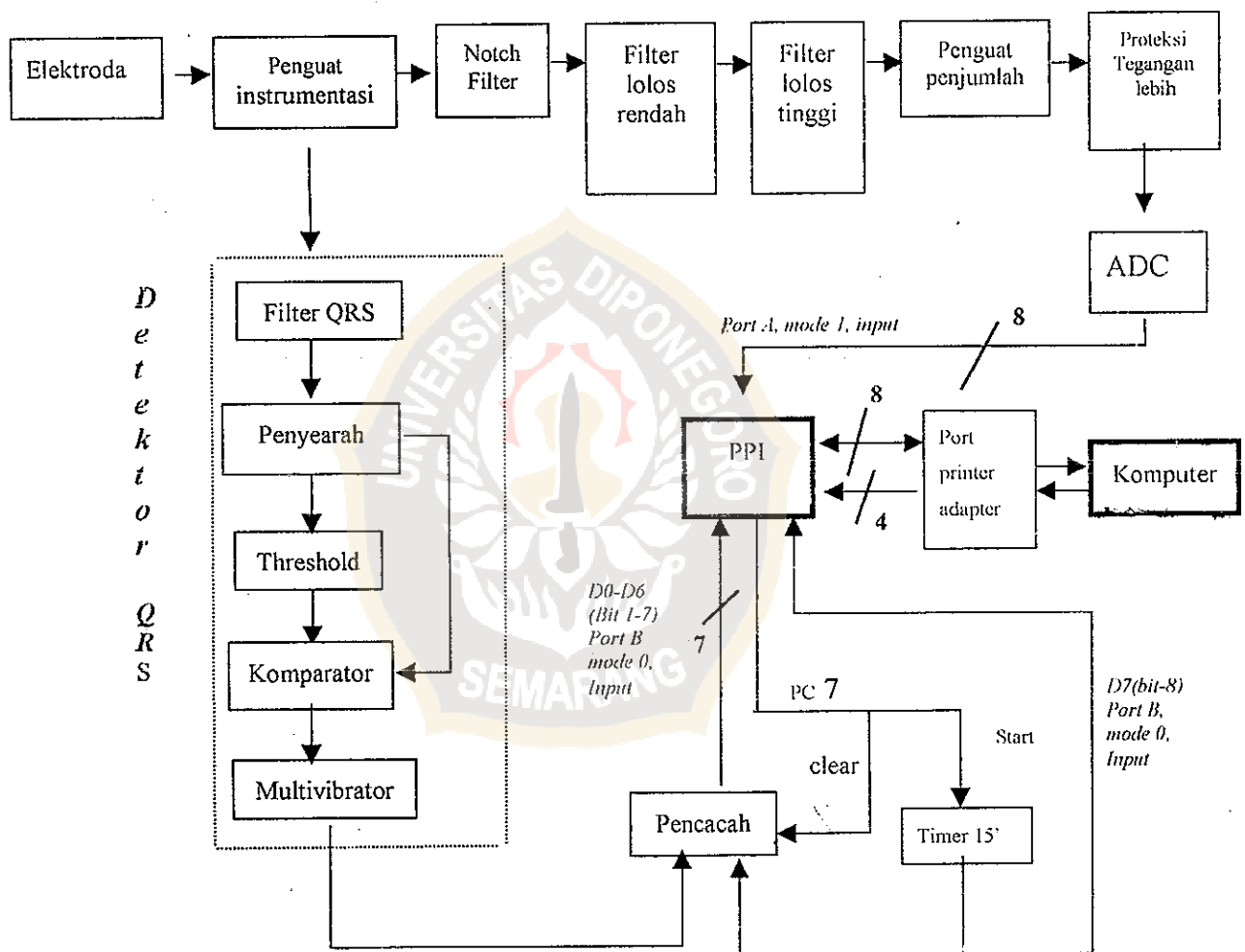


BAB III
PERANCANGAN DAN REALISASI PERANGKAT KERAS DAN
PERANGKAT LUNAK

3.1. Diagram Blok



Gambar 3.1 Blok Diagram Rangkaian Visualisasi dan Penghitung
 Detak Jantung (Tompkins, 1995)

Elektroda akan menerima sinyal dari jantung. Sinyal ini timbul karena adanya aktifitas listrik pada sel-sel jantung yang tersebar sampai kepermukaan

kulit tubuh. Sinyal ini masih lemah (dalam orde milivolt), sehingga perlu dikuatkan dengan penguat instrumentasi.

