ( $AV_1$ )

Diketahui bahwa penguatan pertama mempunyai persamaan sebagai berikut :

$$AV_1 = \frac{V_{o1}}{V_{in}} = \frac{I_0 \cdot Z_{O1}}{I_{in} \cdot Z_{in}}$$

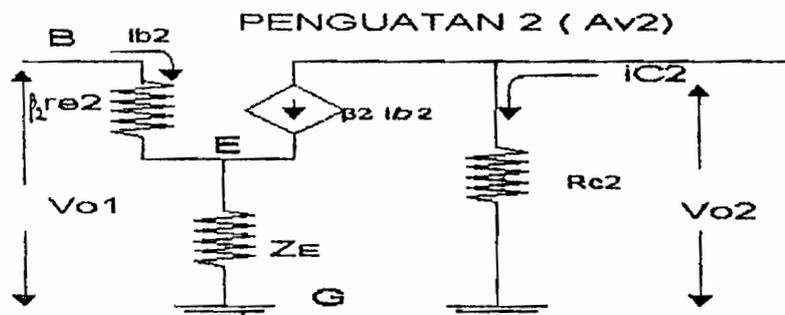
$Z_{in} = Z_s // \beta_1 r_{e1}$ ; dengan  $Z_s$  besar maka diabaikan sehingga  $Z_{in} = \beta_1 r_{e1}$ .

$$AV_1 = \frac{V_{o1}}{V_{in}} = \frac{-i_{c1} \cdot R_{c1}}{i_{b1} \beta_1 r_{e1}} = \frac{-i_{b1} \beta R_{c1}}{i_{b1} \beta_1 r_{e1}} = - \frac{R_{c1}}{r_{e1}}$$

$$r_{e1} = \frac{26 \text{ mV}}{I_E} \quad ; \quad I_{E1} \approx I_{c1} = \beta I_f = 60 \cdot 6,7 \mu\text{A} = 0,000402 \text{ A}$$

$$r_{e1} = \frac{26 \text{ mV}}{0,000402} = 64,68 \Omega$$

Sekarang akan dianalisis penguatan kedua yang modelnya untuk penguatan ini digambar kembali sebagai berikut :



Gambar B.14 Model rangkaian penguat kedua ( $AV_2$ )

Diketahui dari model di atas bahwa penguatan  $V_{o1}$  mempunyai persamaan sebagai berikut :

$$V_{o1} = V_{BE} + V_{EG}$$

$$= i b_2 \beta_2 r_{e2} + (\beta_2 + 1) i b_2 Z_E \dots \dots \dots (7)$$

$$\begin{aligned} V_{o2} &= i c_2 R_{c2} \\ &= -\beta_2 i b_2 R_{c2} \dots \dots \dots (8) \end{aligned}$$

Dari persamaan ( 7 ) ditemukan  $Z_E$ . Sekarang akan dicari besarnya  $Z_E$  terlebih dahulu dengan rangkaian listrik. Karena menggunakan frekuensi menengah kapasitor yang ada di impedansi  $Z_E$  yang ada di gambar B.11 dihubung singkat, sehingga besarnya  $Z_E$  menjadi :

$$Z_E = R_e // R_1$$

$$Z_E = \frac{R_e \cdot R_1}{R_e + R_1}$$

$$Z_E = \frac{680 \Omega \cdot 100 \Omega}{680 \Omega + 100 \Omega}$$

$$Z_E = 87,1 \Omega$$

Dengan diketahuinya impedansi  $Z_E$ , penguatan  $AV_2$  dapat dicari sebagai berikut :

$$AV_2 = \frac{V_{o2}}{V_{o1}} = \frac{-\beta_2 i b_2 R_{c2}}{i b_2 \beta_2 r_{e2} + (\beta_2 + 1) i b_2 Z_E}$$

$$AV_2 = \frac{-R_{c2}}{r_{e2} + Z_E} \dots \dots \dots (9)$$

Telah didapat penguatan pertama dan kedua. Untuk memperoleh penguatan keseluruhan maka kedua hasil penguatan ini dikalikan, sehingga diperoleh penguatan  $AV_{1,2}$  yang merupakan penguatan secara keseluruhan, adalah sebagai berikut :

$$AV_{1,2} = AV_1 \cdot AV_2$$

$$AV_{1,2} = \frac{-R_{c1}}{r_{e1}} \cdot \frac{-R_{c2}}{r_{e2} + Z_E} \dots\dots\dots(10)$$

$$\text{Dicari } r_{e2} = \frac{26 \text{ mV}}{I_{E2}} = \frac{26 \text{ mV}}{10^{-3} \text{ A}} = 26 \Omega$$

Dengan memasukkan nilai komponen yang ada di gambar B.9 ke dalam persamaan (10) maka akan diperoleh  $AV_{1,2}$  sebesar :

$$AV_{1,2} = \frac{10.000 \cdot 3900}{64,68 \cdot (26 + 87,1)}$$

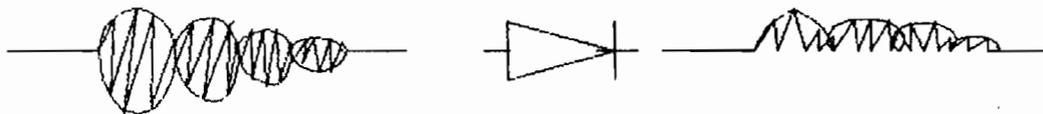
$$AV_{1,2} = \frac{39.000.000}{7315,3} = 5331$$

Demikianlah analisa penguatan yang diperoleh penguatan arus sebesar 149 dan penguatan tegangan sebesar 5331. Sekarang akan dibahas mengenai penyearah.

### III.3.2 Penyearah

Keluaran transduser ultrasonik sebagai penerima telah dikuatkan. Sinyal ini perlu disearahkan mengingat sinyal akhir yang diinginkan berupa gelombang kotak atau

pulsa yang menandakan ada atau tidaknya tegangan dc. Komponen penyearah biasanya adalah dioda. Dioda adalah komponen yang mempunyai kemampuan khusus untuk melalukan arus hanya dalam satu arah. Dioda menjadi sangat konduktif jika diberi tegangan dengan polaritas yang benar yaitu forward bias. Forward bias menghasilkan forward arus. Ketika polaritas berlawanan diaplikasikan disebut reverse bias. Reverse bias menghasilkan reverse arus yang besarnya sangat kecil mendekati nol. Dioda mempunyai resistans yang rendah dalam arah forward bias dan resistans tinggi dalam arah reverse bias. Karena kemampuan dioda inilah, maka mampu bertindak sebagai penyearah.

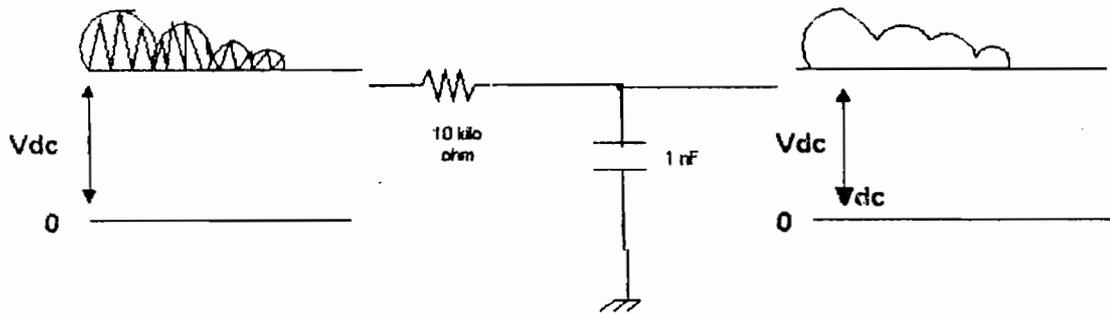


**Gambar B.15 Dioda sebagai penyearah**

Dari gambar B.15 terlihat dioda memotong sinyal AM termodulasi di mana hanya polaritas yang positif saja yang dikeluarkan.

### III.3.3 Penyaring lalu rendah

Digunakan untuk menyaring sinyal-sinyal yang diinginkan. Penyaring yang digunakan dalam tugas akhir ini adalah penyaring frekuensi rendah. Rangkaian penyaring frekuensi rendah dengan masukan dan keluarannya dapat dilihat pada gambar B.16 di bawah ini.

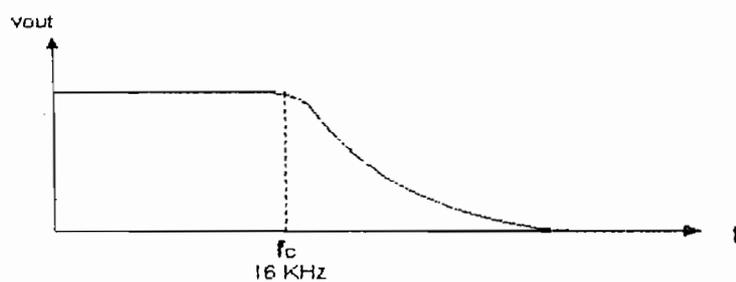


### B.16 Penyaring lalu rendah

Terlihat pada keluaran penyaring frekuensi rendah yaitu sampul dari gelombang AM termodulasi yang telah disearahkan. Frekuensi keluaran penyaring frekuensi rendah ini harus sama atau lebih rendah dari frekuensi yang ditentukan dari nilai C dan R dari penyaring ini yaitu sebesar :

$$f_0 = \frac{1}{2 \pi R C} = \frac{1}{2 \cdot 3,14 \cdot 10000 \cdot 10^{-3}} = 15.923 \text{ Hz} \approx 16 \text{ KHz}$$

Jika digambar dalam bentuk kurva tanggapan frekuensinya dapat dilihat pada gambar B.17 di bawah ini.



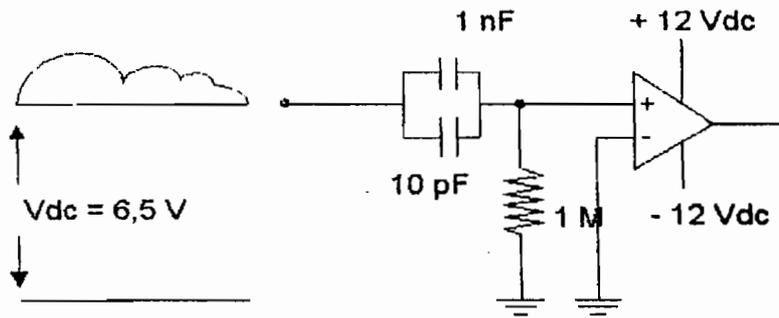
**Gambar B.17** Tanggapan frekuensi penyaring lalu rendah 16 KHz

Dari gambar B.17 di atas, terlihat tegangan maksimum pada frekuensi di bawah 16 Khz. Tegangan mulai menurun pada frekuensi semakin tinggi yaitu frekuensi yang lebih besar dari 16 Khz. Sehingga keluaran pada penyaring seperti pada gambar B.17 mempunyai frekuensi yang rendah daripada frekuensi pembawanya yang mempunyai frekuensi di atas 16 Khz tentunya.

### III.3.4 Bloking Kapasitor

Telah diketahui bahwa keluaran penyaring berupa pulsa di mana masih ada kandungan dc-nya yang terbawa dari keluaran setelah penguat. Komponen dc ini sangat berpengaruh dalam proses perbandingan. Pembanding yang berupa op-amp akan melakukan proses perbandingan jika salah satu pin masukan op-amp ini tegangannya lebih besar dari tegangan kandungan dc dari sinyal yang membawanya. Jadi perlu suatu tegangan referensi sebesar kandungan dc yang dibawa sinyal yang akan dibandingkan. Hal ini sangat merepotkan karena perlu membuat catu daya dc sebagai tegangan referensi.

Dalam tugas akhir ini, komponen dc dari sinyal yang akan dibandingkan ini dihilangkan sehingga tegangan referensinya menjadi nol atau sama saja jika dihubungkan ke ground. Jadi rancangannya lebih murah dan mudah untuk diaplikasikan. Oleh sebab itu perlu komponen yang bisa menghilangkan kandungan dc ini. Komponen tersebut adalah kapasitor sehingga fungsi kapasitor ini sering disebut sebagai kopling kapasitor. Kapasitor ini dipasang paralel antara keluaran penyaring dengan pin pembanding seperti ditunjukkan pada gambar B.18 di bawah ini.

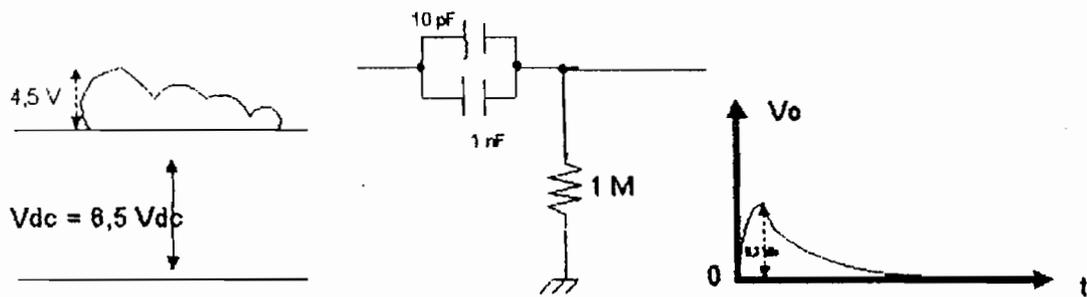


**Gambar B.18 Kapasitor dipasang paralel dengan pembanding**

Terlihat dari gambar B.18 bahwa kapasitor bloking dc ini terdiri dari dua buah kapasitor yang dipasang secara paralel. Hal ini dilakukan karena kapasitor dengan kapasitans 1nF saja belum cukup untuk menghilangkan kandungan dc secara keseluruhan , sehingga perlu ditambahi kapasitor dengan kapasitans 10 pF. Penambahan ini benar-benar menghilangkan kandungan dc-nya.

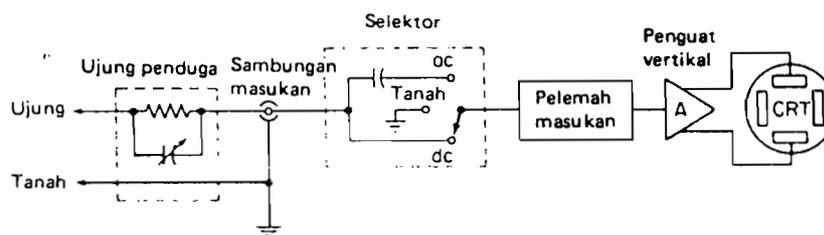
Dari gambar B.18 terlihat juga adanya resistor  $1M\Omega$  yang dipasang paralel dengan kapasitor bloking dc. Hal ini dilakukan agar arus pada pin yang masuk pembanding jika dihubungkan probe osiloskop tidak short ke osiloskop ini. Jika tidak ada resistans  $1M\Omega$  ini arus short ke osiloskop sehingga tidak terjadi proses pembandingan. Pemilihan resistans yang besar ini agar tegangan yang masuk ke pembanding diharapkan sama dengan tegangan keluaran setelah bloking kapasitor.

Pada kenyataannya, sinyal keluaran bloking kapasitor bukan berupa sinyal seperti sinyal masukan yang hilang komponen dc-nya tetapi berupa gelombang gerigi. Perubahan ini dapat dilihat pada gambar B.19 di bawah ini .



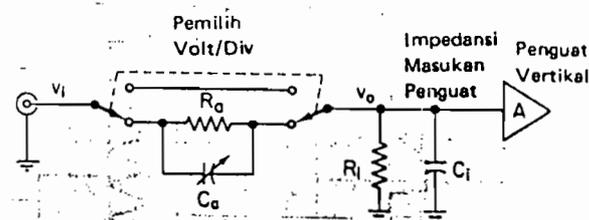
**Gambar B.19** Perubahan bentuk sinyal akibat kapasitor liar dalam osiloskop

Dari gambar B.19 terlihat perubahan bentuk sinyal di keluaran rangkaian blocking kapasitor. Perubahan ini terjadi karena adanya kapasitansi liar pada bagian penyusut di blok diagram sistem penyimpang vertikal osiloskop seperti pada gambar B.20 di bawah ini .



**Gambar B.20** Blok diagram sistem penyimpang vertikal osiloskop

Kapasitansi liar ini pada sistem penyimpang vertikal osiloskop ini terletak di antara penyusut dan penguat. Terlihat pada gambar B.21, rangkaiannya berupa integrator.



**Gambar B.21** Rangkaian penyusut ( atenuator ) dengan kapasitansi liarnya

Pengaruh integrator terhadap masukan berupa sinyal kotak atau pulsa, keluaran integrator bukan lagi berupa pulsa atau kotak seperti pada masukan. Untuk lebih jelasnya, akan dibahas mengenai integrator terlebih dahulu.

Integrator adalah rangkaian yang terdiri dari R dan C yang keluarannya berupa tegangan akibat proses pengisian dan pelepasan muatan dalam kapasitor. Lama pengisian dan pelepasan muatan kapasitor ini, tergantung dari besarnya waktu konstan ( $\tau = R.C$ ). Menurut Floyd T.L (1991), ada 2 kondisi yang berpengaruh terhadap tanggapan pulsa masukan pada integrator, yaitu :

1. Ketika lebar pulsa masukan ( $t_w$ ) sama atau lebih besar dari 5 kali waktu konstan ( $\tau$ )  $\longrightarrow$  ( $t_w \geq 5 \tau$ ), maka pengisian dan pelepasan muatan kapasitor dapat dilakukan secara penuh.
2. Ketika lebar pulsa masukan ( $t_w$ ) lebih kecil dari 5 kali waktu konstan ( $t_w < 5\tau$ ), maka tegangan keluaran kapasitor tidak akan mencapai tegangan masukan secara penuh sebelum akhir pulsa. Untuk waktu konstan yang lebih panjang, keluaran mencapai tegangan yang lebih rendah dari tegangan masukan karena kapasitor tidak dapat mengisi sebanyak selama lebar pulsa masukan. Ketika waktu konstan ( $\tau$ ) jauh lebih besar dari lebar pulsa masukan, kapasitor akan mengisi muatan sangat sedikit dan tegangan keluarannya menjadi hampir ditiadakan.

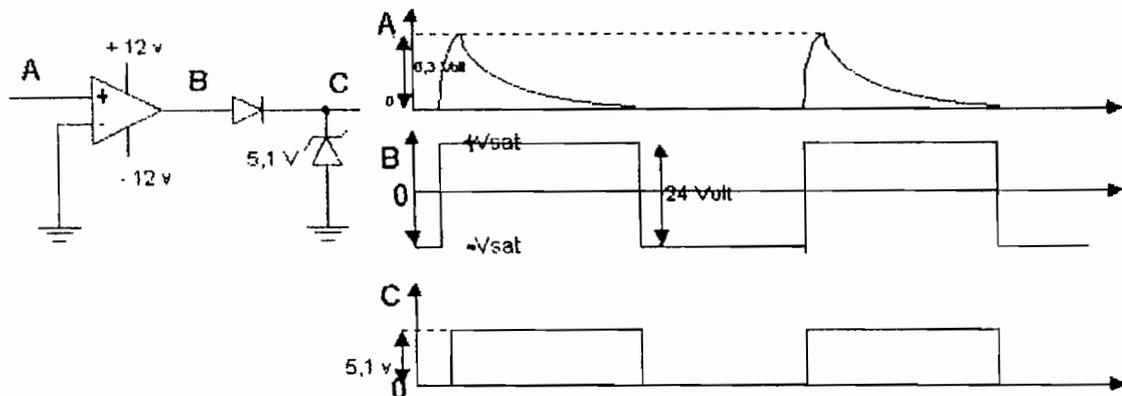
Berdasarkan 2 kondisi di atas, perubahan sinyal pada keluaran blocking kapasitor menjadi gelombang gerigi yang tegangannya jauh lebih kecil dari tegangan masukan, akibat dari lebih besarnya waktu konstan ( $\tau$ ) dari rangkaian penyusut.

### III.3.5 Pembanding, Penyearah keluaran pembanding, dan Pembatas tegangan

Digunakan untuk membentuk gelombang kotak, di mana gelombang kotak ini merupakan masukan untuk pin Input Capture 1 M68HC11. Rangkaian pembanding ini merupakan rangkaian akhir dari bagian pengondisi isyarat. Masukan dari salah satu pembanding ini berasal dari keluaran penyaring dan bloking kapasitor yang telah dibahas di depan. Masukan ini akan dibandingkan dengan sebuah tegangan referensi yang besarnya nol volt karena kita hubungkan dengan ground.

Pembanding yang digunakan adalah sebuah op-amp 741. Masukan pin non inverting merupakan sinyal yang akan dibandingkan, yang merupakan keluaran dari integrator. Masukan pin inverting dihubungkan dengan ground sebagai tegangan acuan. Jika sinyal non inverting tegangannya melebihi nol Volt, maka keluaran op-amp akan mempunyai amplitudo sebesar  $+V_{sat}$  yaitu +11 Volt. Dan jika tegangan sama dengan tegangan acuan maka keluaran op-amp akan sama dengan  $-V_{sat} = -11$  Volt. Jadi besarnya tegangan keluaran op-amp 22 Volt. Karena mikrokontroler pin IC1 ( Input Capture 1 ) hanya membutuhkan tegangan sebesar 5 Volt, maka keluaran op-amp tegangannya perlu disearahkan yaitu dengan pemberian dioda sehingga keluaran dioda tegangannya sebesar 11 Volt. Tegangan ini masih terlalu besar untuk masuk ke dalam pin IC1 mikrokontroler. Oleh sebab itu tegangan 11 Volt ini harus dipotong tegangannya lagi yaitu dengan pemberian dioda zener yang mempunyai pembatas tegangan sebesar 5,1 Volt, sehingga tegangan keluaran diharapkan sebesar 5,1 Volt. Tegangan ini cukup untuk tegangan masukan pada pin IC1. Untuk lebih

mengerti penjelasan ini, diberikan sebuah ilustrasi dari penjelasan di atas dengan gambar B.22 di bawah ini.



**Gambar B.22 Rangkaian pembanding, penyearah dan pembatas tegangan**

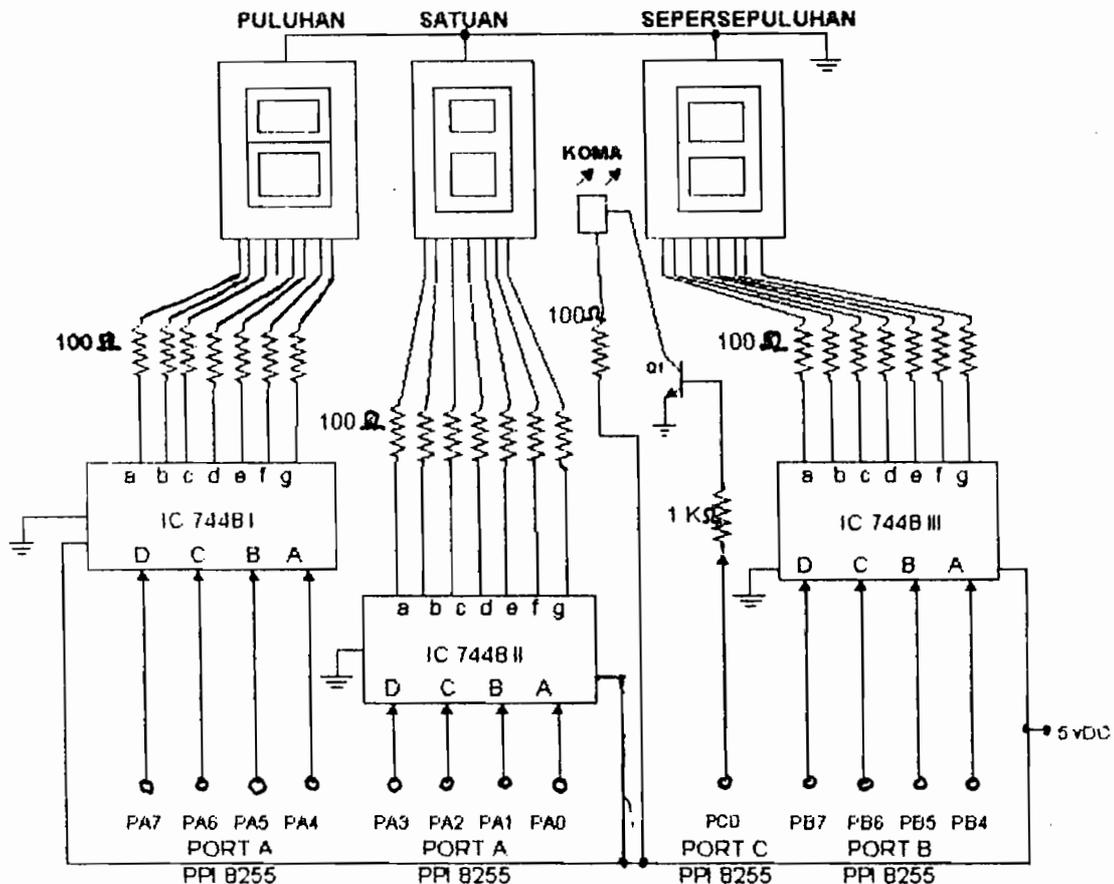
Keterangan gambar B.22 :

- A = gambar sinyal masukan komparator
- B = gambar sinyal keluaran komparator
- C = gambar sinyal keluaran dioda zener sebagai pembatas tegangan

#### III.4 Penampil

Telah dibahas tentang rangkaian pengondisi isyarat. Rangkaian pengondisi isyarat di atas pada dasarnya mengubah tegangan termodulasi AM menjadi tegangan yang berbentuk gelombang kotak dengan tegangan 5,1 Volt.

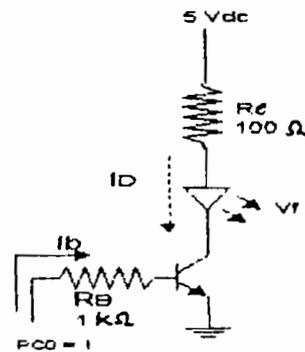
Hal yang tidak kalah pentingnya dengan rangkaian pengondisi isyarat adalah rangkaian penampil. Penampil yang digunakan adalah penampil tujuh segmen. Penampil ini terdiri dari 3 buah penampil tujuh segmen yang digunakan untuk menampilkan hasil pengukuran ketinggian air dalam bak. Rangkaian secara keseluruhan dari penampil dapat dilihat pada gambar B.23 di bawah ini.



**Gambar B.23 Rangkaian penampil secara keseluruhan**

3 buah tujuh segmen menandakan pengukuran yang besarnya terdiri dari puluhan, satuan dan sepersepuluh. Sebelum tujuh segmen sebagai sepersepuluh ada sebuah tanda koma yang menggunakan LED. Dari gambar B.23 di atas terlihat ada 3 buah IC 7448. IC 7448 ini digunakan untuk mengubah bilangan BCD menjadi bilangan tujuh segmen. Masukan IC 7448 ( D C B A ) berasal dari Port PPI 8255. Port PPI 8255 ini diset sebagai keluaran ( lihat Bab II ). Port A ( PA7, PA6, PA5, PA4 ) PPI 8255 digunakan sebagai masukan IC 7448 I yang keluarannya adalah bilangan tujuh segmen yang nantinya untuk menampilkan desimal puluhan dari penampil. Port A ( PA3, PA2, PA1,

PA0 ) PPI 8255 digunakan sebagai masukan IC 7448 II yang keluarannya adalah bilangan tujuh segmen yang nantinya untuk menampilkan desimal satuan. PC0 dari Port C PPI 8255 digunakan sebagai indikasi ( tanda ) koma pada penampil. Transistor Q1 bertindak sebagai saklar. Jika PC0 = 1 maka LED akan hidup dan jika PC0 = 0 maka LED akan mati. Rangkaian tanda koma yang menggunakan transistor sebagai saklar dapat dilihat pada gambar B.24 di bawah ini.



**Gambar B.24 Rangkaian tanda koma**

Transistor dalam rangkaian tanda koma ini berfungsi sebagai saklar. Saat transistor Q1 ON, ada arus basis yang mengalir ke basis transistor menyebabkan resistans yang ada di antara kolektor dan emitor sangat kecil sehingga ada arus yang mengalir di kolektor. Arus yang ada di kolektor ini dibatasi oleh RC. Karena transistor ON maka LED hidup. Transistor dalam keadaan ON biasanya beroperasi dalam daerah saturasi di mana arus maksimum mengalir dalam transistor. Saat transistor ON, ada arus yang mengalir melalui dioda sebesar :

$$I_D = \frac{V_{CC} - V_f - V_{CE}}{R_C} = \frac{5 - 0,7 - 0}{100} = 43 \text{ mA}$$



Biasanya besarnya  $V_{ce}$  sangat kecil dibandingkan dengan nilai  $V_{cc}$  saat saturasi, sehingga dianggap nol. Sedangkan arus yang mengalir di basis transistor sebesar :

$$I_b = \frac{V_B - V_{BE}}{R_B} = \frac{5 - 0,7}{1000 \Omega} = 4,3 \text{ mA}$$

Transistor Off menyebabkan LED mati, karena tak ada arus yang mengalir ke basis transistor sehingga tak ada arus yang mengalir di kolektor transistor. Tegangan di kolektor ke emitor ( $V_{CE}$ ) besarnya sama dengan tegangan supply  $V_{cc}$ . Dalam hal ini transistor beroperasi di daerah cut off di mana ada resistans yang tinggi di antara kolektor dan emitor sehingga tak ada arus yang mengalir di kolektor.

Port B PB7, PB6, PB5, PB4 digunakan sebagai masukan IC 7448 III yang nantinya keluaran IC ini digunakan sebagai penampil desimal sepersepuluhan yang letaknya di belakang koma. Dari gambar B.23 diketahui bahwa di antara IC 7448 dan penampil tujuh segmen ada resistor sebesar  $100 \Omega$ . Resistor  $100\Omega$  ini digunakan sebagai pembagi tegangan dari keluaran IC 7448. Hal ini dilakukan agar LED pada penampil tujuh segmen tidak terbakar karena LED tujuh segmen ini mempunyai batas tegangan yaitu kira-kira 2 Volt. Oleh sebab itu tidak boleh dilampaui.

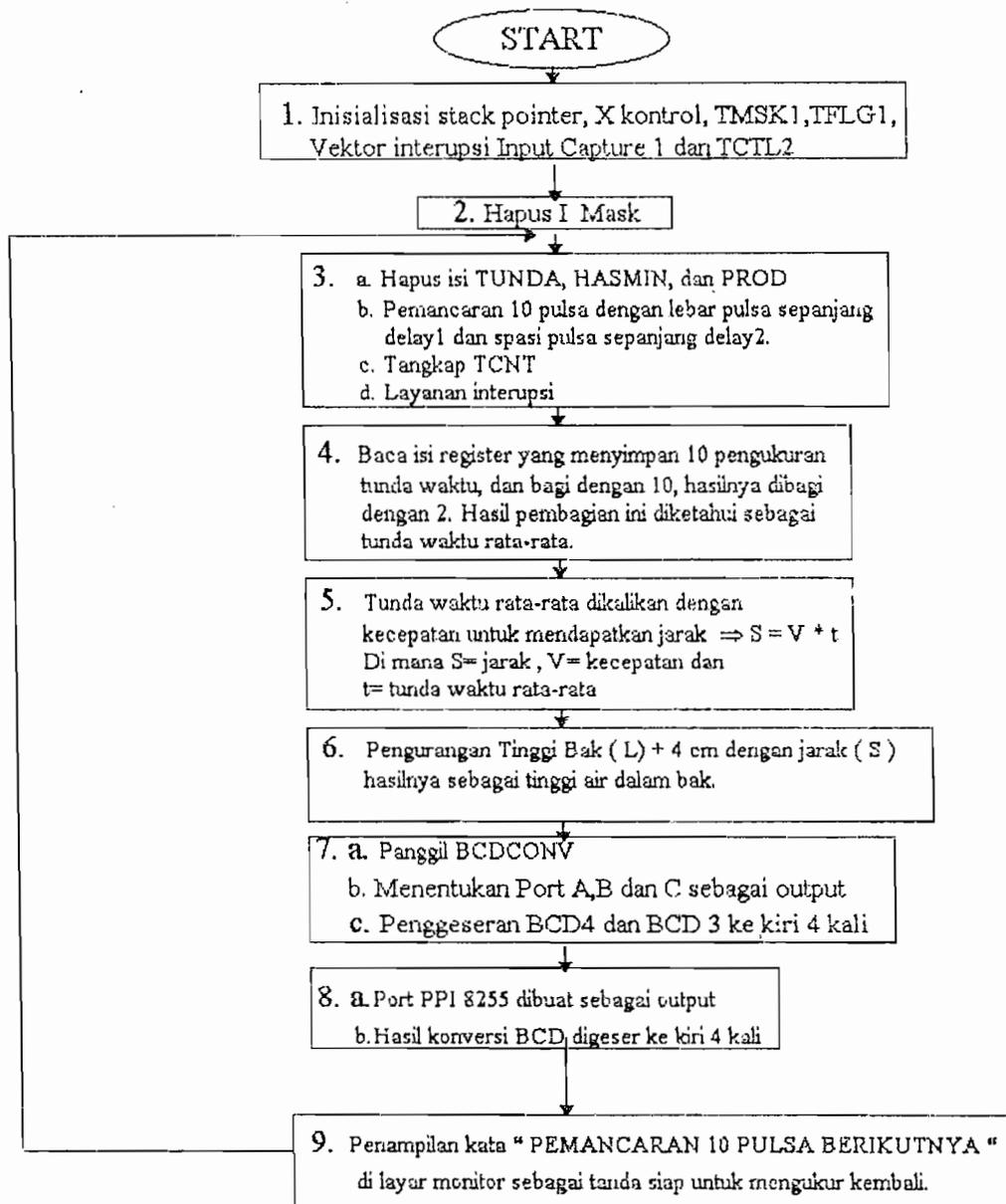
## BAB IV

### PERANGKAT LUNAK

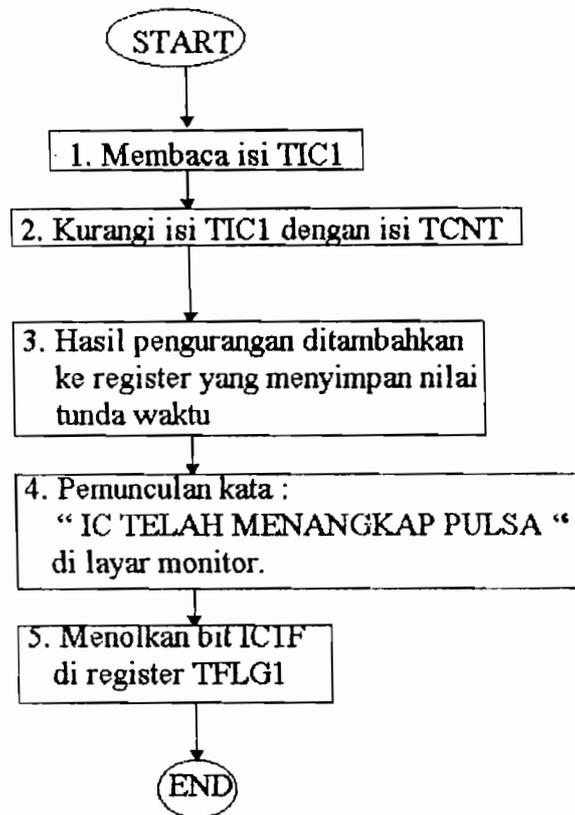
Dalam bab ini akan dibahas mengenai perangkat lunak, biasanya berupa program yang merupakan pusat kendali dari seluruh sistem dalam tugas akhir ini. Diketahui bahwa mikrokontroler dapat bekerja jika diprogram sebelumnya. Program yang digunakan untuk memprogram mikrokontroler M68HC11 menggunakan bahasa assembler.

Jika membahas mengenai perangkat lunak, pasti tidak akan terlepas dari diagram alir. Diagram alir adalah diagram yang digunakan untuk memberikan gambaran sistem secara keseluruhan. Sistem ini dibagi-bagi menurut kegunaannya di setiap kotak dalam diagram alir ini. Dalam tugas akhir ini, perangkat lunak digunakan untuk pengendalian, proses aritmatika, konversi biner ke BCD dan penampilan hasil pengukuran. Diagram alir dalam tugas akhir ini terdiri dari blok-blok. Pembahasan diagram alir ini, akan dibahas perblok dengan tujuan agar pembahasan lebih mendalam dan jelas.

Sebelum lebih lanjut membahas tentang program, diberikan diagram alir (*flow chart*) seperti gambar C.1. Gambar ini adalah gambar diagram alir dari sistem secara keseluruhan. Jadi merupakan diagram alir utama. Dalam pembahasan perblok dari diagram alir utama ini, akan diberikan diagram alir tersendiri.



Gambar C.1 Diagram alir utama



**Gambar C.2 Diagram alir layanan interupsi Input Capture 1**

Akan dijelaskan diagram alir program utama di atas, dengan cara penjelasan tiap blok. Pembahasan mulai dari blok 1.

### **BLOK 1**

Dalam blok ini, pokok dasar program adalah memberikan inisialisasi register-register yang diperlukan antara lain :

1. Menginisialisasi stack pointer dengan alamat \$1FF
2. Menginisialisasi X kontrol, dimana register X dimasukkan \$1000

3. Menginisialisasi Vektor Interupsi Input Capture 1 dimana terletak di lokasi memori \$00E8 - \$00EA yang tinggal di ROM. Vektor Interupsi Input Capture 1 ini digunakan untuk layanan interupsi yang terletak pada alamat AYO setelah input capture menangkap pulsa dari pengirim. Dalam mengawali Vektor Interupsi Input Capture 1 ini, letakkan \$7E ( JMP ) di lokasi \$00E8. Sehingga jika ada perintah layanan interupsi ini, program langsung melompat ( JMP ) ke alamat layanan interupsi dan melayani program-program yang ada di dalamnya , lihat gambar C.2.
4. Menginisialisasi TCTL2 agar Input Capture 1 menangkap pada pinggiran naik.
5. Menginisialisasi TMSK1 dimaksudkan agar Input Capture dapat melakukan interupsi. Dengan menuliskan “ 1 “ pada bit IC1I maka interupsi di Input Capture 1 dimungkinkan.
6. Menginisialisasi TFLG1 dengan cara menolak isi bit IC1F. Hal ini dimaksudkan agar Input Capture 1 dapat ditangkap. Flag ini set ketika Input Capture 1 telah menangkap pulsa pada pinggiran naik ( sesuai dengan isi register TCTL2 ).

## BLOK 2

Blok ini menjelaskan tentang penghapusan bit I mask yang ada di CCR. Hanya dengan perintah CLI maka hal tersebut dapat dilaksanakan. Penghapusan ini dilakukan agar interupsi secara menyeluruh diperbolehkan.

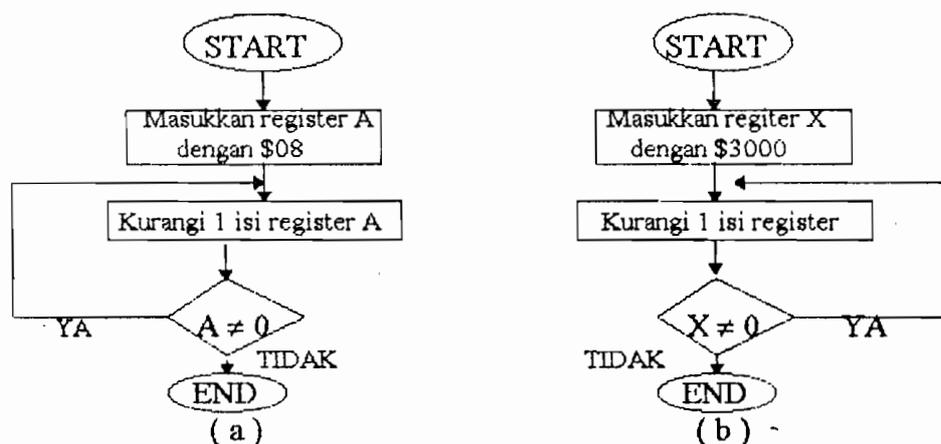
S	X	H	I	N	Z	V	C
---	---	---	---	---	---	---	---

**Gambar 4.3 Register CCR**

### BLOK 3

Di blok ini terjadi beberapa proses yaitu :

1. Penghapusan memori yang ada di alamat *TUNDA* yang berisi penyimpanan tunda waktu. Hal ini dimaksudkan agar isi *TUNDA* dapat diisi dengan isi tunda waktu yang baru. Jika tidak dihapus akan mengganggu proses perhitungan, karena isi yang lama akan ditambahkan dengan isi yang baru.
2. Penghapusan isi yang ada di alamat *PROD MS, MIDH, MIDLS, LS* dan *HASMIN 1, 2, 3, 4*. Hal ini dilakukan untuk menjaga isi yang lama tidak terbawa dalam pengukuran berikutnya.
3. Pemancaran 10 pulsa dalam setiap kali pemancaran. Hal ini dilakukan untuk rata-rata pengukuran.
4. Pemancaran tiap pulsa. Pulsa dipancarkan dengan lebar pulsa sepanjang *Delay 1* dan mempunyai spasi pulsa sepanjang *Delay 2*. Diagram alir dari *Delay 1* dan *Delay 2* dapat dilihat di bawah ini :



Gambar C.4 Diagram Alir ( a ) Delay1 dan ( b ) Delay2

Pulsa - pulsa ini dipancarkan melalui pin PA6 dari PORTA yang beralamat di \$1000. Telah diketahui pada BAB III bahwa pulsa dari pin PA6 ini dihubungkan ke pin reset dari timer 555 yang berfrekuensi 40 KHZ.