Sinyal keluaran dari penguat ini akan di pecah menjadi 2 buah sinyal, yaitu:

1. Sinyal pertama digunakan untuk menampilkan gambar sinyal elektrokardiogram pada layar monitor.
2. Sinyal kedua digunakan untuk menghitung cacah sinyal detak jantung per menit yang akan ditampilkan pada layar monitor.

Sinyal yang pertama akan difilter oleh filter *band-reject* untuk menghilangkan interferensi sinyal EKG karena tegangan jala-jala yang masuk lewat trafo dari *power supply*. Selanjutnya sinyal tersebut akan difilter oleh filter lolos rendah (frekuensi *cut off* 100 Hz) untuk menghilangkan interferensi frekuensi tinggi. Sinyal keluaran dari filter lolos rendah difilter oleh filter lolos tinggi (frekuensi *cut off* 0,5 Hz), untuk menghilangkan interferensi gerakan otot. Selanjutnya sinyal tersebut akan diproses oleh penguat penjumlah supaya diperoleh berupa sinyal jantung DC. Sebelum diberikan kepada ADC untuk dikonversikan, maka sinyal terlebih dahulu dilewatkan ke rangkaian proteksi tegangan dengan tujuan untuk menghindari kerusakan pada ADC karena pengaruh tegangan lebih (*over voltage*) pada masukan ADC. Pada ADC terjadi proses konversi sinyal analog menjadi sinyal digital. Data keluaran ADC diambil oleh komputer dari *port A PPI* melalui *Port Printer Adapter*, untuk ditampilkan pada layar monitor komputer.

Sedangkan sinyal yang kedua akan masuk ke rangkaian filter QRS untuk mengambil bagian sinyal jantung yang membawa informasi denyut jantung.

Sinyal informasi denyut jantung yang diperoleh akan mengalami proses penyearahan pada rangkaian penyearah. Rangkaian penyearah akan menghasilkan sinyal denyut jantung DC yang berupa sinyal dengan dua buah puncak, dengan puncak pertama memiliki amplitudo yang jauh lebih besar daripada puncak kedua. Nilai tegangan puncak tertinggi disimpan oleh rangkaian *threshold* dengan bantuan kapasitor penyimpan tegangan sebesar 1 μF . Keluaran rangkaian *threshold* berupa tegangan dengan amplitudo yang bergantung kepada puncak tegangan sinyal denyut jantung yang digunakan sebagai tegangan referensi komparator. Keluaran rangkaian komparator berupa logika rendah dan logika tinggi yang bergantian sesuai dengan irama denyut jantung. Keluaran komparator digunakan sebagai sinyal pemicu rangkaian *monostabil multivibrator*. Pulsa tunggal yang dihasilkan oleh *monostabil multivibrator* dioperasikan secara logika *AND* dengan keluaran rangkaian *timer* 15 detik, dan digunakan sebagai masukan rangkaian pencacah. Bilangan *biner* keluaran rangkaian pencacah ditampilkan pada layar komputer dengan melalui *Port B* PPI yang diprogram pada mode 0 masukan. Proses penghitungan jumlah denyut jantung dimulai dengan diberikannya sinyal pemicu pada rangkaian *timer* 15 detik yang dihasilkan oleh program didalam komputer dan dikeluarkan melalui *bit* ketujuh *port C* PPI.