Marilah kita analisa Delay1 dan Delay 2. Analisa kita mulai dari Delay1. Program Delay1 adalah sebagai berikut :

```
LDAA #$08
YU DECA
BNE YU
RTS
```

Delay1 ini mempunyai waktu sebesar  $8 \cdot 6 \cdot (500 \cdot 10^{-9}) = 24 \mu\text{s}$  atau mempunyai frekuensi sebesar 41.666 HZ. Nilai 8 diambil dari \$08. Nilai 6 berasal dari eksekusi DECA dan BNE YU yang membutuhkan eksekusi sebanyak 6 kounter di mana setiap kounternya bernilai  $500 \cdot 10^{-9}$ .

Delay 2 mempunyai bentuk diagram alir yang sama dengan Delay2. Delay 2 ini mempunyai waktu kira-kira  $12288 \text{ (heksadesimal 3000)} \cdot 6 \cdot 500 \cdot 10^{-9} = 0,0368$  detik atau 36,8 mili detik.

Spasi pulsa yang mempunyai delay sebesar 36,8 mili detik ini sangatlah cukup untuk pemberian spasi antar pulsa.

5. Penangkapan isi *TCNT* saat pinggiran naik setiap pulsa yang keluar pada pin PA6.
6. Setiap pulsa yang dipancarkan pasti akan diterima oleh pin IC1 (*Input Capture 1*) pada pinggiran naik pulsa yang diterima (ditentukan oleh TCTL2). Saat

menangkap pulsa di pin IC1, mikrokontroller mengeksekusi suatu layanan interupsi, di mana diagram alirnya dapat dilihat pada *diagram alir layanan interupsi*.

Dari diagram alir ini, isi layanan interupsi mikrokontroller dapat dijelaskan sebagai berikut :

- a. Setelah menerima pinggirannya naik pulsa di pin IC1, mikrokontroller menangkap isi TIC1 ( *Timer Input Capture 1* ).
  - b. Isi TIC1 ini dikurangkan dengan isi TCNT yang didapat saat pulsa dikeluarkan pada pin PA6. Isi TCNT ini dalam program disimpan di alamat ISITCNT. Hasil pengurangan ini disimpan di alamat RIN. Isi di alamat RIN merupakan isi tunda waktu setiap pulsa.
  - c. Telah diketahui bahwa setiap pemancaran terdiri dari 10 pulsa. Oleh karena itu, setiap pemancaran akan didapat 10 kali tunda waktu yang akan disimpan di alamat TUNDA, di mana setiap pulsa tunda waktunya disimpan di alamat RIN.
- Maka dapat diringkas bahwa isi di alamat TUNDA merupakan pertambahan isi yang ada di alamat RIN dalam 10 pulsa atau dalam 1 kali pemancaran. Sehingga setiap layanan interupsi, isi RIN selalu ditambahkan dengan isi TUNDA, di mana sebelum pemancaran isi TUNDA akan dihapus terlebih dahulu.
- d. Untuk mengetahui apakah Input Capture telah menangkap atau belum, pada layanan interupsi ini diberikan layanan penampilan kalimat di layar monitor. Kalimat tersebut adalah : " IC TELAH MENANGKAP PULSA ". Untuk

melakukan penampilan ini, mikrokontroller memanggil subroutine yang namanya LOOP, di mana di alamat ini program ditulis sebagai berikut :

```
FCC ' IC TELAH MENANGKAP PULSA '
FCB $04
RTS
```

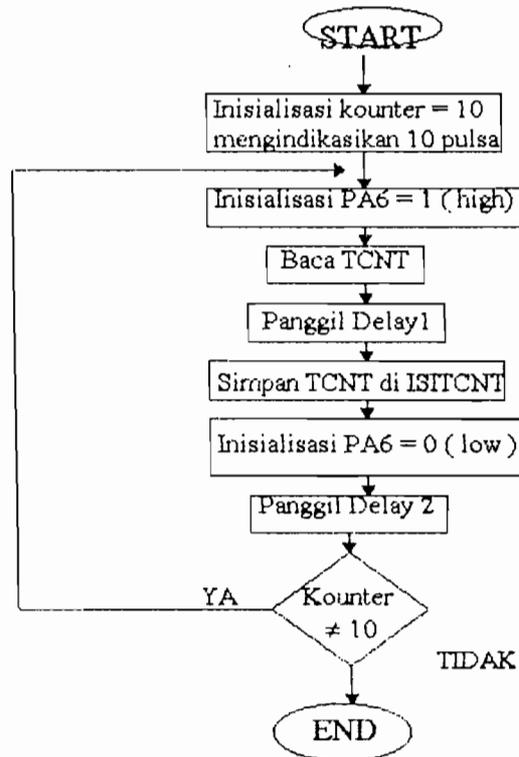
Perintah FCC berarti memasukkan kalimat dalam bentuk ASCII. Perintah FCB menandakan suatu kalimat telah selesai ditulis dengan pemberian \$04. Akhir subrutin ditulis dengan RTS (*Return From Subroutine*). Mikrokontroler kemudian memanggil rutin yang ada di BUFFALO yaitu rutin OUTSTR yang ada di alamat \$FFC7. Dalam rutin ini berisi penampilan dalam bentuk ASCII yang ditunjuk pada alamat yang dimasukkan di register IX dan akhir transmisi karakter ditandai dengan \$04, akibatnya pada subrutin LOOP diakhiri dengan FCB \$04.

e. Sebelum layanan interupsi ini selesai, isi bit IC1F pada register TFLG1 harus dinolkan terlebih dahulu.

f. Selesai layanan interupsi ditandai dengan RTI (*Return From Interrupt*) yang berarti layanan interupsi selesai dan kembali ke program utama.

7. Jika pemancaran pulsa belum 10 kali, maka akan dilakukan pemancaran pulsa lagi sampai pemancaran 10 terpenuhi.

Dari seluruh penjelasan dalam blok 3 ini dapat diringkas dalam diagram alir sebagai berikut :



Gambar C.5 Diagram alir Blok 3

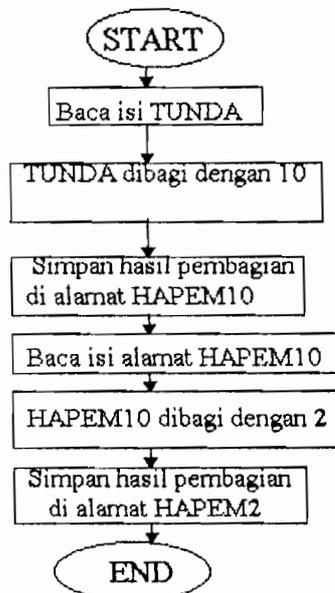
#### BLOK 4

Dalam blok ini, proses rata-rata tunda waktu dilaksanakan. Prosesnya adalah sebagai berikut :

1. Baca isi alamat TUNDA yang berisi tunda waktu setiap 1 kali pemancaran ( 10 pulsa )
2. Untuk mendapatkan rata-rata pengukuran, isi TUNDA ini dibagi dengan 10 dan hasilnya disimpan di alamat HAPEM10.

3. Telah diketahui bahwa setiap pulsa dipancarkan dan diterima akan mempunyai tunda waktu yang terdiri dari waktu antara pemancar dan air ( sebelum dipantulkan air ) dan waktu antara air dengan penerima ( setelah dipantulkan air ). Waktu sebelum dipantulkan dan setelah dipantulkan sama nilainya karena pemasangan ultrasonik sebagai pemancar dan penerima sejajar. Padahal yang diinginkan adalah tunda waktu antara air dan transduser ultrasonik. Oleh karena itu, tunda waktu yang didapat harus dibagi dengan 2. Hasil pembagian ini disimpan di alamat HAPEM2. Isi di alamat HAPEM2 inilah tunda waktu yang diinginkan atau sebagai tunda waktu rata-rata.

Proses dalam blok 4 ini dapat diilustrasikan dalam diagram alir seperti di bawah ini :



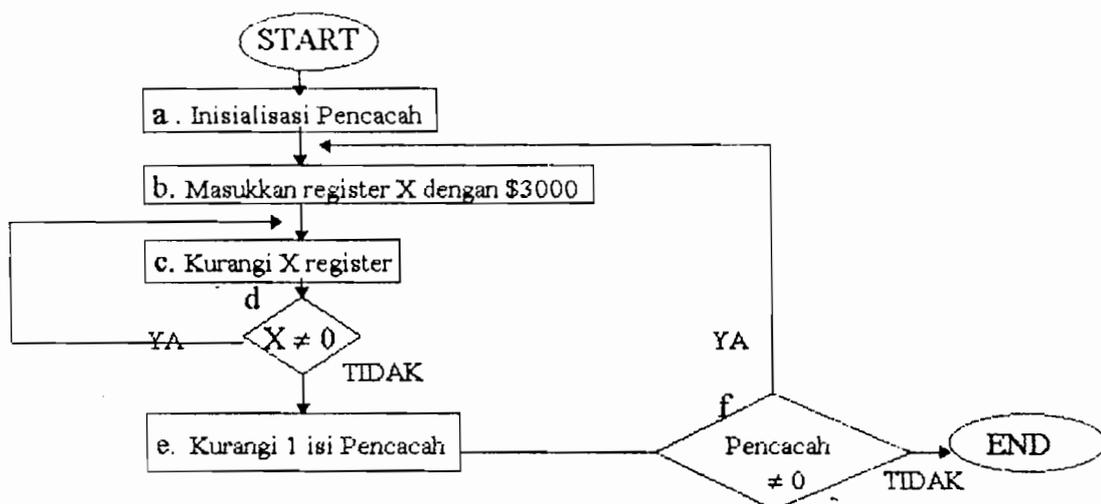
Gambar C.6 Diagram alir Blok 4

## BLOK 5

Dalam blok ini terjadi proses- proses sebagai berikut :

1. Hapus isi PPIC dari PPI 8255. Penggunaan PPIC berhubungan dengan LED yang mengindikasikan sebagai koma. LED ini akan hidup jika isi dari salah satu bit di PPIC yang berhubungan dengan LED bernilai " 1 " dan akan mati jika bernilai "0 ". Penjelasan tentang PPIC akan dijelaskan berikutnya pada blok 8.

Perlu diberi delay setelah proses penghapusan PPIC. Hal ini dimaksudkan agar penampilan LED ini sesuai keinginan dimana LED akan hidup cukup lama saat penampilan dan akan mati sebentar selama delay dan selama proses pengisian PPIC kembali. Besarnya delay tergantung dari subrutin *Delay* yang dipanggil. Subrutin ini pada dasarnya sama dengan subrutin *Delay1* dan *Delay2*, bedanya di subrutin *Delay* yaitu adanya pelipatgandaan waktu yang diinginkan. Untuk lebih jelasnya dapat dilihat pada diagram alir seperti di bawah ini :



**Gambar C.7 Diagram alir Delay**

Dari diagram alir di atas, blok b, c, d dan e sama dengan diagram alir Delay1 dan Delay2. Dari blok b, c, d, dan e ini diketahui bahwa waktu yang didapat adalah  $3000 (12288 \text{ desimal}) \cdot 6 \cdot (500 \cdot 10^{-9}) = 0,0368 \text{ detik}$ . Namun dari blok a, e, dan f menginginkan waktu 0,0368 detik ini dilipatgandakan sebanyak isi kounter yaitu 2 kali. Sehingga waktu yang didapat adalah  $2 \cdot 0,0368 \text{ detik} = 0,0736 \text{ detik}$ .

2. Proses pendapatan jarak antara transduser ultrasonik dengan air ( S ), di mana proses pendapatannya dengan cara mengalikan kecepatan suara ultrasonik ( V ) dengan tunda waktu setiap pemancaran (  $\Delta t$  ) yang disimpan di alamat HAPEM2.

Proses di atas memenuhi persamaan :  $S = V \cdot \Delta t$

Nilai kecepatan suara ultrasonik memenuhi persamaan sebagai berikut :

$$V = 331 + \{ 0,6 \cdot X^\circ (\text{ dalam celcius } ) \}$$

Oleh karena itu nilai kecepatan suara ultrasonik besarnya tergantung dari besarnya suhu saat digunakan. Suhu yang digunakan dalam tugas akhir ini adalah  $27^\circ \text{ C}$ , sehingga didapat nilai  $V = 347 \text{ m/s}$  dan jika diubah ke heksadesimal nilai V menjadi  $15B$ . Jika akan melakukan perkalian dengan berbasis heksadesimal maka hasilnya akan heksadesimal juga. Padahal dalam penampil nanti hasilnya harus dalam bentuk desimal. Oleh karena itu untuk mengubahnya ke desimal, perlu dikalikan dengan satu kounter mikrokontroller yaitu 1 kounter =  $500 \cdot 10^{-9}$  detik. Perkalian ini sangat sulit untuk dilakukan, oleh sebab itu akan diambil angka 5 dari 500 ns ini agar perhitungannya bisa dilaksanakan. Akhirnya kecepatan ini dikalikan dengan 5 sehingga nilainya menjadi  $V \cdot 5 \text{ h} = 15B \text{ h}$   $\cdot 5 \text{ h} = 6C7\text{h}$ .  $5\text{h}$  ini mengindikasikan nilai kounter dalam

mikrokontroller. 6C7h ini dalam program dimasukkan sebagai MULTIPLR, sedangkan isi HAPEM2 dimasukkan dalam MULTICND ( 2 byte ).

Proses perkalian MULTIPLR dan MULTICND adalah sebagai berikut :

$$\begin{array}{r}
 \begin{array}{cc}
 MS & LS \\
 XX & XX \\
 XX & XX \\
 \hline
 \end{array} & \begin{array}{c}
 (MULTICND) \\
 (MULTIPLR)
 \end{array} & \times \\
 \begin{array}{cc}
 MS & LS \\
 XX & XX \\
 \hline
 \end{array} & (P1) & \\
 \begin{array}{cc}
 MS & LS \\
 XX & XX \\
 \hline
 \end{array} & (P2) & \\
 \begin{array}{cc}
 MS & LS \\
 XX & XX \\
 \hline
 \end{array} & (P3) & \\
 \begin{array}{cc}
 MS & LS \\
 XX & XX \\
 \hline
 \end{array} & (P4) & \\
 \hline
 \begin{array}{cccc}
 MS & MIDH & MIDL & LS \\
 XX & XX & XX & XX \\
 \hline
 \end{array} & & + & (PROD)
 \end{array}$$

### Penjelasan

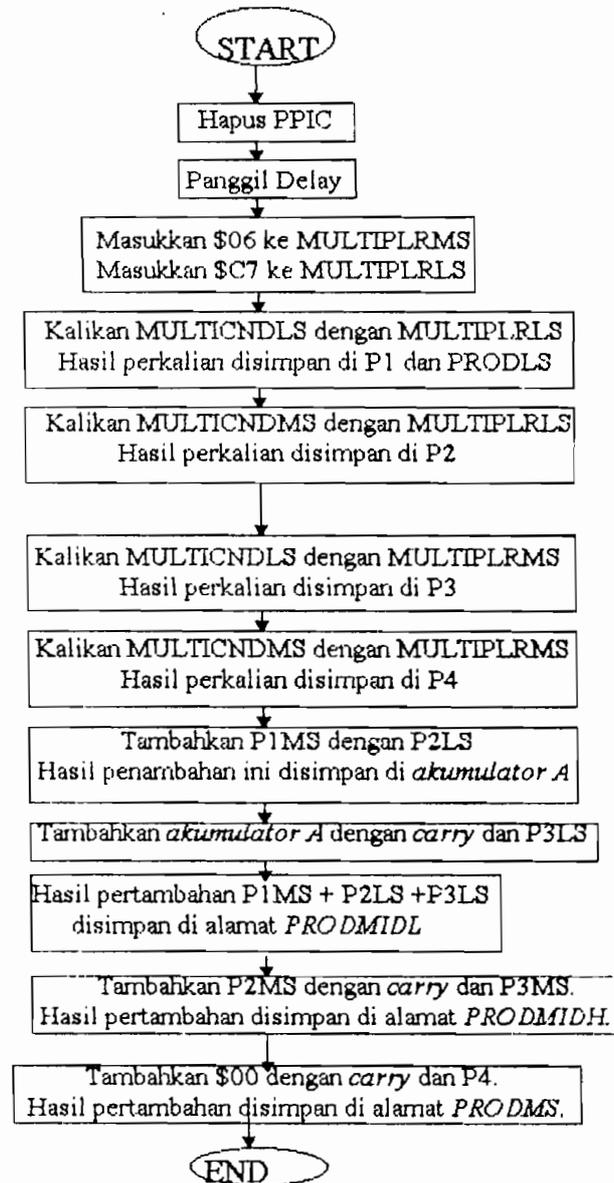
#### A. Proses Perkalian

- a) Hasil P1 merupakan perkalian antara MULTICNDLS dengan MULTIPLRLS
- b) Hasil P2 merupakan perkalian antara MULTICNDMS dengan MULTIPLRLS
- c) Hasil P3 merupakan perkalian antara MULTICNDLS dengan MULTIPLRMS
- d) Hasil P4 merupakan perkalian antara MULTICNDMS dengan MULTIPLRMS

#### B. Proses Pertambahan

- a) PRODLS adalah P1LS
- b) PRODMIDL merupakan jumlah dari pertambahan ( P1MS + P2LS + P3LS )
- c) PRODMIDH merupakan jumlah dari pertambahan ( P2MS + P3MS + P4LS )
- d) PRODMS adalah P4MS

Penjelasan blok 5 ini dapat dibaca secara grafis dengan melihat diagram alir seperti berikut ini :



**Gambar C.8 Diagram alir Blok 5**

## Blok 6

Dalam blok ini dijelaskan tentang pengurangan tinggi bak ( L ) dengan jarak ultrasonik dari air ( S ). Hasil pengurangan ini mengindikasikan tinggi air dalam suatu bak( T ). Tinggi bak ( L ) yang digunakan sebenarnya adalah 38 cm, tetapi karena ada selisih 3,984 cm setelah diadakan percobaan (lihat lampiran B ), maka tinggi bak ( L ) yang digunakan dalam program ini adalah 41,984 cm. Jadi persamaannya adalah  $T(\text{cm}) = 41,984 (\text{cm}) - S(\text{cm})$ . S ( jarak ultrasonik dengan air ) disimpan di alamat PROD.

41,984 cm ini dalam program dibuat menjadi 4198400 ( desimal ) atau jika dibuat heksadesimal menjadi \$00401000. Mengapa 41,984 cm ini harus dibuat 4198400( desimal ) ? Hal ini dikarenakan isi PROD ( jarak antara ultrasonik dengan air ) juga besar. Untuk lebih jelasnya dapat dijelaskan seperti berikut .