3.2. Realisasi Perangkat Keras

Rangkaian-rangkaian yang diperlukan dalam pembuatan perangkat keras ini adalah sebagai berikut:

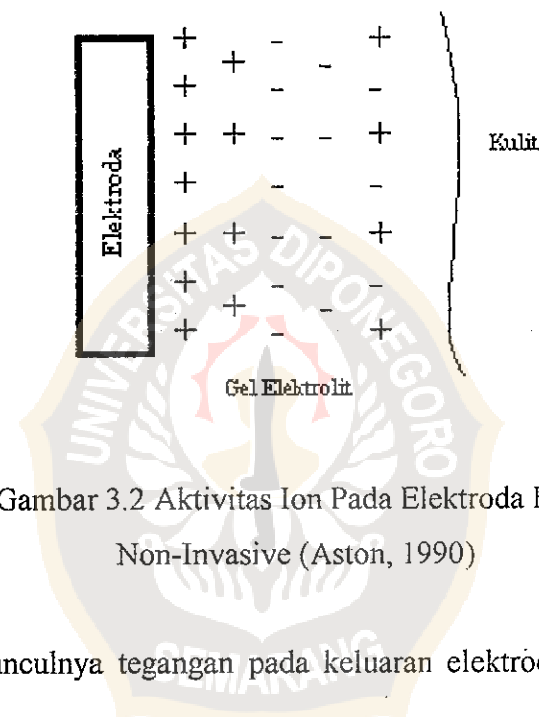
1. Rangkaian Elektroda EKG

2. Rangkaian Penguat Instrumentasi
3. Penampil Sinyal EKG
 - 3.1. Rangkaian Filter *Band-Reject*
 - 3.2. Rangkaian Filter Lolos Rendah
 - 3.3. Rangkaian Filter Lolos Tinggi
 - 3.4. Rangkaian Penguat penjumlah
 - 3.5. Rangkaian Proteksi Tegangan Lebih
 - 3.6. Rangkaian ADC 0804
4. Penghitung Denyut Nadi
 - 4.1. Rangkaian Filter QRS
 - 4.2. Rangkaian Penyearah
 - 4.3. Rangkaian *Threshold*
 - 4.4. Rangkaian Komparator
 - 4.5. Rangkaian *Monostabil multivibrator*
 - 4.6. Rangkaian Pencacah *7 Bit*
 - 4.7. Rangkaian *Timer* 15 detik
5. Rangkaian *Port* Printer Adapter
6. Rangkaian PPI8255



3.2.1. Elektroda EKG

Elektroda EKG yang digunakan dalam perancangan perangkat keras dipilih dari jenis *non-invasive*. Pemilihan ini didasarkan pada beberapa hal diantaranya kemudahan penempatan dan untuk meniadakan rasa nyeri yang akan timbul seperti jika digunakan jenis *invasive*.



Gambar 3.2 Aktivitas Ion Pada Elektroda EKG Non-Invasive (Aston, 1990)

Proses munculnya tegangan pada keluaran elektroda EKG *non-invasive* dapat dijelaskan sebagai berikut. Arus listrik yang muncul pada otot jantung akan tersebar sampai ke permukaan tubuh, dan besarnya bervariasi bergantung kepada jaraknya ke jantung. Arus listrik yang muncul pada permukaan tubuh akan mengakibatkan terjadinya beda potensial pada kedua sisi gel elektrolit elektroda EKG. Dalam hal ini gel elektrolit akan berfungsi sebagai kopling kapasitif antara permukaan tubuh dan elektroda EKG. Terjadinya perubahan muatan ion pada permukaan tubuh (karena aktifitas otot jantung), akan menyebabkan terjadinya perubahan muatan pada sisi gel elektrolit yang bersentuhan dengan elektroda.

Munculnya perubahan muatan ion pada masukan elektroda EKG akan direspon sebagai sinyal listrik pada keluaran elektroda EKG (Aston, 1990).

3.2.2. Penguat Instrumentasi

Sinyal yang dihasilkan oleh elektroda EKG masih sangat lemah sehingga harus dikuatkan sebelum diproses lebih lanjut. Karena sinyal EKG diukur dengan menggunakan teknik pengukuran differensial maka penguat yang digunakan adalah penguat differensial.