Misalnya tinggi air = 38 cm, tunda waktu yang didapat adalah :

$$\begin{aligned} t = S / V &\Rightarrow S = 0,38 (\text{ m }) / ( 331 + ( 0,6 \cdot 27 ) (\text{ m/s }) \\ &= 0,38 / 347 \\ &= 0,00109 \text{ detik} = 1,09 \text{ milidetik} \end{aligned}$$

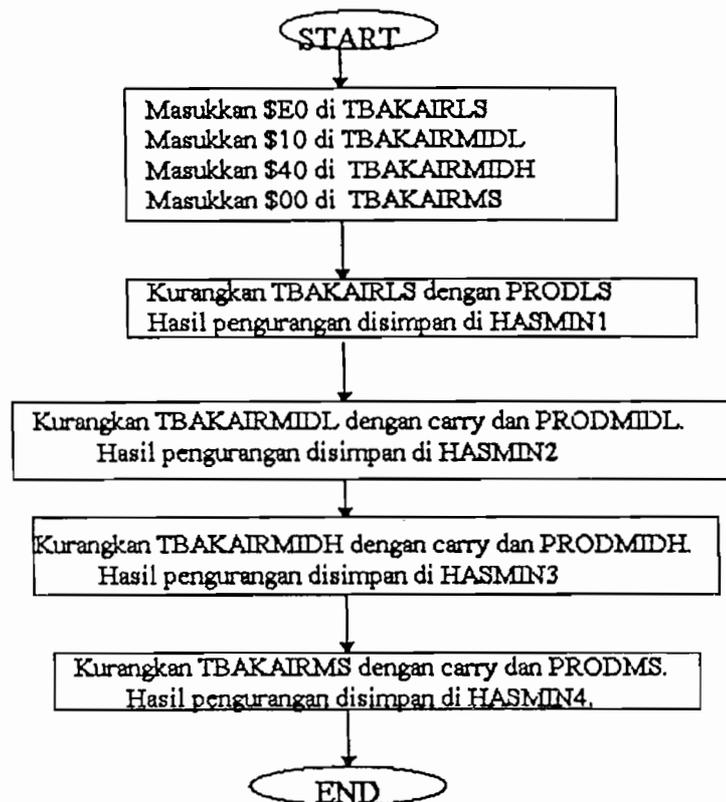
Untuk mengubah nilai t ( detik ) ini menjadi bilangan heksadesimal, maka nilai t ini harus dibagi dengan nilai clock / kounter yang besarnya 500 nano detik yaitu :

$$(1,09 \cdot 10^{-3}) / ( 500 \cdot 10^{-9} ) = 2180 (\text{ desimal }) = 884 (\text{ heksadesimal })$$

Maka jika dihubungkan dengan blok 5 dimana  $V = 6C7h$  maka akan didapat :

$$\begin{aligned} S = V \cdot t &\Rightarrow S = 6C7h \cdot 884h \\ S &= 39E130h \end{aligned}$$

Jika nilai S sebagai pengurang saja nilainya besar, maka nilai tinggi bak ( L ) harus dibuat besar juga. Besarnya nilai tinggi bak ( L ) disimpan di alamat TBAKAIR yaitu \$00401000. Untuk lebih jelasnya penjelasan di blok 6 ini, dapat dilihat diagram alir dibawah ini :



**Gambar C.9 Diagram alir Blok 6**

## **BLOK 7**

Dalam blok ini terjadi beberapa proses antara lain :

### 1. Pemanggilan Subrutin BCDCONV

Perlu diketahui bahwa jika berhubungan dengan mikrokontroller M68HC11 dan melakukan suatu aritmatika, maka akan berhubungan dengan bilangan

heksadesimal. Pada blok 6 telah didapat tinggi air dalam bak yang disimpan di alamat HASMIN. Isi di alamat HASMIN ini akan ditampilkan di penampil Tujuh Segmen. IC 7448 ini mengubah bilangan BCD menjadi bilangan yang bisa ditampilkan di penampil ini. Oleh sebab itu diperlukan subrutin yang mengubah bilangan heksadesimal menjadi bilangan BCD. Subrutin ini adalah subrutin BCDCONV. Untuk lebih jelasnya dapat dilihat diagram alir dari subrutin BCDCONV seperti terlihat di gambar C.10.

Sebelum penjelasan diagram alir dari BCDCONV ini, akan dijelaskan gambaran utama dari proses konversi ini.

Bilangan biner biasanya diekspresikan dalam bentuk :

$$NB = b_{(n-1)}, b_{(n-2)}, \dots, b_1, b_0 \dots\dots\dots(11)$$

di mana  $n$  adalah jumlah bit dan  $b$  adalah bilangan biner. Bilangan desimal  $ND$  dari  $NB$  adalah :

$$ND = b_{(n-1)} \cdot 2^{(n-1)} + b_{(n-2)} \cdot 2^{(n-2)} + \dots\dots\dots + b_1 \cdot 2^1 + b_0 \cdot 2^0 \dots\dots\dots(12)$$

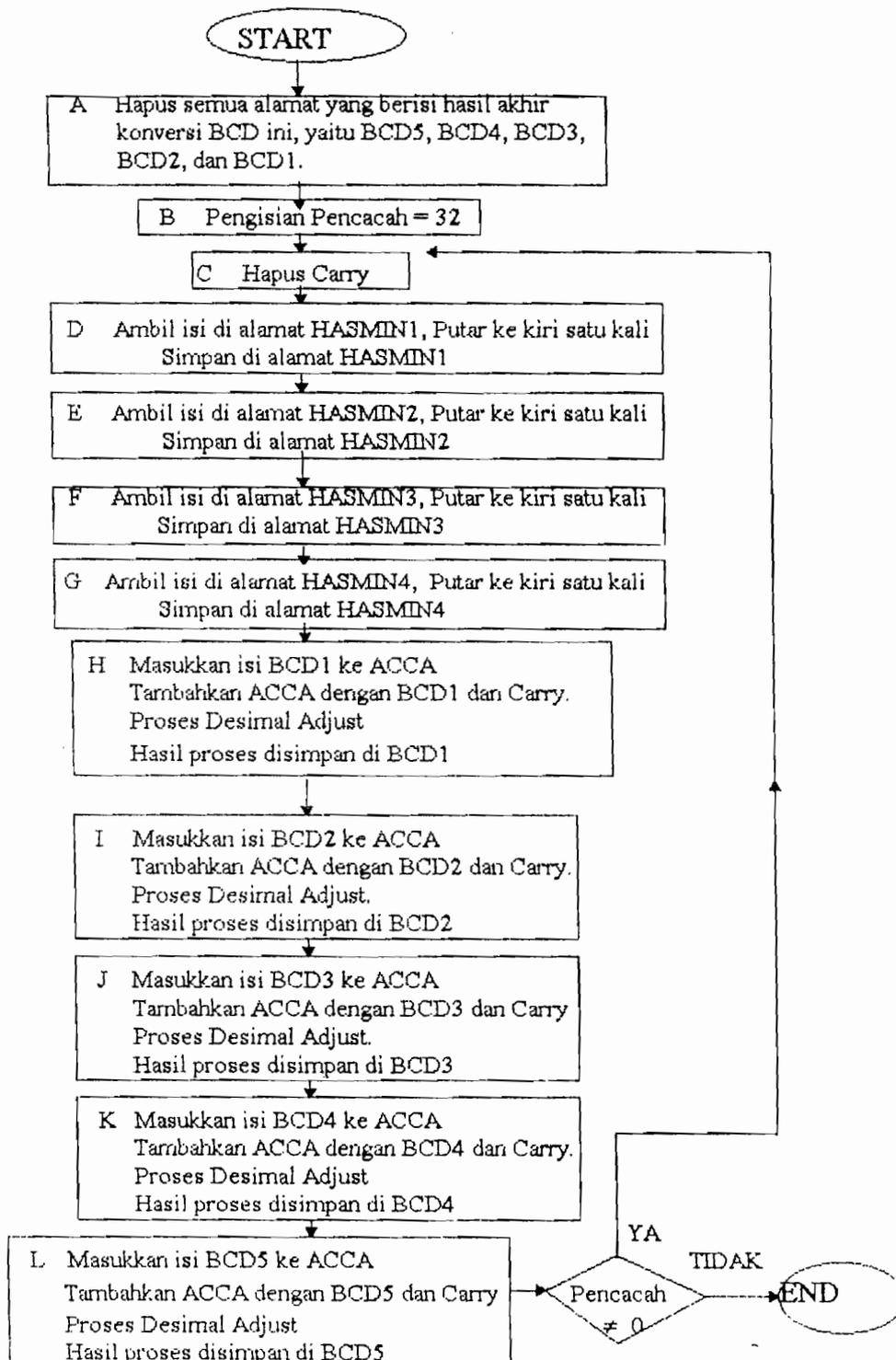
$ND$  ini dapat ditulis dalam bentuk perkalian dengan 2 tanpa adanya pangkat seperti di atas. Misal nilai desimal  $ND$  dari 5 bit biner secara normal dapat ditulis:

$$ND = b_4 \cdot 2^4 + b_3 \cdot 2^3 + b_2 \cdot 2^2 + b_1 \cdot 2^1 + b_0 \dots\dots\dots(13)$$

Jika diubah dalam bentuk perkalian dengan 2 tanpa pangkat adalah sebagai berikut :

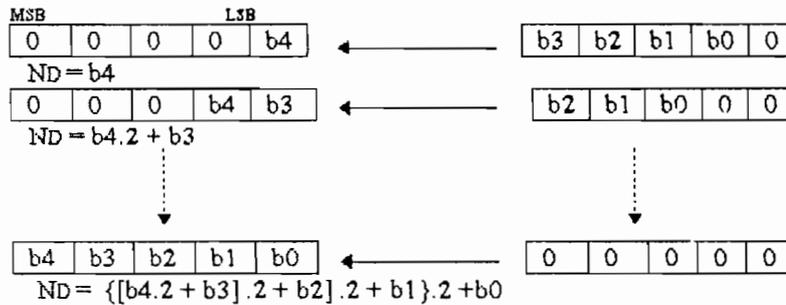
$$ND = \{ [ b_4 \cdot 2 + b_3 ] \cdot 2 + b_2 \} \cdot 2 + b_1 \} \cdot 2 + b_0 \dots\dots\dots(14)$$

Perkalian dengan 2 dapat diartikan menggeser bilangan biner ke kiri sebanyak jumlah bilangan biner tersebut. Jika ada 5 bit bilangan biner maka terjadi 5 kali



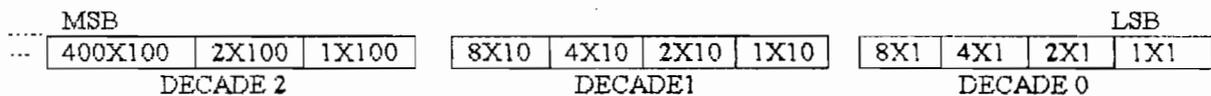
**Gambar C.10** Diagram alir Pengubahan Biner menjadi BCD

penggeseran. Contoh penggeseran 5 bit bilangan biner dapat diilustrasikan pada gambar di bawah ini :



**Gambar C.11 Contoh penggeseran 5 bit ke kiri**

Jika NB ingin diubah menjadi bilangan BCD yang formatnya seperti dibawah ini, maka



setiap dekade nilainya dalam desimal harus lebih kecil atau sama dengan 9.

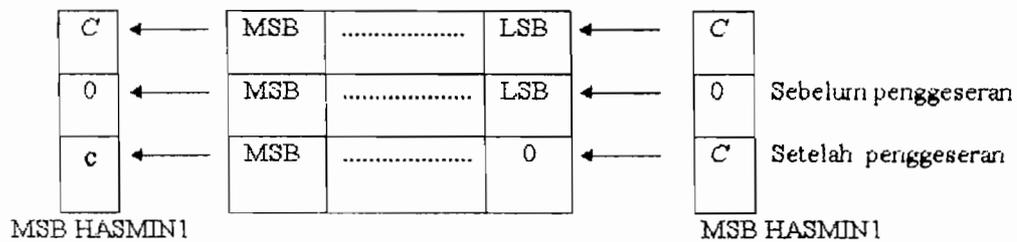
Telah dijelaskan garis besar dari perubahan biner ke BCD. Akan dimulai untuk penjelasan diagram alir BCDCONV di atas. *Blok A* dari diagram alir ini menjelaskan tentang penghapusan isi yang ada di alamat yang merupakan hasil akhir dari BCDCONV ini. Alamat-alamat tersebut adalah BCD5, BCD4, BCD3, BCD2, dan BCD1. BCD5 adalah nilai BCD yang merupakan dekade yang paling besar dan BCD1 adalah nilai BCD yang merupakan dekade yang paling kecil.

*Blok B* menjelaskan tentang penggeseran yang terjadi adalah 32 kali penggeseran karena bit yang akan digeser adalah 32 bit yang ada di alamat HASMIN yaitu HASMIN4, 3, 2, 1. HASMIN4 merupakan alamat yang isinya paling besar ( *most*

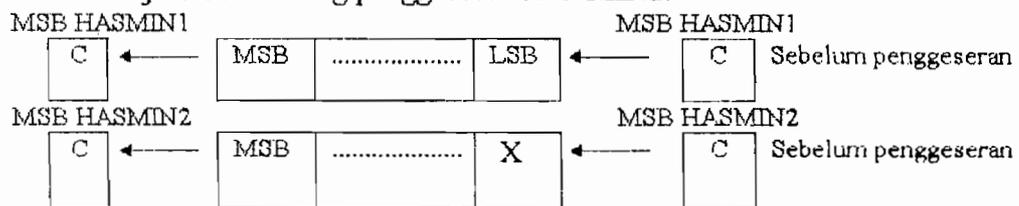
*significant* ) dan HASMIN1 merupakan alamat yang isinya paling kecil nilainya ( *least significant* ). Masing-masing HASMIN ini terdiri dari 8 bit atau 1 byte.

Blok C menjelaskan tentang dihapusnya nilai carry yang ada di CCR.

Blok D menjelaskan tentang penggeseran ke kiri satu kali dari HASMIN1 dan hasil penggeseran disimpan di alamat HASMIN1. Nilai carry setelah penggeseran besarnya tergantung dari nilai MSB dari HASMIN1.

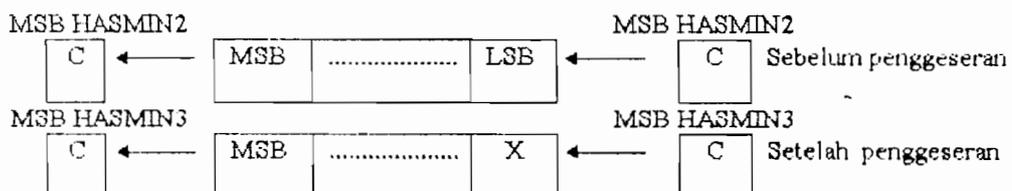


Blok E menjelaskan tentang penggeseran HASMIN2.



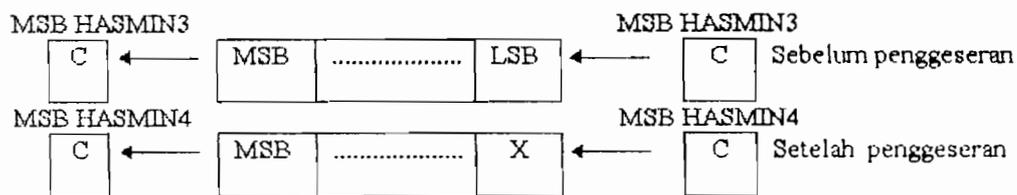
Sebelum penggeseran, nilai carry adalah nilai dari MSB HASMIN1 setelah penggeseran di blok D. Setelah penggeseran, nilai carry sekarang tergantung dari nilai MSB HASMIN2 . LSB HASMIN2 ( X ) nilainya sekarang adalah MSB dari HASMIN1.

Blok F menjelaskan mengenai penggeseran dari HASMIN3



Sebelum penggeseran isi carry adalah nilai dari MSB HASMIN2 setelah penggeseran di blok E. Setelah penggeseran di blok F ini, nilai carry sekarang adalah nilai MSB HASMIN3 dan LSB HASMIN3 ( X ) nilainya adalah MSB HASMIN2.

Blok G menjelaskan tentang penggeseran dari HASMIN 4.



Sebelum penggeseran isi carry adalah nilai dari MSB HASMIN3 yang terjadi setelah penggeseran di blok F. Setelah penggeseran di blok G ini, nilai carry sekarang adalah MSB dari HASMIN4 . LSB HASMIN4 isinya adalah nilai MSB HASMIN3.

Blok H menjelaskan tentang penambahan  $BCD1 + BCD1 + \text{carry}$ . Isi carry adalah nilai MSB HASMIN4. Hasil penambahan ini disimpan di alamat BCD1. Mengapa BCD1 harus ditambahkan dengan BCD1 dalam blok ini ? Hal ini dikarenakan agar penggeseran benar-benar tepat yaitu dari HASMIN1 ke BCD1.

Ambil contoh : Jika isi BCD1 sebelumnya adalah 0001 dan isi carry adalah 1, maka jika ditambahkan  $BCD1 + \text{carry}$ , hasilnya akan 0010. Hasil ini tidak mencerminkan penggeseran dari HASMIN4 ke BCD1. Oleh sebab itu, 0010 harus ditambahkan dengan 0001 (isi dari alamat BCD1) agar penggeseran tercapai yaitu 0011. Agar isi BCD1 ini senantiasa selalu mempunyai kandungan nilai BCD maka isi di BCD1 ini harus di buat

desimal dengan desimal adjust DAA ( *Desimal Adjust Accumulator* ). Hasil dari proses di blok G ini disimpan di BCD1.

Blok I isinya hampir sama dengan Blok H. Perbedaannya hanya dalam hal carry. Diblok ini carry sama dengan 1, jika proses dalam desimal adjust di blok H menghasilkan carry yang isinya sama dengan 1. Hasil dari proses di blok I ini disimpan di BCD2.

Blok J isinya hampir sama dengan Blok I. Perbedaannya terletak dari mana carry tersebut datang. Carry = 1 jika isi BCD2 setelah di DAA ( blok I ) menghasilkan carry = 1. Hasil dari seluruh proses di blok K ini disimpan di alamat BCD3.

Blok K isinya hampir sama dengan Blok K. Perbedaannya terletak dari mana carry tersebut datang. Carry = 1 jika isi BCD3 setelah di DAA ( blok J ) menghasilkan carry = 1. Hasil dari seluruh proses di blok K ini disimpan di alamat BCD4.

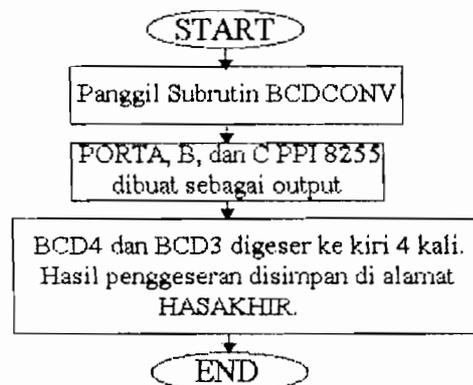
Blok L isinya hampir sama dengan Blok K. Perbedaannya terletak dari mana carry tersebut datang. Carry = 1 jika isi BCD4 setelah di DAA ( blok K ) menghasilkan carry = 1. Hasil dari seluruh proses di blok L ini disimpan di alamat BCD5. Blok M menjelaskan apakah penggeseran isi HASMIN sudah digeser semua sehingga isi HASMIN nilainya nol semua. Jika belum, maka proses penggeseran kembali dijalankan, dengan melompat ke blok C. Jika sudah, maka kembali ke program utama yang ditandai dengan instruksi RTS.

2. Menentukan PPI 8255 yaitu Port A, B, dan C sebagai output semua. Hal ini dilakukan dengan pemberian \$80 ke PPI register kontrol yang ada di alamat \$6003.

3. Proses penggeseran 4 kali terhadap akumulator D. Isi akumulator D adalah BCD4 dan BCD3 di mana BCD4 berada di akumulator A dan BCD3 berada di akumulator B. Penggeseran ini dilakukan karena isi BCD4 nilainya terletak di 4 bit pertama dari akumulator A.

Contoh : BCD4 isinya \$03 dan BCD3 isinya \$42 maka dalam akumulator D sebelum penggeseran isinya adalah \$0342. Setelah digeser 4 kali maka isi akumulator D menjadi \$3420. Hasil penggeseran ini disimpan di alamat HASAKHIR. Isi di alamat HASAKHIR merupakan isi yang akan ditampilkan di penampil tujuh segmen.

Dari seluruh blok 7 ini, jika dibuat diagram alirnya akan menjadi seperti berikut :



Gambar C.12 Diagram alir Blok 7

## BLOK 8

Dalam blok ini dijelaskan tentang penampilan hasil pengukuran di penampil tujuh segmen. Hasil pengukuran yang akan ditampilkan terletak di alamat HASAKHIR. Isi alamat HASAKHIR berkapasitas 2 byte, di mana isi high byte ditampilkan melalui

PORTA dan isi low byte ditampilkan melalui PORTB PPI 8255. Keluaran PORTA PPI 8255 terletak disebelah kanan koma dari penampil tujuh segmen. Sedangkan keluaran PORTB PPI 8255 terletak di sebelah kanan koma dari penampil tujuh segmen.

Untuk mengindikasikan koma, maka dipasang LED yang terletak di antara tujuh segmen keluaran PORTA dan PORTB PPI 8255. Nyala tidaknya LED sebagai tanda koma ini tergantung dari isi LSB PORTC PPI 8255. Jika berisi \$01 maka LED nyala dan jika berisi \$00, LED mati.

### BLOK 9

Di blok ini dijelaskan tentang proses penampilan kalimat " PEMANCARAN 10 PULSA BERIKUTNYA " di layar monitor. Proses penampilan ini telah dijelaskan di penjelasan di blok 3. Penampilan kalimat ini menandakan pengukuran siap dilakukan lagi yaitu dengan melompat ke blok 3 dari program utama.

### PROSES DOWNLOAD DARI PC KE MIKROKONTROLER

Pengguna M68HC11 dapat menggunakan program monitor yang bernama BUFFALO ( Bit User Fast Friendly Aid to Logical Operations ). Program monitor ini disimpan di dalam ROM MCU. Program monitor BUFFALO digunakan untuk menterjemahkan mnemonik ke dalam bahasa mesin. Biasanya jika kita menggunakan mikrokontroler, program yang kita tulis berupa mnemonik-mnemonik. Mnemonik-mnemonik ini dapat ditulis di PC ( Personal Computer ). Untuk menjalankan



mikrokontroler M68HC11, program yang ditulis di PC yang berupa mnemonik-mnemonik harus diubah menjadi file S-records.