Penguat instrumentasi merupakan salah satu penerapan penguat operasional *double-ended-input single-ended-output*. Rangkaian ini menggunakan tiga buah penguat operasional yang dirangkai sebagai penguat differensial. Karakteristik penguat instrumentasi adalah memiliki *CMRR* yaitu kemampuan untuk meredam desah yang muncul pada terminal masukannya, memiliki penguatan yang besar, serta memiliki derajat kestabilan yang tinggi karena gain tidak tergantung kepada tahanan beban.

Pada prinsipnya penguat instrumentasi adalah sebuah differensial penguat dengan tambahan rangkaian *buffer* pada kedua terminal masukannya. Gambar 3.3 merupakan rangkaian penguat instrumentasi yang dirancang.

Rangkaian ini dibangun dengan menggunakan penguat operasional tipe LF353, yaitu jenis penguat operasional dengan masukan JFET (berimpedansi input sangat tinggi), memiliki *bandwidth* lebar dan berkecepatan tinggi. Rangkaian penguat yang mengikuti penguat instrumentasi adalah penguat *non-inverting* berfungsi untuk menjaga supaya penguat instrumentasi tetap stabil, karena pengaruh sinyal-sinyal desah (disebabkan oleh gerakan otot tubuh dan kedipan mata) dapat mengakibatkan penguat instrumentasi mengalami saturasi, disamping itu juga berfungsi untuk menguatkan sinyal lebih lanjut.

Berdasarkan persamaan pada Lampiran A.6 maka penguatan total rancangan pada gambar 3.3 adalah sebagai berikut (Aston, 1990):

$$\begin{aligned}
 A_{VTotal} &= A_{VBuffer} \cdot A_{VDifferensial} \cdot A_{VNonInverting} && \text{(Aston, 1990)} \\
 A_{VTotal} &= \left(\frac{22k + 22k + 10k + R_{Var10k}}{10k + R_{Var10k}} \right) \cdot \left(\frac{47k}{10k} \right) \cdot \left(\frac{150k}{4k7} + 1 \right) \\
 A_{VTotal} &= \left(\frac{54k + R_{Var10k}}{10k + R_{Var10k}} \right) \cdot (4,7) \cdot (32,9) \\
 A_{VTotal} &= (4,7) \cdot (32,9) \cdot \left(\frac{54k + R_{Var10k}}{10k + R_{Var10k}} \right)
 \end{aligned}$$

Kondisi tersebut diatas adalah syarat terjadi keseimbangan pada penguat differensial. Keseimbangan dapat diperoleh dengan cara mengatur harga tegangan pada terminal masukan positip penguat differensial sampai pada nilai tertentu.

3.2.3. Penampil Sinyal EKG

Sinyal EKG setelah dikuatkan oleh penguat instrumentasi akan memiliki amplitudo sebesar 1 volt. Dengan amplitudo sebesar 1 volt ini, maka sinyal cukup besar untuk diolah lebih lanjut. Sinyal keluaran penguat instrumentasi akan dibagi menjadi dua bagian, bagian pertama untuk menampilkan sinyal EKG dan bagian ke dua untuk menghitung denyut nadi.

3.2.3.1. Filter Band-Reject

Berfungsi untuk menghilangkan interferensi sinyal karena tegangan jala-jala listrik yang masuk kerangkaian melalui trafo pada supply tegangan (adaptor). Filter *band-reject* yang dirancang diharapkan memiliki kemampuan untuk menahan sinyal berfrekuensi 50 Hz.

Gambar 3.4 adalah rangkaian filter *band-reject* yang dirancang. Sesuai dengan prosedur perencanaan pada Lampiran F maka dapat dilakukan perhitungan sebagai berikut:

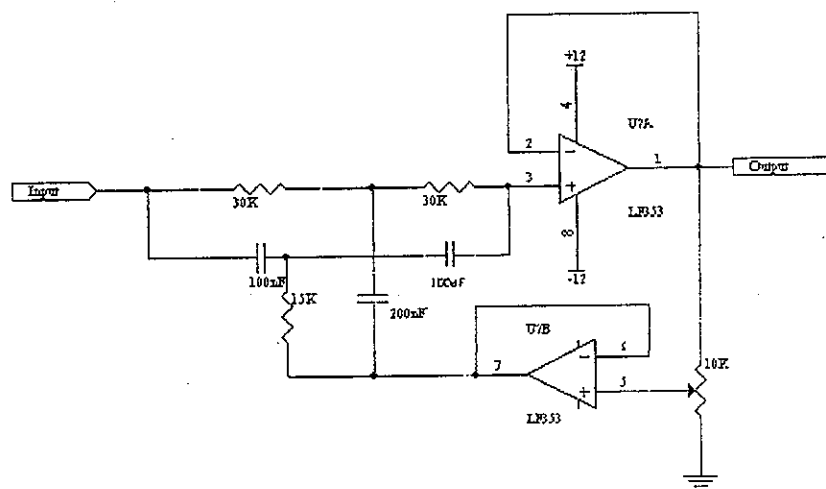
1. $R_{1,2}$ dipilih $30\text{ k}\Omega$, dan $R_3 = 15\text{ k}\Omega$
2. $C_{1,2}$ dihitung dari persamaan

$$C_{1,2} = \frac{1}{2\pi f_0 R} \quad (\text{Aston, 1990})$$

$$C_{1,2} = \frac{1}{2 \cdot 3,14 \cdot 50 \cdot 30\text{k}\Omega}$$

$$C_{1,2} \cong 100\text{nF}$$

Jadi $C_{1,2} = 100\text{nF}$ dan $C_3 = 200\text{nF}$ (merupakan kapasitor 100nF yang diparalel).

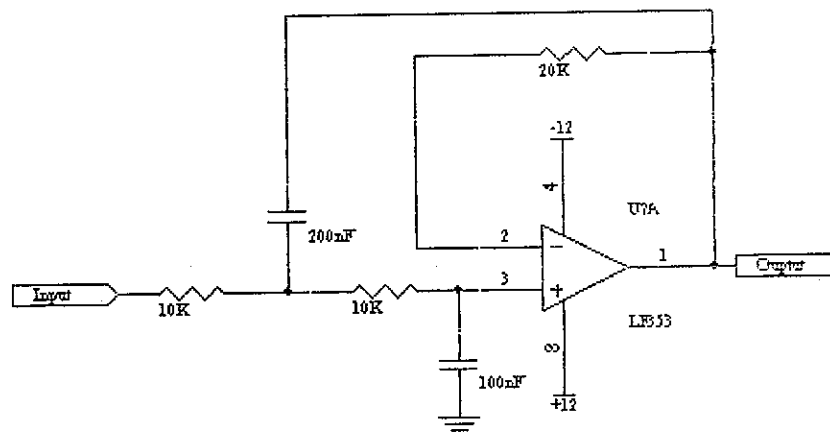


Gambar 3.4 Rangkaian Filter Band-Reject

3.2.3.2. Filter Lolos Rendah

Berfungsi untuk menghilangkan interferensi sinyal berfrekuensi tinggi yang merupakan desah bagi sinyal EKG. Sinyal berfrekuensi tinggi dapat muncul karena gelombang radio frekuensi tinggi.

Karena sinyal EKG memiliki batas frekuensi tertinggi 100 Hz, maka filter lolos rendah dirancang untuk memiliki frekuensi *cut-off* 100 Hz. Pada rancangan ini digunakan filter Butterworth lolos rendah dengan gain -40 dB, yaitu filter lolos rendah orde 2.



Gambar 3.5 Rangkaian Filter Butterworth
Lolos Rendah -40 dB

Gambar 3.5 adalah rangkaian filter lolos rendah yang dirancang. Sesuai dengan Lampiran B maka dapat dilakukan penghitungan sebagai berikut:

1. Ditentukan $f_c = 100$ hz
2. Dipilih $R_1 = R_2 = R = 10K\Omega$ dan $R_3 = 20K\Omega$
3. Dihitung harga C_1 dari persamaan:

$$C_1 = \frac{0,707}{2\pi f_c R} \quad (\text{Aston, 1990})$$

$$C_1 = \frac{0,707}{2 \cdot 3,14 \cdot 100 \cdot 10k\Omega}$$

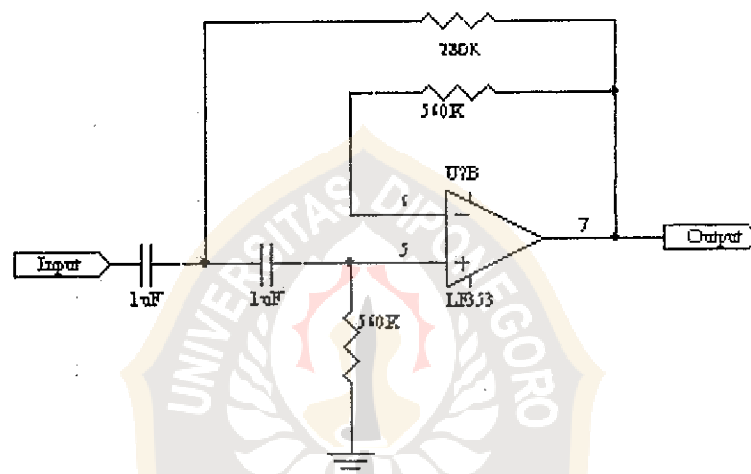
$$C_1 = 110nF$$

Nilai kapasitor yang paling mendekati yang ada di pasaran yaitu sebesar 100nF sehingga $C_{1,2} = 100nF$ dan $C_3 = 200nF$ (merupakan kapasitor 100nF yang diparalel).

3.2.3.3. Filter Lolos Tinggi

Berfungsi untuk menghilangkan interferensi sinyal berfrekuensi rendah. Sinyal berfrekuensi rendah muncul salah satunya akibat gerakan otot pada tubuh.

Sinyal EKG memiliki batas bawah 0,5 Hz, sehingga dirancang filter yang memiliki frekuensi *cut-off* 0,5 Hz. Rangkaian filter lolos tinggi yang dipilih merupakan jenis Butterworth dengan penguatan +40 dB, yaitu filter orde 2.



Gambar 3.6 Rangkaian Filter Butterworth
Lolos Tinggi +40 dB

Gambar 3.6 adalah rangkaian filter lolos tinggi yang dirancang pada rangkaian penampil sinyal EKG. Sesuai dengan Lampiran C maka dapat dilakukan penghitungan sebagai berikut:

1. Ditentukan $f_c = 0,5$ hz
2. Dipilih $C_1 = C_2 = C = 1\mu\text{F}$
3. Dihitung harga R_2 dari persamaan:

$$R_2 = \frac{0,707}{2\pi f_c C} \quad (\text{Aston, 1990})$$

$$R_2 = \frac{0,707}{2 \cdot 3,14 \cdot 0,5 \cdot 10^{-6}}$$

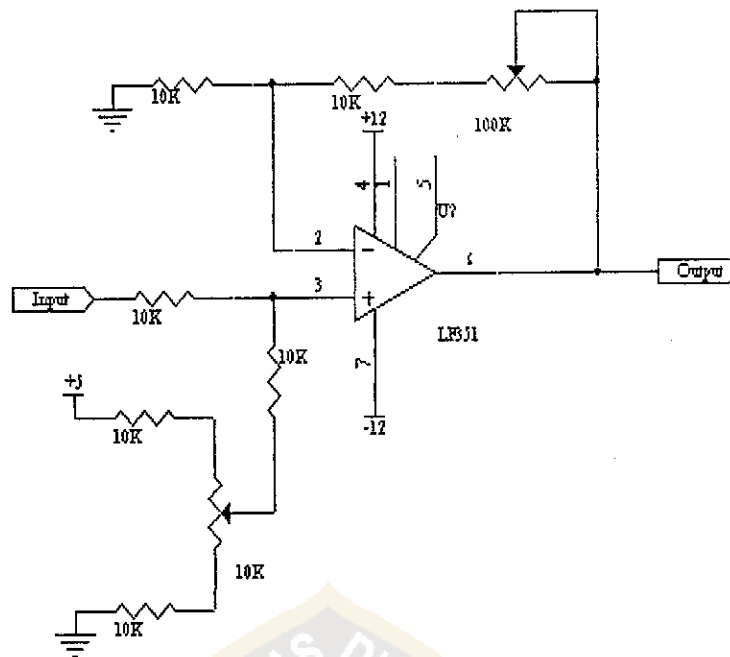
$$R_2 = 225k\Omega$$

Nilai resistor yang paling mendekati yang ada di pasaran yaitu sebesar $280k\Omega$ sehingga $R_{1,f} = 560k\Omega$ dan $R_2 = 280k\Omega$.