Langkah-langkah membuat program sampai mikrokontroler M68HC11 bekerja, dapat dijelaskan sebagai berikut :

1. Memanggil file Edit atau E yang digunakan untuk menulis mnemonik yang merupakan kode instruksi yang membentuk suatu program. Jika program yang ditulis sudah selesai, maka disimpan dengan ekstension ASM.
2. Untuk membuktikan apakah program yang ditulis sudah benar, dalam arti sudah sesuai dengan peraturan bahasa assemblernya, maka program harus ditulis dengan ekstension LST. Cara pembuatan file ini adalah sebagai berikut :

**A > AS11 Nama File. ASM -L>Nama File.LST**

AS11 adalah nama sebuah assembler M68HC11 yang digunakan untuk mengecek benar tidaknya program yang kita buat dengan cara membuat file baru dengan nama file yang sama dari program yang kita buat dengan ekstension LST. Jadi perintah di atas merupakan proses pembuatan file dengan ekstension LST. Selain file LST yang dibentuk dengan perintah di atas , dibentuk juga file dengan ekstension S19 dengan nama file yang sama. File dengan ekstension S19 ini merupakan S-record yang akan di download ke M68HC11. Bentuk file dengan ekstension LST dan S19 dapat dilihat di lampiran B dan C.

3. Setelah tidak ada kesalahan pada program yang kita buat maka program tersebut harus dimasukkan ke mikrokontroller dengan cara mendownload file dengan ekstension S19 . Sebelum download kita harus memanggil file PROCOMM.EXE. File PROCOMM ini adalah file yang digunakan untuk download file dengan ekstension S19 ( S-records ) ke M68HC11. File ini digunakan sebagai file awal penggunaan BUFFALO. Cara - cara mendownload adalah sebagai berikut :

```
C > procomm ↵
```

Perintah di atas bertujuan untuk masuk ke BUFFALO. Setelah menjalankan file PROCOMM, maka di layar akan muncul tanda prompt ( > ) yang artinya BUFFALO sudah bisa digunakan. Dengan munculnya prompt maka proses download bisa dimulai dengan cara sebagai berikut :

```
> LOAD T ↵
```

Setelah instruksi di atas, tekan *tombol page up* pilih nomor 7 yaitu *ASCII*. Dengan penekanan nomor 7 ini proses download dilaksanakan.

4. Setelah proses download selesai, maka muncul kata done yang menandakan transfer S-records selesai dan mikrokontroller siap mengerjakan program yang telah didownload. Perintah mengerjakan program yang telah didownload ditulis setelah prompt dengan perintah sebagai berikut :

```
done  
> G (alamat awal program)
```

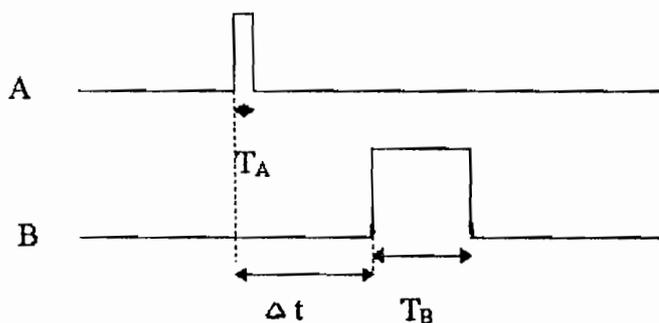
Perintah di atas bertujuan agar mikrokontroler M68HC11 menjalankan program mulai dari alamat awal program. Dalam tugas akhir ini alamat awal dimulai dari alamat \$2000.

## BAB V

### DATA DAN PEMBAHASAN

Dalam bab ini akan dibahas mengenai pembahasan data antara lain data tunda waktu, hasil penampil, arus dan tegangan yang digunakan alat ukur ini, serta kandungan memori program yang dibutuhkan. Data tunda waktu dan hasil penampil ini akan dibandingkan dengan hasil secara teoriti. Dari perbandingan ini diharapkan didapat suatu kesimpulan yang dapat membantu dalam analisa.

Data tunda waktu yang diperoleh dengan cara pengamatan osiloskop. Tunda waktu ini diperoleh dengan cara menghitung periode dari pinggiran naik pulsa yang keluar dari pin PA6 Port A M68HC11 sampai pinggiran naik pulsa yang diterima pin input capture M68HC11. Ilustrasi mengenai tunda waktu dapat dilihat pada gambar di bawah ini :



A = Gambar pulsa yang keluar dari PA6 Port A M68HC11 dengan lebar pulsa  $T_A$  sebesar  $11,5 \mu s$

B = Gambar pulsa yang keluar dari pembanding yang merupakan masukan dari pin input capture M68HC11 dengan lebar pulsa  $T_B$  yang besarnya tergantung dari tingginya air. Nilai  $T_B$  berkisar dari 6 ms sampai 8 ms.

$\Delta t$  = tunda waktu

Hasil pengamatan tunda waktu ini dapat dilihat pada tabel 2.1 di bawah ini.

Tabel ini bukan hanya terdapat tunda waktu tetapi terdapat juga data pengamatan hasil penampil tujuh segmen dan data perolehan tunda waktu secara teoritis.

**Tabel 2.1 Hasil pengamatan dan teoritis tunda waktu**

Tinggi air dalam bak ( Cm )	Hasil Penampil ( Cm )	Tunda Waktu ( $\Delta t / 2$ ) Hasil Pengamatan ( ms )	Tunda Waktu Hasil Teoritis ( ms )	Selisih Tunda Waktu Hasil Pengamatan dan Teoritis ( ms )
0	00,5	1,175	1,095	0,08
1	1,4	1,15	1,066	0,084
2	2,5	1,125	1,037	0,088
3	2,8	1,1	1,008	0,092
4	3,8	1,075	0,97	0,105
5	5,0	1,05	0,95	0,1
6	5,9	1	0,92	0,08
7	7,4	0,975	0,89	0,085
8	8,3	0,95	0,86	0,09
9	9,4	0,925	0,83	0,095
10	10,4	0,9	0,8	0,1
11	12,0	0,875	0,778	0,097
12	13,0	0,85	0,74	0,13
13	14,0	0,825	0,72	0,105
14	14,2	0,8	0,69	0,11
15	15,2	0,76	0,66	0,1
16	16,5	0,74	0,63	0,11
17	17,0	0,72	0,6	0,12
18	18,2	0,7	0,576	0,124
19	19,0	0,68	0,547	0,13
20	20,0	0,64	0,518	0,12

21	20,9	0,62	0,489	0,13
22	21,9	0,58	0,46	0,12
23	23,1	0,56	0,43	0,13
24	24,0	0,54	0,4	0,14
25	25,0	0,52	0,374	0,146
26	26,3	0,48	0,345	0,135
27	27,2	0,46	0,317	0,143
28	28,0	0,44	0,288	0,152
29	29,3	0,42	0,25	0,17
30	30,1	0,38	0,23	0,15
31	31,0	0,36	0,2	0,16

Tunda waktu secara teoritis diperoleh dengan menggunakan rumus  $S = V \cdot t$ , dimana  $S$  adalah jarak ( m ),  $t$  adalah tunda waktu ( detik ) dan  $V$  adalah kecepatan suara diudara yang besarnya tergantung suhu sehingga kecepatan ini mempunyai persamaan yang juga tergantung suhu yaitu  $V = [ 331 + ( 0,6 \cdot T ) ]$  di mana  $T$  adalah suhu dalam celcius. Dalam tugas akhir ini digunakan suhu  $27^\circ \text{C}$  sehingga didapat  $V = 347,2 \text{ m/detik}$ .

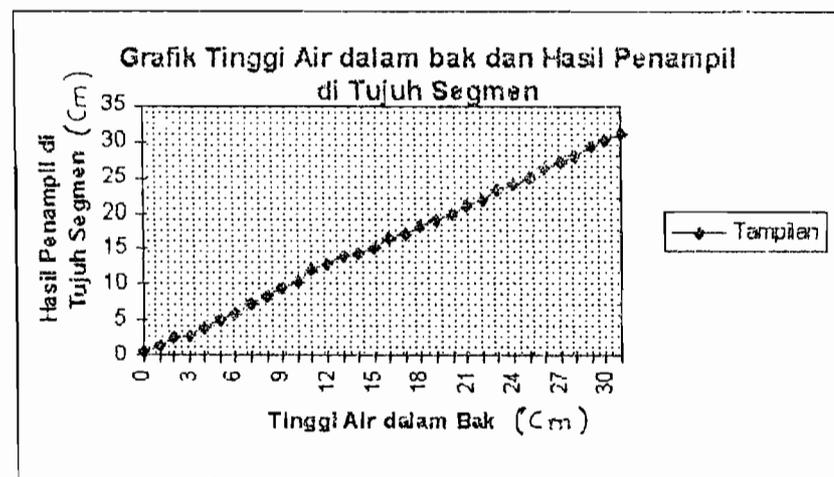
Dari data tabel di atas terlihat selisih tunda waktu hasil pengamatan dengan tunda waktu hasil teoritis pada suhu  $27^\circ \text{C}$ . Rata-rata selisih tunda waktu adalah 0,116 milisekon diperoleh dari pembagian jumlah 32 data selisih tunda waktu dengan 32 data pengukuran. Diketahui tinggi bak 38 cm, dengan tinggi bak sesungguhnya 37 cm ditambah 1cm yang merupakan jarak bak dengan transduser ultrasonik. Tunda waktu hasil pengamatan pada tinggi bak 32 cm sampai dengan 37 cm tidak terbaca artinya tunda waktunya selalu berubah-ubah sehingga menghasilkan hasil penampil pada ketinggian ini selalu berubah-ubah. Hal ini terjadi karena transduser ultrasonik dalam memancarkan pulsa mengalami pemantulan pada banyak tempat di mana pada sisi

penerimaan pada transduser ultrasonik ini daya pemantulannya sama kuatnya. Lain halnya dengan ketinggian di bawahnya di mana pengukurannya menghasilkan hasil penampil yang mendekati kebenaran. Hal ini dikarenakan ultrasonik dalam memancarkan pulsa hanya satu pantulan yang kuat dan merupakan paling kuat dari pantulan yang lainnya yang diterima pada sisi penerima transduser ultrasonik.

Pada ketinggian 32 sampai dengan 37 cm ini sebenarnya bisa diukur asalkan jarak transduser ultrasonik sebagai pemancar dan penerima diperkecil. Hal ini tidak dilakukan karena dengan memperkecil jarak dari kedua transduser ini mengakibatkan sinyal tambahan yang merupakan sinyal bayangan pada sisi penerima transduser ultrasonik. Sinyal bayangan ini berada di depan sinyal sesungguhnya. Sinyal bayangan ini dapat hilang jika jarak kedua transduser minimal 10 cm. Oleh sebab itu jarak kedua transduser ultrasonik ini dipasang 10 cm. Sinyal bayangan ini harus dihilangkan karena akan mengganggu dalam proses pengukuran. Sinyal bayangan ini tidak pernah berubah posisinya pada ketinggian air di mana pun dan amplitudonya juga tetap. Oleh sebab itu harus dihilangkan karena jika tidak dihilangkan, pada sisi penerima sinyal bayangan ini akan dimasukkan dalam penyaring dan ikut dibandingkan pada pembanding sehingga akan menjadi sinyal kotak yang akan dimasukkan pada pin input capture M68HC11. Karena posisi dari sinyal bayangan ini selalu tetap pada ketinggian air di mana pun, akan menghasilkan tunda waktu yang selalu sama pada ketinggian air di mana pun juga sehingga akan menghasilkan hasil pengukuran yang juga selalu sama.

Sekarang kembali lagi ke pembahasan mengenai rata-rata selisih tunda waktu. Tunda waktu secara teoritis adalah perjalanan sinyal dari transduser ultrasonik sebagai pemancar sampai transduser ultrasonik sebagai penerima. Selisih tunda waktu ini, diperoleh dari perjalanan sinyal dari transduser ultrasonik sebagai penerima sampai pin input capture M68HC11 dan dari pin PA6 mikrokontroler sampai keluaran timer 17555. Selisih rata-rata tunda waktu yang diperoleh sebesar 0,116 ms. Nilai sebesar 0,116 ms ini, jika dikalikan dengan kecepatan sebesar 347 m/s mempunyai jarak sebesar 4 cm. Jarak ini ternyata hampir sama dengan penambahan 3,984 cm pada perangkat lunak.

Tabel 2.1 dapat dibuat grafik yaitu antara tinggi air dalam bak dengan hasil penampil yang dapat dilihat pada gambar D.1 di bawah ini :



**Gambar D.1.** Grafik Tinggi air dalam bak dengan Hasil Penampil

Grafik pada gambar D.1 dapat dibuat linear dengan metode regresi linear. Regresi linear pada dasarnya digunakan untuk melinearkan hasil pengukuran yang tertampil di tujuh segmen. Melinearkan berarti membuat garis lurus dari grafik pada gambar D.1. Ungkapan matematis untuk garis lurus adalah

$$y = a_0 + a_1x + E \dots\dots\dots(15)$$

$a_0$  dan  $a_1$  adalah koefisien-koefisien yang masing-masing mewakili perpotongan kemiringan. Sedangkan  $E$  adalah galat atau sisi antara model dan pengamatan, yang dapat dinyatakan dengan persamaan :

$$E = y - a_0 - a_1x \dots\dots\dots(16)$$

Jadi, galat atau sisa adalah ketidaksesuaian antara nilai yang sejati  $y$  dan nilai hampiran  $a_0 + a_1x$ , yang diramalkan oleh persamaan linear.  $a_1$  dapat dicari dengan persamaan sebagai berikut :

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} \dots\dots\dots(17)$$

dan  $a_0$  juga dapat dicari dengan persamaan sebagai berikut :

$$a_0 = \bar{y} - a_1 \bar{x} \dots\dots\dots(18)$$

Selain  $a_0$  dan  $a_1$ , dicari juga  $S_r$ ,  $S_t$  dan  $r$ .  $S_t$  adalah besarnya sebaran dalam peubah tak bebas ( $y$ ) yang ada sebelum regresi.  $S_r$  adalah jumlah kuadrat sisa-sisa di sekeliling garis regresi atau sebaran yang tertinggal setelah regresi. Selisih antara dua besaran ( $S_t - S_r$ ) digunakan untuk mengukur perbaikan atau pengurangan galat

sehubungan dengan model garis lurus. Selisih ini dapat dinormalkan terhadap galat total sebagai berikut :

$$r^2 = \frac{St - Sr}{St} \dots\dots\dots( 19 )$$

$r$  adalah koefisien korelasi dan  $r^2$  adalah koefisien determinasi. Untuk kecocokan yang sempurna,  $Sr = 0$  dan  $r^2 = 1$ , yang menandakan bahwa garis tersebut menerangkan 100% dari ketidaktetapan ( kevariabilisan ). Untuk  $r = r^2 = 0$  dan  $Sr = St$ , kecocokan ini menyatakan tidak ada perbaikan.

Nilai  $St$  dapat didekati dengan persamaan :

$$St = y_i - \bar{y} \dots\dots\dots( 20 )$$

$\bar{y}$  adalah  $y_i$  rata - rata.

$\bar{x}$  adalah  $x_i$  rata -rata.

dan  $Sr$  dapat didekati dengan persamaan :

$$Sr = \sum_{i=1}^n ( y_i - a_0 - a_1 x_i )^2 \dots\dots\dots( 21 )$$

Akan dibuat tabel regresi yang di dalamnya terdapat nilai dari persamaan - persamaan di atas jika  $x_i$  dimasukkan sebagai tinggi air dalam bak ( cm ) dan  $y_i$  dimasukkan sebagai hasil penampil tujuh segmen. Tabel regresi tersebut dapat dilihat pada tabel 2.2 di bawah ini :

Tabel 2.2 Data Perhitungan Regresi

$x_i$	$y_i$	$Sr$ $(y_i - \bar{y})^2$	$Sr$ $(y_i - a_0 - a_1 x_i)^2$
0	0,5	231,6103516	0,108421442
1	1,4	205,0266016	0,051155485
2	2,5	174,7353516	0,104379041
3	2,8	166,8941016	0,144415865
4	3,8	142,0566016	0,146780325
5	5,0	114,8916016	0,034676971
6	5,9	96,40785156	0,083703669
7	7,4	69,20160156	0,043091866
8	8,3	55,03785156	0,010917632
9	9,4	39,92660156	0,040557587
10	10,4	28,28910156	0,039319247
11	12,0	13,82910156	0,632331088
12	13,0	7,391601563	0,627413152
13	14,0	2,954101563	0,622514411
14	14,2	2,306601563	0,000198881
15	15,2	0,269101562	0,000295868
16	16,5	0,610351562	0,078232557
17	17,0	1,641601563	0,04990644

18	18,2	6,156601563	0,000702028
19	19,0	10,76660156	0,052713476
20	20,0	18,32910156	0,054145792
21	20,9	26,84535156	0,112755471
22	21,9	38,20785156	0,114845852
23	23,1	54,48285156	0,020160443
24	24,0	68,57910156	0,060067051
25	25,0	86,14160156	0,061595364
26	26,3	111,9628516	0,002373398
27	27,2	131,8191016	0,002957270
28	28,0	150,8291016	0,066295499
29	29,3	184,4503516	0,0015554137
30	30,1	206,8203516	0,0267899762
31	31,0	233,5166016	0,071168428
$\Sigma 497$	503,3	2681,987501	3,404840344

Dengan data-data yang ada di tabel 2.2 diperoleh :

$$a_1 = \frac{n \sum x_i y_i - \sum x_i \sum y_i}{n \sum x_i^2 - (\sum x_i)^2} = \frac{32 ( 10504,5 ) - 249636,8}{32 ( 10383 ) - 246016}$$

$$a_1 = 1,00309833$$

Sedangkan  $a_0$  diperoleh sebagai berikut :

$$\bar{y} = 503,3 / 32 = 15,71875$$

$$a_0 = \bar{y} - a_1 \bar{x} = 15,71875 - 15,5 \cdot 1,00309833$$

$$a_0 = 0,170725885$$

Setelah didapat  $a_1$  dan  $a_0$ , mudah untuk mendapatkan  $y$  yaitu :

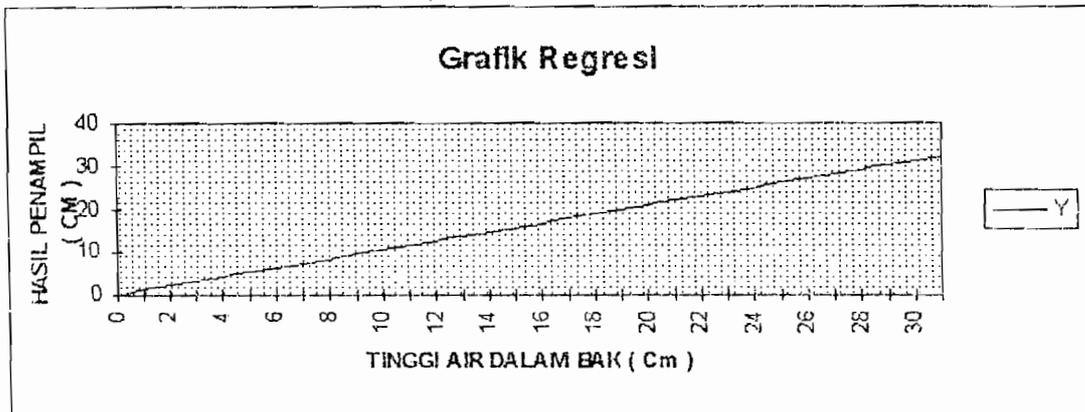
$$y = a_0 + a_1 x = 0,170725885 + 1,00309833 \cdot x \dots\dots\dots(22)$$

Dari persamaan 22 dapat diperoleh tabel yang merupakan regresi dari grafik pada gambar D.1. Tabel tersebut dapat dilihat pada tabel 2.3.

**Tabel 2.3** Data linearisasi dari grafik yang ada di gambar D.1

$x$	$y$	$x$	$y$	$x$	$y$
0	0,170725885	11	11,53480752	22	22,89888915
1	1,203824215	12	12,56790585	23	23,93198748
2	2,236922545	13	13,60100418	24	24,96508581
3	3,270020875	14	14,63410251	25	25,99818414
4	4,303119205	15	15,66720084	26	27,03128247
5	5,336217535	16	16,70029917	27	28,06438080
6	6,369315865	17	17,73339750	28	29,09747913
7	7,402414195	18	18,76649583	29	30,13057746
8	8,435512525	19	19,79959416	30	31,16367579
9	9,468610855	20	20,83269249	31	32,19677412
10	10,50170919	21	21,86579082		

Dari tabel 2.3 di atas dapat di buat grafik regresi seperti pada gambar D.2 di bawah ini :



Gambar D.2 Grafik regresi antara tinggi air dalam bak dan hasil penampil

Dari tabel 2.2, koefisien korelasi ( $r$ ) dapat diperoleh sebagai berikut :

$$r^2 = \frac{St - Sr}{St} = \frac{2681,987501 - 3,404840344}{2681,987501} = 0,998730478$$

sehingga  $r$  diperoleh sebesar 0,999365037.

Alat ukur ketinggian air ini kadang-kadang melakukan pengukuran yang salah.

Hal ini disebabkan oleh beberapa faktor penyebab kesalahan di antaranya :

1. Faktor Permukaan Air

Jika air bergelombang maka dapat menyebabkan kesalahan dalam pengukuran. Hal ini disebabkan karena dalam setiap pemancaran yang terdiri dari 10 pulsa yang dipancarkan, pulsa dipantulkan pada sisi yang berbeda-beda yaitu bisa di puncak gelombang, lembah gelombang atau pada sisi yang lainnya. Hal ini akan

menyebabkan hasil pengukuran lebih besar atau bisa lebih kecil dari hasil pengukuran yang sesungguhnya. Alat ukur ini bisa mengukur dengan baik jika permukaan air tenang tidak bergelombang.

## 2. Faktor Frekuensi Pemancar

Menurut penelitian yang saya lakukan ternyata perubahan frekuensi pemancar dapat mengubah hasil pengukuran. Semakin kecil frekuensi maka hasil penampil semakin besar. Hal ini dikarenakan semakin kecil frekuensi, gelombang AM termodulasi setelah penguat pada bagian penerimaan semakin besar amplitudo dan periodenya. Hal ini menyebabkan gelombang kotak setelah pembanding periodenya semakin besar sehingga tunda waktu yang didapat semakin besar, sehingga menyebabkan hasil penampil semakin besar. Perolehan tunda waktu hasil pengamatan yang ada di tabel menggunakan frekuensi 41.666 Hz (Selama PA6 dalam keadaan aras tinggi). Jika frekuensi pemancar ini di ubah-ubah dapat menyebabkan kesalahan pada hasil pengukuran.

## 3. Faktor Penempatan Transduser Ultrasonik

Jika penempatan transduser ultrasonik pada permukaan bak tidak tepat, maka hasil pengukuran juga tidak tepat. Hal ini disebabkan pemantulan pulsa yang dipancarkan tidak berada pada titik pantul yang tepat. Penempatan transduser ultrasonik pada posisi yang tepat sangatlah sulit dilakukan karena walaupun pada ketinggian tertentu hasil pengukuran tepat tetapi pada ketinggian yang lain hasilnya akan tidak tepat. Faktor inilah yang banyak menyebabkan kesalahan pada alat ukur ini.

#### 4. Faktor Suhu

Faktor suhu ini memang tidak banyak berpengaruh dalam hasil pengukuran jika suhu sekitar tidak berubah secara drastis. Dalam tugas akhir ini, digunakan suhu  $27^{\circ}\text{C}$  karena pada waktu pembuatan alat ukur ini, suhu rata-rata adalah  $27^{\circ}\text{C}$ . Walaupun tidak berpengaruh begitu besar namun merupakan penyebab faktor kesalahan dalam skala kecil.

Dari keempat faktor di atas, ternyata kesalahan banyak ditimbulkan oleh faktor penempatan transduser ultrasonik. Hal ini dapat dilihat pada hasil tabel 2.1 yaitu pada ketinggian 11, 12, dan 13 cm, hasil pengukuran pada penampil naik 1 cm yaitu menjadi 12,13 dan 14 cm. Jika penempatan transduser ultrasonik ini kita geser pada posisi yang tepat tentunya akan menghasilkan pengukuran yang tepat. Hal ini sulit diatasi, oleh sebab itu merupakan salah satu kelemahan dari alat ukur ini. Alat ukur ini selain punya kelemahan, juga mempunyai kelebihan yaitu :

1. Dapat mengukur dengan resolusi 0,1 cm
2. Alat ukur ini tidak berhubungan secara langsung dalam arti tidak menyentuh media ukur yaitu air, tetapi berhubungan secara elektronis dengan penggunaan transduser ultrasonik yaitu yang diletakkan di atas bak.
3. Hasil pengukuran dapat dimengerti dengan jelas karena hasil tampilannya berbentuk digit.
4. Proses pengukuran berlangsung dengan cepat, karena adanya pengendali mikrokontroler.

Untuk mendeteksi apakah 10 pulsa yang dipancarkan setiap pemancaran oleh transduser ultrasonik sebagai pemancar sampai di transduser ultrasonik sebagai penerima yaitu dengan cara melihat layar monitor komputer. Pada layar monitor akan tertampil :

IC TELAH MENANGKAP PULSA  
 PEMANCARAN 10 PULSA BERIKUTNYA

Jika kata “ IC TELAH MENANGKAP PULSA “ telah berjumlah 10 kata, maka pemancaran dikatakan sukses, karena 10 pulsa yang dipancarkan dalam setiap pemancaran, telah diterima oleh transduser ultrasonik sebagai penerima. Kata “ PEMANCARAN 10 PULSA BERIKUTNYA “ menandakan bahwa proses pengukuran telah terjadi dan hasilnya telah ditampilkan, sehingga siap untuk memancarkan 10 pulsa berikutnya.

Alat ukur ini menggunakan konsumsi arus sebesar 0,8 A yang digunakan antara lain untuk mencatu mikrokontroler M68HC11, bagian penerima dan bagian penampil. Tegangan yang digunakan untuk mencatu mikrokontroler, dan penampil adalah 5 Volt. Sedangkan tegangan yang digunakan untuk mencatu rangkaian pengondisi isyarat adalah  $\pm 12$  Volt. Telah diketahui bahwa mikrokontroler dapat bekerja jika sudah diprogram. Program yang digunakan sebagai pengendali alat ukur ini membutuhkan memori sebesar

447 byte yang awal programnya dimulai dari alamat \$2000 dan berakhir di alamat \$21bf. Program berada dalam mikrokontroler selama daya ada. Untuk menghapus program ini dari mikrokontroler dapat dilakukan dengan menekan tombol reset.

## BAB VI

### KESIMPULAN DAN SARAN

Dengan melihat tabel 2.1 pada bab V, dapat disimpulkan bahwa alat ukur ketinggian air ini, dapat digunakan untuk mengukur ketinggian air dengan ketinggian maksimum 31 cm. Alat ukur ini dapat mengukur dengan ketelitian 0,1 cm. Tentunya alat ukur ini tidak sempurna, karena pada jarak-jarak tertentu hasil penampil tidak sesuai yang diinginkan. Hal ini disebabkan oleh faktor-faktor penyebab kesalahan yang telah dijelaskan pada bab V.

Tunda waktu secara praktek hasilnya lebih besar daripada tunda waktu secara teoritis. Hal ini disebabkan karena tunda waktu secara teoritis hanya diukur dari pulsa yang dipancarkan di transduser ultrasonik sebagai pemancar sampai transduser ultrasonik sebagai penerima setelah pemantulan oleh permukaan air. Sedangkan tunda waktu secara praktek diukur dari pinggir naik pulsa yang keluar dari pin PA6 Port A M68HC11 sampai pinggir naik pulsa yang diterima Input Capture 1 M68HC11. Jadi tunda waktu secara praktek lebih besar karena adanya pemancar, pengondisi isyarat, dan kabel-kabel.

Alat ukur ini dapat ditingkatkan jarak pengukurannya yaitu dengan memperbaiki dan mengubah baik pada perangkat keras maupun pada perangkat lunaknya. Pada perangkat keras perlu dikuatkan gain penguatnya jika jarak pengukuran bertambah

besar, perubahan frekuensi cut off-nya di penyaring, dan perubahan kapasitornya di bloking kapasitor. Selain itu perlu diubah penampilnya jika yang ingin ditampilkan lebih banyak digitnya. Pada perangkat lunak, perlu diadakan perubahan penentuan tinggi bak baru, dan perubahan pada subrutin heksadesimal ke BCD.

Demikianlah alat ukur ketinggian air menggunakan transduser ultrasonik dengan pengendalian mikrokontroler M68HC11 dibuat dengan segala kelebihan dan kekurangannya. Perbaikan-perbaikan dari kekurangan alat ukur ini juga telah dijelaskan dengan sebaik-baiknya.

## DAFTAR PUSTAKA

- Malik , N.R, ( 1995 ), *Electronic Circuits*, New Jersey : Prentice Hall International.
- Boylestad, L.R, (1997 ), *Introductory Circuits Analysis*, Eight edition, New Jersey:  
Prentice Hall International.
- Rockis,G, (1995 ), *Solid State Fundamentals for Electricians*, Homewood Illionois :  
American Technical Publishers.
- Millman.J & Sutanto ( 1992 ) , *Mikroelektronika Sistem Digital dan Rangkaian  
Analog* , Jakarta : Erlangga.
- Wasito, S ( 1992 ) , *Data Sheet Book1 : Data IC Linier, TTL dan CMOS*, Jakarta :  
Elex Media Komputindo, Gramedia.
- Steeman, J.P.M, ( 1996 ) , *Data Sheet Book 2* , Jakarta : Elex Media Komputindo,  
Gramedia.
- Stanley, D.W, ( 1994 ) , *Operational Amplifier With Linear Integrated Circuits*,  
Mac Millan College Publishing Company.
- Giancoli, C.D & Imawan,C ( 1994 ) , *Fisika* , Jilid1, Edisi ke-4, Jakarta : Erlangga.
- Coughlin, F.R & Driscoll, F.F & Soemitro ( 1985 ) , *Penguat Operasional dan  
Rangkaian Terpadu Linier*, Jakarta : Erlangga.
- Kissel , E.T, ( 1997 ), *Industrial Electronics*, Singapura : Prentice Hall International.

- Simpson, D.C , ( 1996 ) , *Industrial Electronics* , New Jersey : Prentice Hall International.
- Floyd, T.L , ( 1997 ) , *Electronics Fundamental Circuits, Device, and Applications*, Second edition, Singapore : Macmillan.
- Cooper, D.W & PakPahan,S , ( 1985 ) , *Intrumentasi Elektronika dan Teknik Pengukuran*, Edisi ke-2, Jakarta Pusat : Erlangga.
- Sutopo, B. , *Diktat Kuliah Mikroprosesor Lanjut*
- Tulus Kid, *Ultrasonic Remote Control*
- Okyanto, Y , ( 1998 ) , *Pengatur Cahaya Berbasis Mikrokontroler M68HC11* , Yogyakarta.
- Skroder, J.C , ( 1997 ) , *Using the M68HC11 Microcontroller*, New Jersey: Prentice Hall International.
- Motorola , ( 1995 ) , *M68HC11 E Series Technical Data* .
- Motorola , ( 1991 ) , *M68HC11 Referensi Manual*.
- Motorola , ( 1992 ) , *M68HC11 EVBU, Universal Evaluation Board User's Manual*
- Texas Instrument, ( 1984 ) , *Designing with TTL Integrated Circuit*.

## LAMPIRAN A

### INSTRUKSI - INSTRUKSI M68HC11

**A.1 Tabel Instruksi Pengisian, Pemindahan dan Penyimpanan**

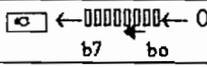
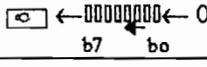
Mnemonic	Operasi	deskripsi	mode alamat	kode kondisi SXHINZVC
CLR (opr)	menghapus isi memori	$0 \Rightarrow M$	EXT IND	----0100
CLRA	menghapus akumulator A	$0 \Rightarrow A$	INH	----0100
CLRB	menghapus akumulator B	$0 \Rightarrow B$	INH	----0100
LDAA (opr)	mengisi akumulator A	$M \Rightarrow A$	IMM DIR EXT IND	...- $\uparrow\downarrow$ 0-
LDAB (opr)	mengisi akumulator B	$M \Rightarrow B$	IMM DIR EXT IND	...- $\uparrow\downarrow$ 0-
LDD (opr)	mengisi akumulator D	$M \Rightarrow A,$ $M+1 \Rightarrow B$	IMM DIR EXT IND	...- $\uparrow\downarrow$ 0-
PULA	menarik isi A dari stack	$SP = SP + 1$ $A \uparrow$ stk	INH	-----
PULB	menarik isi B dari stack	$SP = SP + 1$ $A \uparrow$ stk	INH	-----
PSHA	menyimpan isi A ke stack	$A \downarrow$ stk $SP = SP - 1$	INH	-----
PSHB	menyimpan isi B ke stack	$A \downarrow$ stk $SP = SP - 1$	INH	-----
STAA (opr)	menyimpan isi A ke memori	$A \Rightarrow M$	DIR EXT IND	...- $\uparrow\downarrow$ 0-
STAB (opr)	menyimpan isi B ke memori	$B \Rightarrow M$	DIR EXT IND	...- $\uparrow\downarrow$ 0-
STD (opr)	menyimpan isi D ke memori	$D \Rightarrow M$	DIR EXT IND	...- $\uparrow\downarrow$ 0-
TAB	memindahkan isi A ke B	$A \Rightarrow B$	INH	...- $\uparrow\downarrow$ 0-
TAP	memindahkan isi A ke CCR	$A \Rightarrow CCR$	INH	$\uparrow\downarrow\uparrow\downarrow\uparrow\downarrow$
TBA	memindahkan isi B ke A	$B \Rightarrow A$	INH	...- $\uparrow\downarrow$ 0-
TPA	memindahkan isi B ke CCR	$B \Rightarrow CCR$	INH	-----
XGDX	menukarkan isi D dengan IX	$D \leftrightarrow IX$	INH	-----
XGDY	menukarkan isi D dengan IY	$D \leftrightarrow IY$	INH	-----

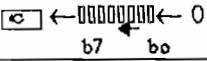
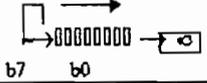
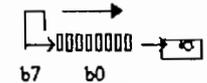
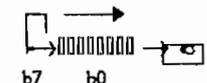
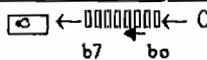
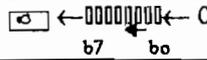
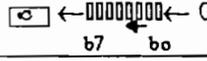
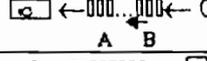
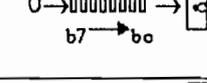
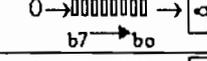
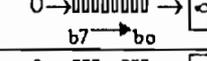
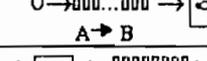
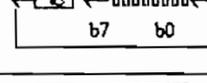
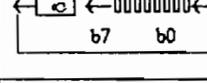
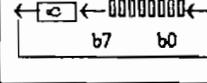
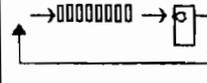
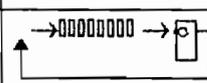
## A.2 Tabel Instruksi Aritmatika

Mnemonik	Operasi	deskripsi	mode alamat	kode kondisi SXHINZVC
ABA	menambahkan akumulator	$A + B \Rightarrow A$	INH	- - $\downarrow$ - $\downarrow\downarrow\downarrow\downarrow$
ABX	menambahkan B ke X	$IX + (00:B) \Rightarrow IX$	INH	- - - - -
ABY	menambahkan B ke Y	$IY + (00:B) \Rightarrow IY$	INH	- - - - -
ADCA	menambahkan isi memori dan carry ke A	$A + M + C \Rightarrow A$	IMM DIR EXT IND	- - $\downarrow$ - $\downarrow\downarrow\downarrow\downarrow$
ADCB	menambahkan isi memori dan carry ke B	$B + M + C \Rightarrow B$	IMM DIR EXT IND	- - $\downarrow$ - $\downarrow\downarrow\downarrow\downarrow$
ADDA	menambahkan isi memori ke A	$A + M \Rightarrow A$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
ADDB	menambahkan isi memori ke B	$B + M \Rightarrow B$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
ADDD	menambah isi ke memori C	$D + (M:M+1) \Rightarrow D$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
CBA	membandingkan A dengan B	$A - B$	INH	- - - $\downarrow\downarrow\downarrow\downarrow$
CMPA	membandingkan A dengan memori	$A - M$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
CMPB	membandingkan B dengan memori	$B - M$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
CPD	membandingkan D dengan memori	$D - M : M+1$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
DAA	penyesuaian desimal A	menyesuaikan hasil dengan format BCD	INH	- - - $\downarrow\downarrow\downarrow\downarrow$
DEC	menurunkan isi memori	$M - 1 \Rightarrow M$	EXT IND	- - - $\downarrow\downarrow\downarrow$ -
DECA	menurunkan A	$A - 1 \Rightarrow A$	INH	- - - $\downarrow\downarrow\downarrow$ -
DECB	menurunkan B	$B - 1 \Rightarrow B$	INH	- - - $\downarrow\downarrow\downarrow$ -
INC	menaikkan isi memori	$M + 1 \Rightarrow M$	EXT IND	- - - $\downarrow\downarrow\downarrow$ -
INCA	menaikkan A	$A + 1 \Rightarrow A$	INH	- - - $\downarrow\downarrow\downarrow$ -
INCB	menaikkan B	$B + 1 \Rightarrow B$	INH	- - - $\downarrow\downarrow\downarrow$ -
NEG	komplemen dua dari memori	$0 - M \Rightarrow M$	EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$
NEGA	komplemen dua A	$0 - A \Rightarrow A$	INH	- - - $\downarrow\downarrow\downarrow\downarrow$
NEGB	komplemen dua B	$0 - B \Rightarrow B$	INH	- - - $\downarrow\downarrow\downarrow\downarrow$
SBA	A dikurangi B	$A - B \Rightarrow A$	INH	- - - $\downarrow\downarrow\downarrow\downarrow$
SBCA	A dikurangi M dan carry	$A - M - C \Rightarrow A$	IMM DIR EXT IND	- - - $\downarrow\downarrow\downarrow\downarrow$

SBCB	B dikurangi M dan carry	$A \cdot M \cdot C \Rightarrow A$	IMM DIR EXT IND	- - - $\uparrow\uparrow\uparrow\uparrow$
SUBA	A dikurangi M	$A - M \Rightarrow A$	IMM DIR EXT IND	- - - $\uparrow\uparrow\uparrow\uparrow$
SUBB	B dikurangi M	$B - M \Rightarrow B$	IMM DIR EXT IND	- - - $\uparrow\uparrow\uparrow\uparrow$
SUBD	D dikurangi M	$D - M : M + 1 \Rightarrow A$	IMM DIR EXT IND	- - - $\uparrow\uparrow\uparrow\uparrow$
TST	Test isi M nol atau negatif	$M - 0$	EXT IND	- - - $\uparrow\uparrow 00$
TSTA	Test isi A nol atau negatif	$A - 0$	INH	- - - $\uparrow\uparrow 00$
TSTB	Test isi B nol atau negatif	$B - 0$	INH	- - - $\uparrow\uparrow 00$
MUL	Perkalian A dan B	$A \cdot B \Rightarrow D$	INH	- - - - - $\uparrow$
FDIV	Pembagian pecahan	$D / IX \Rightarrow IX$ $r \Rightarrow D$	INH	- - - - $\uparrow\uparrow\uparrow$
IDIV	Pembagian bilangan bulat	$D / IX \Rightarrow IX$ $r \Rightarrow D$	INH	- - - - $\uparrow 0 \uparrow$
COM	komplemen satu dari M	$\$FF - M \Rightarrow M$	EXT IND	- - - $\uparrow\uparrow 01$
COMA	komplemen satu dari A	$\$FF - A \Rightarrow A$	INH	- - - $\uparrow\uparrow 01$
COMB	komplemen satu dari B	$\$FF - B \Rightarrow B$	INH	- - - $\uparrow\uparrow 01$

### A.3 Instruksi Logika, Geser dan Rotasi

Mnemonic	Operasi	deskripsi	mode alamat	kode kondisi SXHINZVC
ANDA	AND A dengan memori	$A \cdot M \Rightarrow A$	IMM DIR EXT IND	- - - $\uparrow\uparrow 0 -$
ANDB	AND B dengan memori	$B \cdot M \Rightarrow B$	IMM DIR EXT IND	- - - $\uparrow\uparrow 0 -$
EORA	Exclusive OR A dengan memori	$A \oplus M \Rightarrow A$	IMM DIR EXT IND	- - - $\uparrow\uparrow 0 -$
EORB	Exclusive OR B dengan memori	$B \oplus M \Rightarrow B$	IMM DIR EXT IND	- - - $\uparrow\uparrow 0 -$
ORAA	OR A dengan A	$A + M \Rightarrow A$	IMM DIR EXT IND	- - - $\uparrow\uparrow 0 -$
ORAB	OR B dengan B	$B + M \Rightarrow B$	IMM DIR EXT IND	- - - $\uparrow\uparrow 0 -$
ASL	geser M ke kiri		EXT IND	- - - $\uparrow\uparrow\uparrow\uparrow$
ASLA	geser A ke kiri		INH	- - - $\uparrow\uparrow\uparrow\uparrow$

ASLB	geser B ke kiri		INH	- - -↑↑↑↑
ASLD	geser D ke kiri		INH	- - -↑↑↑↑
ASR	geser M ke kanan secara aritmatika		EXT IND	- - -↑↑↑↑
ASRA	geser A ke kanan secara aritmatika		INH	- - -↑↑↑↑
ASRB	geser B ke kanan secara aritmatika		INH	- - -↑↑↑↑
LSL	= ASL		EXT IND	- - -↑↑↑↑
LSLA	= ASLA		INH	- - -↑↑↑↑
LSLB	= ASLB		INH	- - -↑↑↑↑
LSLD	=ASLD		INH	- - -↑↑↑↑
LSR	geser M ke kanan secara logika		EXT IND	- - -0↑↑↑
LSRA	geser A ke kanan secara logika		INH	- - -0↑↑↑
LSRB	geser B ke kanan secara logika		INH	- - -0↑↑↑
LSRD	geser D ke kanan secara logika		INH	- - -0↑↑↑
ROL	rotasi M ke kiri		EXT IND	- - -↑↑↑↑
ROLA	rotasi A ke kiri		INH	- - -↑↑↑↑
ROLB	rotasi B ke kiri		INH	- - -↑↑↑↑
ROR	rotasi M ke kanan		EXT IND	- - -↑↑↑↑
RORA	rotasi A ke kanan		INH	- - -↑↑↑↑
RORB	rotasi B ke kanan		INH	- - -↑↑↑↑

## A.4 Instruksi Indeks dan Tumpukan

Mnemonik	Operasi	deskripsi	mode alamat	kode kondisi SXHINZVC
CPX	membandingkan X dengan memori	$IX - M : M+1$	IMM DIR EXT IND	- - - - $\updownarrow\updownarrow\updownarrow$
CPY	membandingkan Y dengan memori	$IY - M : M+1$	IMM DIR EXT IND	- - - - $\updownarrow\updownarrow\updownarrow$
DEX	menurunkan X	$IX - 1 \Rightarrow IX$	INH	- - - - $\downarrow$ - -
DEY	menurunkan Y	$IY - 1 \Rightarrow IY$	INH	- - - - $\downarrow$ - -
INX	menaikkan X	$IX + 1 \Rightarrow IX$	INH	- - - - $\uparrow$ - -
INY	menaikkan Y	$IY + 1 \Rightarrow IY$	INH	- - - - $\uparrow$ - -
LDX	mengisi X	$M : M+1 \Rightarrow IX$	IMM DIR EXT IND	- - - - $\updownarrow\updownarrow$ 0 -
LDY	mengisi Y	$M : M+1 \Rightarrow IY$	IMM DIR EXT IND	- - - - $\updownarrow\updownarrow$ 0 -
PSHX	push X	$IX \downarrow \text{Stk}, SP = SP - 2$	INH	- - - - - - - -
PSHY	push Y	$IY \downarrow \text{Stk}, SP = SP - 2$	INH	- - - - - - - -
PULX	pull X	$SP = SP + 2,$ $IX \uparrow \text{Stk}$	INH	- - - - - - - -
PULY	pull Y	$SP = SP + 2,$ $IY \uparrow \text{Stk}$	INH	- - - - - - - -
STX	menyimpan X	$IX \Rightarrow M : M+1$	DIR EXT IND	- - - - $\updownarrow\updownarrow$ 0 -
STY	menyimpan Y	$IY \Rightarrow M : M+1$	DIR EXT IND	- - - - $\updownarrow\updownarrow$ 0 -
TSX	memindahkan isi SP ke X	$SP + 1 \Rightarrow IX$	INH	- - - - - - - -
TSY	memindahkan isi SP ke Y	$SP + 1 \Rightarrow IY$	INH	- - - - - - - -
TXS	memindahkan isi X ke SP	$IX - 1 \Rightarrow SP$	INH	- - - - - - - -
TYS	memindahkan isi Y ke SP	$IY - 1 \Rightarrow SP$	INH	- - - - - - - -
DES	menurunkan SP	$SP - 1 \Rightarrow SP$	INH	- - - - - - - -
INS	menaikkan SP	$SP + 1 \Rightarrow SP$	INH	- - - - - - - -
LDS	mengisi SP	$M : M+1 \Rightarrow SP$	IMM DIR EXT IND	- - - - $\updownarrow\updownarrow$ 0 -
STS	menyimpan SP	$SP \Rightarrow M : M+1$	DIR EXT IND	- - - - $\updownarrow\updownarrow$ 0 -

### A.5. Instruksi Lompatan, percabangan, dan interupsi

Mnemonik	Operasi	Deskripsi	mode alamat	kode kondisi SXHINZVC
JMP	Jump	-	EXT IND	.....
JSR	Jump to Subroutine	-	DIR EXT IND	.....
RTI	Return from Interupt	-	INH	↑↓↑↓↑↓↑↓
RTS	Return from Subroutine	-	INH	.....
SWI	Software Interupt	-	INH	--- 1 ---
WAI	Wait for Interrupt	-	INH	.....
BCC	Mencabang jika carry nol	?C = 0	REL	.....
BCE	Mencabang jika carry satu	?C = 1	REL	.....
BEQ	Mencabang jika Zero = 1	?Z = 1	REL	.....
BGE	Mencabang jika $\geq 0$	?N $\oplus$ V = 0	REL	.....
BGT	Mencabang jika > Zero	?Z + (N $\oplus$ V) = 0	REL	.....
BHI	Mencabang jika >	?C + Z = 0	REL	.....
BHS	Mencabang jika $\geq$	?C = 0	REL	.....
BLE	Mencabang jika < Zero	?Z + (N $\oplus$ V) = 1	REL	.....
BLO	Mencabang jika <	?C = 1	REL	.....
BLS	Mencabang jika <	?C + Z = 1	REL	.....
BLT	Mencabang jika < Zero	?N $\oplus$ V = 1	REL	.....
BMI	Mencabang jika minus	?N = 1	REL	.....
BNE	Mencabang jika tidak = Zero	?Z = 1	REL	.....
BPL	Mencabang jika plus	?N = 0	REL	.....
BRA	Selalu mencabang	?I = 1	REL	.....
BRCLR (opr)	Mencabang jika bit (S) nol	?Mmm = 0	DIR IND	.....

### A.6 Instruksi Operasi bit dan register kondisi

Mnemonic	Operasi	deskripsi	mode alamat	kode kondisi SXHINZVC
BCLR M mm	menghapus bit	$\overline{M} \cdot (mm) \Rightarrow M$	DIR IND,X IND,Y	- - - - $\uparrow\downarrow$ 0 -
BITA M	test bit register A dengan isi memori	$A \cdot M$	IMM DIR EXT IND,X IND,Y	- - - - $\uparrow\downarrow$ 0 -
BITB M	test bit register B dengan isi memori	$B \cdot M$	IMM DIR EXT IND,X IND,Y	- - - - $\uparrow\downarrow$ 0 -
BSET M mm	mengeset bit	$M \cdot (mm) \Rightarrow M$	DIR IND,X IND,Y	- - - - $\uparrow\downarrow$ 0 -
CLC	menghapus bit carry	$0 \Rightarrow C$	INH	- - - - - - 0
CLI	menghapus interrupt mask	$0 \Rightarrow I$	INH	- - - 0 - - - -
CLV	menghapus overflow flag	$0 \Rightarrow V$	INH	- - - - - - 0 -
SEC	mengeset carry	$1 \Rightarrow C$	INH	- - - - - - 1
SEI	mengeset interrupt mask	$1 \Rightarrow I$	INH	- - - 1 - - - -
SEV	mengeset overflow flag	$1 \Rightarrow V$	INH	- - - - - - 1 -

### A.7 Instruksi - Instruksi Lain

Mnemonic	Operasi	deskripsi	mode alamat	kode kondisi SXHINZVC
NOP	tidak ada operasi	tidak ada operasi	INH	- - - - - -
STOP	menghentikan clock internal	-	INH	- - - - - -
TEST	test dalam mode test	bus alamat mencacah	INH	- - - - - -

**Keterangan :**

IMM : Immediate  
DIR : Direct  
EXT : Extentended  
IND : Indexed  
INH : Inherent

**Kode kondisi :**

- bit tidak berubah
- 0 bit selalu clear
- 1 bit selalu set
- ↕ bit clear atau set bergantung hasil operasi
- ↓ bit dapat clear tetapi tidak dapat set

## LAMPIRAN B

### PROGRAM TUGAS AKHIR DENGAN EKSTENSION LST

```

0001
*****
*
0002      *          PROGRAM UNTUK TUGAS AKHIR          *
0003      *      JUDUL : PENGUKURAN KETINGGLAN AIR DENGAN ULTRASONIK *
0004      *          BERBASIS MIKROKONTROLLER M68HC11      *
0005      *      OLEH : BERNADUS GUNAWAN SUDARSONO / no.Mhs : 945114008 *
0006
*****
*
0007
*****
*
0008      *      ALAMAT-ALAMAT PORT DAN REGISTER YANG DIGUNAKAN
*
0009
*****
*
0010 1000  PORTA      EQU $1000  ;Alamat PORTA
0011 100e  TCNT      EQU $100E  ;Alamat TCNT
0012 1010  TIC1      EQU $1010  ;Alamat TIC1
0013 0010  ISITCNT   EQU $0010  ;Penyimpan isi TCNT saat pulsa pinggiran naik
0014 0020  HAPEM10   EQU $0020  ;Penyimpan hasil pernbagian (TUNDA * 10)/10
0015 0024  TUNDA     EQU $0024  ;Penyimpan hasil penangkapan tunda waktu(10 pulsa)
0016 0022  HAPEM2    EQU $0022  ;Penyimpan hasil pembagian HAPEM10/2
0017 0012  RIN       EQU $0012  ;Penyimpan hasil pengurangan TIC1 - ISITCNT
0018 0022  MULTICNDMS EQU $0022  ;MS byte dari HAPEM2 yang akan dikalikan
0019 0023  MULTICNDLS EQU $0023  ;LS byte dari HAPEM2 yang akan dikalikan
0020 0040  MULTIPLRMS EQU $0040  ;MS byte sebagai pengali
0021 0041  MULTIPLRLS EQU $0041  ;LS byte sebagai pengali
0022 0080  P1MS      EQU $0080  ;Hasil MS byte perkalian MULTICNDLS*MULTIPLRLS
0023 0081  P1LS      EQU $0081  ;Hasil LS byte perkalian MULTICNDLS*MULTIPLRLS
0024 0082  P2MS      EQU $0082  ;Hasil MS byte perkalian MULTICNDMS*MULTIPLRLS
0025 0083  P2LS      EQU $0083  ;Hasil LS byte perkalian MULTICNDMS*MULTIPLRLS
0026 0084  P3MS      EQU $0084  ;Hasil MS byte perkalian MULTICNDLS*MULTIPLRMS
0027 0085  P3LS      EQU $0085  ;Hasil LS byte perkalian MULTICNDLS*MULTIPLRMS
0028 0086  P4MS      EQU $0086  ;Hasil MS byte perkalian MULTICNDMS*MULTIPLRMS
0029 0087  P4LS      EQU $0087  ;Hasil LS byte perkalian MULTICNDMS*MULTIPLRLS
0030 0090  PRODMS     EQU $0090  ;Hasil perkalian yang merupakan MS byte
0031 0091  PRODMIDH   EQU $0091  ;Hasil perkalian yang merupakan MID-H byte
0032 0092  PRODMIDL   EQU $0092  ;Hasil perkalian yang merupakan MID-L byte
0033 0093  PRODL     EQU $0093  ;Hasil perkalian yang merupakan LS byte
0034 0012  TBAKMS     EQU $0012  ;Tinggi bak air yaitu $00401300

```

```

0035 0013   TBAKMIDH   EQU $0013
0036 0014   TBAKMIDL   EQU $0014
0037 0015   TBAKLS     EQU $0015
0038 0042   HASMIN4    EQU $0042   ;Hasil pengurangan TBAK dengan PROD
0039 0043   HASMIN3    EQU $0043
0040 0044   HASMIN2    EQU $0044
0041 0045   HASMIN1    EQU $0045
0042 0030   BCD5       EQU $0030   ;Hasil konversi Biner (isi HASMIN) ke BCD
0043 0031   BCD4       EQU $0031
0044 0032   BCD3       EQU $0032
0045 0033   BCD2       EQU $0033
0046 0034   BCD1       EQU $0034
0047 0035   HASAKHIR   EQU $0035   ;Hasil akhir yang akan ditampilkan di SEGMENT
0048 6000   PPIA       EQU $6000   ;PPI PORTA 8255
0049 6001   PPIB       EQU $6001   ;PPI PORTB 8255
0050 6002   PPIC      EQU $6002   ;PPI PORTC 8255
0051 6003   PPIREG    EQU $6003   ;Kontrol Register PPI 8255
0052

```

```
*****
```

```

*
0053          *          PROGRAM MULAI DIKERJAKAN          *
0054          * 1. Inisialisasi Stack Pointer,Interrupt Vector Input Capture 1, TCTL2(Ti- *
0055          * mer Control 2, TMSK1 (Timer Interrupt Mask 1). *
0056          * 2. Menolkan bit IC1F (Input Capture 1 Flag ) pada register TFLG1. Hal *
0057          * ini dilakukan karena untuk menangkap input capture,bit IC1F harus nol. *
0058

```

```
*****
```

```

*
0059 2000          ORG $2000          ;Alamat AWAL PROGRAM
0060 2000 8e 01 ff   LDS #$1FF          ;Inisialisasi Stack Pointer
0061 2003 ce 10 00   LDX #$1000        ;Inisialisasi X kontrol
0062 2006 86 7e     LDAA #$7E          ;Isi $7E ke $00E8 arti:Melompat ke-
0063 2008 97 e8     STAA $00E8        ;Interrupt Vector Input Capture 1 (IC1)
0064 200a cc 20 f6   LDD #AYO          ;Alamat AYO dimasukkan ke Interrupt Vector IC1
0065 200d dd e9     STD $00E9
0066 200f 86 10     LDAA #$10          ;Inisialisasi TCTL2 agar menangkap IC pada -
0067 2011 a7 21     STAA $21,X        ;pinggiran naik
0068 2013 1c 22 04   BSET $22,X $04    ;Inisialisasi TMSK1 agar IC1 dapat diinterupsi
0069 2016 1d 23 fb   BCLR $23,X $FB    ;Menolkan bit IC1F pada register TFLG1
0070

```

```
*****
```

```

*
0071          * 1. Isi di alamat TUNDA harus dinolkan terlebih dahulu, agar isi yang lama *
0072          * tidak terbawa dalam perhitungan. *
0073          * 2. CLI (Clear I Mask) digunakan untuk menolkan isi I pada CCR(Condition *
0074          * Code Register).CLI ini digunakan agar Interupsi bisa dilaksanakan. *
0075          * 3. MEMBUAT PULSA : *
0076          * a.Inisialisasi banyaknya pulsa yang akan dipancarkan setiap kali pemanca- *
0077          * ran di transduser ultrasonik dan penampilan di SEVEN SEGMENT. *
0078          * b.Membuat lebar pulsa selebar DELAY1 dan spasi pulsa sepanjang DELAY2 *
0079          * di PA6 (PORTA pin 6). *
0080          * c.Membuat jarak antar pemancaran (setiap pemancaran 10 pulsa) sepanjang *

```

```

0081      * DELAY      *
0082      * 3. Membaca isi TCNT(Timer Counter) saat pulsa pinggiran naik dipancarkan *
0083      * dan disimpan di alamat ISITCNT.      *
0084

```

```
*****
```

```
*
```

```

0085 2019 0e      CLI      ;Clear I mask
0086 201a cc 00 00 LAGI    LDD #$00      ;Menolkan isi alamat TUNDA
0087 201d dd 24      STD TUNDA
0088 201f c6 0a      LDAB #10      ;Inisialisasi banyaknya pulsa
0089 2021 7f 00 42   CLR HASMIN4   ;Hapus isi HASMIN4
0090 2024 7f 00 43   CLR HASMIN3   ;Hapus isi HASMIN3
0091 2027 7f 00 44   CLR HASMIN2   ;Hapus isi HASMIN2
0092 202a 7f 00 45   CLR HASMIN1   ;Hapus isi HASMIN1
0093 202d 7f 00 90   CLR PRODMS    ;Hapus isi PRODMS
0094 2030 7f 00 91   CLR PRODMIDH  ;Hapus isi PRODMIDH
0095 2033 7f 00 92   CLR PRODMIDL  ;Hapus isi PRODMIDL
0096 2036 7f 00 93   CLR PRODL    ;Hapus isi PRODL
0097 2039 86 40      RU      LDAA #$40      ;Masukkan $40 di Akumulator A (ACCA)
0098 203b b7 10 00   STAA PORTA    ;Simpan isi ACCA di PORTA yaitu
                                           PA6='1'(HIGH)
.....
0099 203e 18 fe 10 0e LDY TCNT      ;Isi TCNT dimasukkan pada register IY
0100 2042 bd 21 66   JSR DELAY1    ;PA6 dalam keadaan HIGH selama DELAY1
0101 2045 18 df 10   STY ISITCNT   ;Isi TCNT yang ada di IY disimpan di ISITCNT
0102 2048 86 00      LDAA #$00     ;Memasukkan $00 di ACCA (Akumulator A)
0103 204a b7 10 00   STAA PORTA    ;Simpan isi ACCA di PORTA yaitu PA6='0'
                                           (LOW)
.....
0104 204d bd 21 6c   JSR DELAY2    ;PA6 dalam keadaan LOW selama DELAY2
0105 2050 5a        DECB          ;Kurangi 1 isi ACCB (Akumulator B)
0106 2051 26 e6     BNE RU        ;Jika isi ACCB belum nol, bercabang ke RU
0107

```

```
*****
```

```

0108      * PROSES 1 :      *
0109      * 1. Membaca isi alamat TUNDA setiap pemancaran (10 pulsa) dan dibagi dengan *
0110      * 10 untuk hasil rata-rata pengukuran. Hasil rata-rata ini disimpan di *
0111      * alamat HAPEM10.      *
0112      * 2. Membaca isi alamat HAPEM10 dan dibagi dengan 2. Hal ini dilakukan untuk *
0113      * mendapatkan waktu sesungguhnya yaitu:  $T_a + T_b$  *
0114      * Waktu(t) sesungguhnya = ----- *
0115      *  $2$  *
0116      *  $T_a + T_b$  yaitu waktu yang didapat setiap kali pulsa dipancarkan dan di- *
0117      * tangkap.  $T_a$  = waktu yang didapat sebelum dipantulkan air *
0118      *  $T_b$  = waktu yang didapat setelah dipantulkan air *
0119      * Hasil pembagian ini disimpan di alamat HAPEM2. *
0120

```

```
*****
```

```
*
```

```

0121 2053 dc 24      LDD TUNDA      ;Baca isi alamat TUNDA dan simpan di ACCD
0122 2055 ce 00 0a   LDX #$0A      ;Masukkan $0A(desimal:10) di register X (Pembagi)
0123 2058 02        IDIV          ;Isi ACCD dibagi isi Register X (TUNDA/10)
0124 2059 df 20     STX HAPEM10   ;Simpan hasil pembagian di alamat HAPEM10

```

```

0125 205b dc 20      LDD HAPEM10 ;Masukkan isi alamat HAPEM10 ke ACCD
                   ;(Akumulator D)
0126 205d ce 00 02  LDX #02      ;Masukkan 02 (Desimal:2) ke register X
0127 2060 02        IDIV          ;Isi ACCD dibagi isi Register X (HAPEM10/2)
0128 2061 df 22     STX HAPEM2    ;Simpan hasil pembagian (HAPEM10/2)di alamat
HAPEM2
0129
*****
0130      * PROSES 2 :
0131      * S = V * t di mana : S = Jarak antara transduser ultrasonik dengan air
0132      * V = Kecepatan suara ( V=331+(0.6 * X(suhu) )
0133      * Jika menggunakan 27 derajat celcius maka besar-
0134      * nya V = 347 (desimal) = 15B (Heksadesimal)
0135      * t = waktu sesungguhnya(disimpan di alamat HAPEM2)
0136      * Jika kita menggunakan perkalian dengan berbasis heksadesimal maka hasil-
0137      * akan heksadesimal juga. Untuk mengubahnya ke desimal perlu dikalikan de-
0138      * ngan 1 counter = 1 heksadesimal = 500 ns = 500 * 10(-9).
0139      * Perkalian ini sangat sulit dilakukan oleh sebab itu kita ambil angka 5
0140      * dari 500 ns ini agar perhitungannya bisa dilaksanakan.
0141      * Maka untuk Multiplier (MULTIPLR)/Pengali : V*5 = 15BH * 5H = 6C7H.
0142      * Maka S = 6C7H * t (waktu sesungguhnya yang disimpan di alamat HAPEM2)
0143      * Hasil perkalian disimpan di alamat PROD.
0144
*****
*
0145 2063 7f 60 02  CLR PPIC      ;Hapus isi PORTC PPI 8255
0146 2066 bd 21 5a  JSR DELAY     ; Panggil subrutin DELAY
0147 2069 7f 00 86  CLR P4MS      ;Hapus isi di alamat P4MS
0148 206c 86 06     LDAA #06      ;Masukkan MS byte pengali 06 di MULTIPLRMS
0149 206e 97 40     STAA MULTIPLRMS
0150 2070 86 c7     LDAA #07      ;Masukkan LS byte pengali 07 di MULTIPLRMS
0151 2072 97 41     STAA MULTIPLRMS
0152 2074 96 23     LDAA MULTICNDLS ;Masukkan isi alamat MULTICNDLS di ACCA
0153 2076 d6 41     LDAB MULTIPLRMS ;Masukkan isi alamat MULTIPLRMS di ACCB
0154 2078 3d        MUL          ;ACCA*ACCB ----> ACCD
0155 2079 dd 80     STD $0080     ;Simpan hasil MULTICNDLS*MULTIPLRMS di
                   ;$0080(P1)
0156 207b d7 93     STAB PRODL    ;Simpan isi ACCB di alamat PRODL
0157 207d 96 22     LDAA MULTICNDMS ;Masukkan isi alamat MULTICNDMS di ACCA
0158 207f d6 41     LDAB MULTIPLRMS ;Masukkan isi alamat MULTIPLRMS di ACCB
0159 2081 3d        MUL          ;ACCA*ACCB ----> ACCD
0160 2082 dd 82     STD $0082     ;Simpan hasil MULTICNDMS*MULTIPLRMS di
                   ;$0082(P2)
0161 2084 96 23     LDAA MULTICNDLS ;Masukkan isi alamat MULTICNDLS di ACCA
0162 2086 d6 40     LDAB MULTIPLRMS ;Masukkan isi alamat MULTIPLRMS di ACCB
0163 2088 3d        MUL          ;ACCA*ACCB ----> ACCD
0164 2089 dd 84     STD $0084     ;Simpan hasil MULTICNDLS*MULTIPLRMS di
                   ;$0084(P3)
0165 208b 96 22     LDAA MULTICNDMS ;Masukkan isi alamat MULTICNDMS di ACCA
0166 208d d6 40     LDAB MULTIPLRMS ;Masukkan isi alamat MULTIPLRMS di ACCB
0167 208f 3d        MUL          ;ACCA*ACCB ----> ACCD

```

```

0168 2090 dd 86      STD $0086      ;Simpan hasil MULTICNDMS*MULTIPLRMS di
.....$0086(P4)
0169 2092 96 80      LDAA P1MS      ;ACCA <---- P1MS
0170 2094 d6 83      LDAB P2LS      ;ACCB <----P2LS
0171 2096 1b         ABA            ;P1MS + P2LS -----> ACCA
0172 2097 99 85      ADCA P3LS      ;ACCA + Carry(C) + P3LS -----> ACCA
0173 2099 97 92      STAA PRODMIDL ;Simpan isi ACCA di alamat PRODMID-L
0174 209b 96 82      LDAA P2MS      ;ACCA <---- P2MS
0175 209d 99 84      ADCA P3MS      ;ACCA + Carry(C) + P3MS -----> ACCA
0176 209f 99 87      ADCA P4LS      ;ACCA + Carry(C) + P4LS -----> ACCA
0177 20a1 97 91      STAA PRODMIDH ;Simpan isi ACCA di alamat PRODMID-H
0178 20a3 96 86      LDAA P4MS      ;ACCA <---- P4MS
0179 20a5 89 00      ADCA # $00     ;ACCA + Carry(C) + # $00 -----> ACCA
0180 20a7 97 90      STAA PRODMS   ;Simpan isi ACCA di alamat PRODMS
0181
*****
0182      * PROSES 3 :      *
0183      * Agar sesuai dengan judul yaitu pengukuran ketinggian air dari suatu bak,      *
0184      * maka tinggi bak (L) harus dikurangi dengan S(Jarak tranduser ultrasonik      *
0185      * dengan air) yang hasilnya menandakan ketinggian air dari bak(T) tersebut.      *
0186      * Tinggi bak(L) yang digunakan sebenarnya adalah 38 cm, tetapi karena hasil-      *
0187      * pengukuran di sevent segmen mempunyai selisih sebesar 3,984 cm dengan tinggi      *
0188      * bak(L) yang sebenarnya, maka tinggi bak(L) dalam program ini menjadi 41,984 cm      *
0189      * yaitu merupakan pertambahan tinggi bak(L) yang sebenarnya (38 cm) dengan      *
0190      * selisih sebesar 3,984 cm. Maka untuk mendapatkan ketinggian air maka didapat-      *
0191      * kan suatu rumus yaitu T=L-S atau T(cm)=41,984(cm)-S(cm), dimana S disimpan di      *
0192      * alamat PROD. T(cm) disimpan di alamat HASMIN. 41,984 cm ini dalam program ini      *
0193      * adalah 4198400 (desimal) dan jika diubah dalam heksadesimal $00401000.      *
0194
*****
0195 20a9 86 00      LDAA # $00     ;Masukkan # $00 di alamat TBAKLS
0196 20ab 97 15      STAA TBAKLS
0197 20ad 86 10      LDAA # $10     ;Masukkan # $10 di alamat TBAKMID-L
0198 20af 97 14      STAA TBAKMIDL
0199 20b1 86 40      LDAA # $40     ;Masukkan # $40 di alamat TBAKMID-H
0200 20b3 97 13      STAA TBAKMIDH
0201 20b5 86 00      LDAA # $00     ;Masukkan # $00 di alamat TBAKMS
0202 20b7 97 12      STAA TBAKMS
0203 20b9 96 15      LDAA TBAKLS   ;ACCA <----- TBAKLS
0204 20bb d6 93      LDAB PRODL   ;ACCB <----- PRODL
0205 20bd 10         SBA           ;ACCA - ACCB -----> ACCA
0206 20be 97 45      STAA HASMIN1 ;Simpan hasil pengurangan (ACCA) di HASMIN1
0207 20c0 96 14      LDAA TBAKMIDL ;ACCA <---- TBAKMID-L
0208 20c2 92 92      SBCA PRODMIDL ;ACCA - PRODMID-L - CARRY(C) -----> ACCA
0209 20c4 97 44      STAA HASMIN2 ;Simpan hasil pengurangan (ACCA) di HASMIN2
0210 20c6 96 13      LDAA TBAKMIDH ;ACCA <---- TBAKMID-H
0211 20c8 92 91      SBCA PRODMIDH ;ACCA - PRODMID-H - CARRY(C) -----> ACCA
0212 20ca 97 43      STAA HASMIN3 ;Simpan hasil pengurangan (ACCA) di HASMIN3
0213 20cc 96 12      LDAA TBAKMS   ;ACCA <---- TBAKMS
0214 20ce 92 90      SBCA PRODMS  ;ACCA - PRODMS - CARRY(C) -----> ACCA
0215 20d0 97 42      STAA HASMIN4 ;Simpan hasil pengurangan (ACCA) di HASMIN4

```

0216

```

*****
0217      * Untuk mengeluarkan data di SEVEN SEGMENT maka perlu suatu perubahan dari *
0218      * heksadesimal menjadi BCD. Yang perlu diubah yaitu isi alamat HASMIN. Un- *
0219      * tuk keperluan perubahan heksadesimal menjadi BCD dapat dipanggil Subrou- *
0220      * tine BCDCONV. *
0221      * Karena MS Byte BCD4 berada pada posisi yang tidak diinginkan maka perlu *
0222      * suatu penggeseran sebanyak 4 kali dengan ASLD dan hasil disimpan di alamat *
0223      * HASAKHIR. Untuk lebih jelas tentang penggeseran ini dapat dibaca pada *
0224      * BAB IV. *
0225

```

\*

```

0226 20d2 bd 21 73      JSR BCDCONV      ;Panggil Subroutine BCDCONV
0227 20d5 86 80         LDAA #$80        ;ACCA <---- #$80
0228 20d7 b7 60 03     STAA PPIREG     ;Simpan ACCA di PPIREG
0229 20da dc 31         LDD BCD4        ;ACCD <---- BCD4
0230 20dc 05           ASLD            ;Geser ACCD ke kiri satu kali
0231 20dd 05           ASLD            ;Geser ACCD ke kiri satu kali
0232 20de 05           ASLD            ;Geser ACCD ke kiri satu kali
0233 20df 05           ASLD            ;Geser ACCD ke kiri satu kali
0234 20e0 dd 35       STD HASAKHIR    ;Simpan ACCD setelah digeser ke alamat HASAKHIR
0235

```

\*\*\*\*\*

```

0236      * Mengeluarkan hasil akhir di PORTA, PORTB dan PORTC di PPI 8255 *
0237      * 1. PORTA digunakan untuk mengeluarkan 2 digit sebelum koma. *
0238      * 2. PORTB digunakan untuk mengeluarkan 2 digit setelah koma. *
0239      * 3. PORTC digunakan untuk membuat pancaran LED (indikasi koma) *
0240      * PORTC yang digunakannya satu bit saja. *
0241      * Setelah 10 pulsa yang telah ditangkap diproses dan hasilnya keluar pada *
0242      * SEVENT SEGMENT, maka mikrokontroller siap untuk pemacaran 10 pulsa *
0243      * berikutnya dengan menampilkan kata pada monitor yaitu : *
0244      * PEMANCARAN 10 PULSA BERIKUTNYA...!!! *
0245      * Dengan adanya kata di atas maka mikrokontroller siap memancarkan 10 pulsa *
0246      * dan menangkapnya dengan Input Capture 1 dengan cara melompat ke alamat *
0247      * LAGI (Pemancaran ,penangkapan,dan penampilan selesai looping ke LAGI) *
0248

```

\*\*\*\*\*

\*

```

0249 20e2 b7 60 00     STAA PPIA      ;Simpan isi ACCA ke Port A PPI 8255
0250 20e5 f7 60 01     STAB PPIB      ;Simpan isi ACCB ke Port B PPI 8255
0251 20e8 86 01         LDAA #$01      ;Masukkan #$01 untuk bit PC0 =1 dari PORTC
0252 20ea b7 60 02     STAA PPIC
0253 20ed ce 21 33     LDX #YUK      ;Menampilkan kata'PEMANCARAN 10 PULSA
0254                  * BERIKUTNYA...!!!'setelah 10 pulsa *
0255 20f0 bd ff c7     JSR $FFC7     ;Memanggil subroutine OUTSTRG
0256 20f3 7e 20 1a     JMP LAGI      ;Melompat ke alamat LAGI untuk pemancaran lagi
0257

```

\*\*\*\*\*

```

0258      * SUBROUTINE AYO Digunakan untuk menangkap masukan pinggir an positif *
0259      * pulsa dan counter berhenti. Isi counter dibaca di *
0260      * register TIC1(Timer Input Capture 1 ).Isi TIC1 di- *

```

```

0261      *      kurangi dengan isi di alamat ISITCNT(pembacaan isi      *
0262      *      counter saat pulsa pinggiran naik dipancarkan di PA6).      *
0263      *      Hasil pengurangan ini ditambahkan dengan isi di ala-      *
0264      *      mat TUNDA sebelumnya. Hasil penambahan ini disimpan di      *
0265      *      alamat TUNDA. Saat penangkapan pinggiran naik pulsa pa-      *
0266      *      da Input Capture 1 ditandai dengan munculnya kata :      *
0267      *      IC TELAH MENANGKAP PULSA pada monitor dengan memasuk-      *
0268      *      kan subroutine LOOP dan OUTSTRG.      *
0269      *      Agar Input Capture bisa menangkap pinggiran naik pul-      *
0270      *      sa maka bit IC1F pada register TFLG1 dinolkan      *
0271
*****
0272 20f6 fc 10 10      AYO      LDD TIC1      ;Baca isi TIC1 dan simpan di ACCD
0273 20f9 93 10      SUBD ISITCNT      ;ACCD - ISITCNT -----> ACCD
0274 20fb 7f 00 12      CLR RIN      ;Hapus isi RIN
0275 20fe dd 12      STD RIN      ;Simpan hasil pengurangan di alamat RIN
0276 2100 96 13      LDAA $0013      ;Masukkan isi LS byte RIN ke ACCA
0277 2102 d6 25      LDAB $0025      ;Masukkan isi LS byte TUNDA ke ACCB
0278 2104 1b      ABA      ;ACCA + ACCB -----> ACCA
0279 2105 97 25      STAA $0025      ;ACCA ----> TUNDA ( LS Byte )
0280 2107 96 12      LDAA $0012      ;Masukkan isi MS byte RIN ke ACCA
0281 2109 99 24      ADCA $0024      ;ACCA+Carry(C)+MS byte TUNDA ---> ACCA
0282 210b 97 24      STAA $0024      ;ACCA -----> TUNDA ( MS Byte )
0283 210d ce 21 19      LDX #LOOP      ;Masukkan alamat LOOP ke Register X
0284 2110 bd ff c7      JSR $FFC7      ;Panggil Subroutine OUTSTRING
0285 2113 86 04      LDAA #$04      ;Menolkan isi bit IC1F di register TFLG1
0286 2115 b7 10 23      STAA $1023
0287 2118 3b      RTI      ;Return from Interrupt
0288
*****
0289 *      SUBROUTINE UNTUK MENGELUARKAN KATA : 'IC TELAH MENANGKAP PULSA' *
0290      *      DAN KATA : 'PEMANCARAN 10 PULSA BERIKUTNYA'      *
0291      *      PADA LAYAR MONITOR.      *
0292
*****
0293 2119 49 43 20 54 45 4c      LOOP      FCC 'IC TELAH MENANGKAP PULSA'
      41 48 20 4d 45 4e
      41 4e 47 4b 41 50
      20 50 55 4c 53 41
0294 2131 04      FCB $04
0295 2132 39      RTS
0296 2133 50 45 4d 41 4e 43      YUK      FCC 'PEMANCARAN 10 PULSA BERIKUTNYA...!!!'
      41 52 41 4e 20 31
      30 20 50 55 4c 53
      41 20 42 45 52 49
      4b 55 54 4e 59 41
      2e 2e 2e 2e 21 21
      21
0297 2158 04      FCB $04
0298 2159 39      RTS
0299

```

```

*****
0300          *                SUBROUTINE DELAY                *
0301
*****
*
0302 215a 86 02      DELAY  LDAA #2
0303 215c ce d6 00   NA     LDX #$D600
0304 215f 09        YA     DEX
0305 2160 26 fd     BNE YA
0306 2162 4a        DECA
0307 2163 26 f7     BNE NA
0308 2165 39        RTS
0309 2166 86 08     DELAY1 LDAA #$08
0310 2168 4a        YU     DECA
0311 2169 26 fd     BNE YU
0312 216b 39        RTS
0313 216c ce 30 00  DELAY2 LDX #$3000
0314 216f 09        YI     DEX
0315 2170 26 fd     BNE YI
0316 2172 39        RTS
0317
*****
0318          *                SUBROUTINE UNTUK MENGUBAH HEKSADESIMAL ATAU BINER                *
0319          *                MENJADI BCD                *
0320
*****
*
0321 2173 7f 00 30   BCD CONV CLR BCD5      ;Hapus isi alamat BCD5
0322 2176 7f 00 31   CLR BCD4      ;Hapus isi alamat BCD4
0323 2179 7f 00 32   CLR BCD3      ;Hapus isi alamat BCD3
0324 217c 7f 00 33   CLR BCD2      ;Hapus isi alamat BCD2
0325 217f 7f 00 34   CLR BCD1      ;Hapus isi alamat BCD1
0326 2182 c6 20      LDAB #32      ;Counter = 32 untuk penggeseran
.....sebanyak 32kali
0327 2184 0c        ULANG   CLC          ;Clear Carry
0328 2185 96 45     LDAA HASMIN1 ;ACCA <---- HASMIN1
0329 2187 49        ROLA         ;Putar isi ACCA ke kiri satu kali
0330 2188 97 45     STAA HASMIN1 ;Simpan isi ACCA setelah diputar ke
.....HASMIN1
0331 218a 96 44     LDAA HASMIN2 ;ACCA <---- HASMIN2
0332 218c 49        ROLA         ;Putar isi ACCA ke kiri satu kali
0333 218d 97 44     STAA HASMIN2 ;Simpan isi ACCA setelah diputar ke
.....HASMIN2
0334 218f 96 43     LDAA HASMIN3 ;ACCA <---- HASMIN3
0335 2191 49        ROLA         ;Putar isi ACCA ke kiri satu kali
0336 2192 97 43     STAA HASMIN3 ;Simpan isi ACCA setelah diputar ke
.....HASMIN3
0337 2194 96 42     LDAA HASMIN4 ;ACCA <---- HASMIN4
0338 2196 49        ROLA         ;Putar isi ACCA ke kiri satu kali
0339 2197 97 42     STAA HASMIN4 ;Simpan isi ACCA setelah diputar ke
.....HASMIN4
0340 2199 96 34     LDAA BCD1    ;ACCA <---- BCD1

```

```

0341 219b 99 34      ADCA BCD1 ;ACCA + BCD1 + Carry(C) -----> ACCA
0342 219d 19         DAA ;Desimal Adjust ACCA
0343 219e 97 34      STAA BCD1 ;Simpan isi ACCA setelah proses DAA ke BCD1
0344 21a0 96 33      LDA A BCD2 ;ACCA <---- BCD2
0345 21a2 99 33      ADCA BCD2 ;ACCA + BCD2 + Carry(C) -----> ACCA
0346 21a4 19         DAA ;Desimal Adjust ACCA
0347 21a5 97 33      STAA BCD2 ;Simpan isi ACCA setelah proses DAA ke BCD2
0348 21a7 96 32      LDA A BCD3 ;ACCA <---- BCD3
0349 21a9 99 32      ADCA BCD3 ;ACCA + BCD3 + Carry(C) -----> ACCA
0350 21ab 19         DAA ;Desimal Adjust ACCA
0351 21ac 97 32      STAA BCD3 ;Simpan isi ACCA setelah proses DAA ke BCD3
0352 21ae 96 31      LDA A BCD4 ;ACCA <---- BCD4
0353 21b0 99 31      ADCA BCD4 ;ACCA + BCD4 + Carry(C) -----> ACCA
0354 21b2 19         DAA ;Desimal Adjust ACCA
0355 21b3 97 31      STAA BCD4 ;Simpan isi ACCA setelah proses DAA ke BCD4
0356 21b5 96 30      LDA A BCD5 ;ACCA <---- BCD5
0357 21b7 99 30      ADCA BCD5 ;ACCA + BCD5 + Carry(C) -----> ACCA
0358 21b9 19         DAA ;Desimal Adjust ACCA
0359 21ba 97 30      STAA BCD5 ;Simpan isi ACCA setelah proses DAA ke BCD5
0360 21bc 5a         DECB ;Kurangi 1 isi ACCB(isi Counter)
0361 21bd 26 c5      BNE ULANG ;Jika isi Counter != 0, bercabang ke ULANG
0362 21bf 39         RTS ;Return from Subroutine
0363
0364

```

## LAMPIRAN C

## TAMPILAN FILE S- RECORDS ( S19 )

S12320008E01FFCE1000867E97E8CC20F6DDE98610A7211C22041D23FB0ECC0000DD249  
S12320200A7F00427F00437F00447F00457F00907F00917F00927F00938640B7100018FEA3  
S1232040100EBD216618DF108600B71000BD216C5A26E6DC24CE000A02DF20DC20CE00271  
S123206002DF227F6002BD215A7F00868606974086C797419623D6413DDD80D7939622D651  
S1232080413DDD829623D6403DDD849622D6403DDD869680D6831B99859792968299849E5  
S12320A0879791968689009790860097158610971486409713860097129615D6931097458F  
S12320C0961492929744961392919743961292909742BD21738680B76003DC310505050578  
S12320E0DD35B76000F760018601B76002CE2133BDFFC77E201AFC101093107F0012DD121F  
S12321009613D6251B9725961299249724CE2119BDFFC78604B710233B49432054454C414  
S123212048204D454E414E474B41502050554C5341043950454D414E434152414E20313068  
S12321402050554C534120424552494B55544E59412E2E2E2E21212104398602CED600092B  
S123216026FD4A26F73986084A26FD39CE30000926FD397F00307F00317F00327F00337FC0  
S12321800034C6200C964549974596444997449643499743964249974296349934199734A6  
S12321A0963399331997339632993219973296319931199731963099301997305A26C539CF  
S9030000FC