3.2.3.4. Penguat Penjumlah

Sinyal EKG merupakan sinyal *AC*, dengan amplitudo yang berubah-ubah disekitar nol dalam satu periode. Karena rangkaian akuisisi hanya dapat mengubah sinyal analog *DC* menjadi sinyal digital, maka sinyal EKG yang berupa sinyal analog *AC*, harus diolah untuk menjadi sinyal *DC*. Yaitu dengan cara menjumlahkan sinyal EKG dengan tegangan sebesar $+2,5 \text{ VDC}$ sehingga diperoleh sinyal EKG yang berubah-ubah disekitar $2,5 \text{ V}$. Hal ini dilakukan dengan menggunakan rangkaian penguat penjumlah.

Rangkaian penguat penjumlah pada prinsipnya akan menambahkan level *DC* ke sinyal EKG sehingga terjadi proses penggeseran amplitudo sinyal EKG kearah positif dan menguatkan sinyal yang telah digeser. Besarnya pergeseran bergantung kepada jumlah level tegangan *DC* yang dikenakan. Gambar 3.7 adalah rangkaian penguat penjumlah yang dirancang.



Gambar 3.7 Rangkaian Penguat Penjumlah

Berdasarkan persamaan Lampiran D.1 maka rangkaian dapat dianalisa sebagai berikut:

$$V_0 = \left(\frac{R_f}{R_1} + 1 \right) \left(\frac{R_b}{R_a + R_b} V_1 + \frac{R_a}{R_a + R_b} V_2 \right) \quad (\text{Boylestad, 1992})$$

$$V_0 = \left(\frac{10K + Var_{100K}}{10K} + 1 \right) \left(\frac{10K}{10K + 10K} V_{input} + \frac{10K}{10K + 10K} V_2 \right)$$

karena

$$V_2 = \frac{5}{10K + 10K + 10K} (10K + Var_{10K})$$

maka

$$V_0 = \left(\frac{10K + Var_{100K}}{10K} + 1 \right) \left(\frac{10K}{10K + 10K} V_{input} + \frac{10K}{10K + 10K} \frac{5}{10K + 10K + 10K} (10K + Var_{10K}) \right)$$

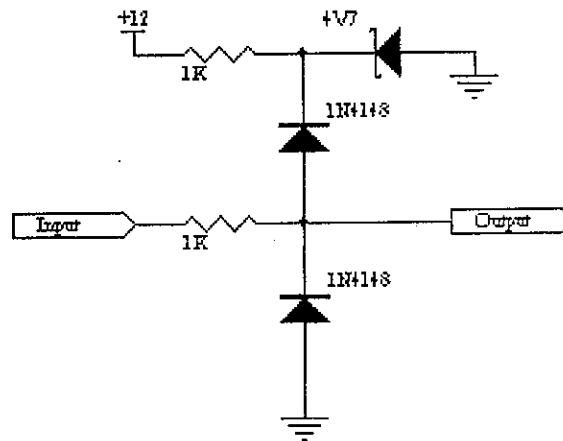
$$V_0 = \left(\frac{10K + Var_{100K}}{10K} + 1 \right) \left(\frac{1}{2} V_{input} + \frac{1}{12} (10K + Var_{10K}) \right)$$

Dengan demikian rangkaian yang dirancang memiliki penguatan yang dapat diatur dan penggeseran sinyal EKG yang dapat diatur juga. Pengaturan ini sangat berguna pada saat proses kalibrasi amplitudo sinyal EKG.

3.2.3.5. Proteksi Tegangan Lebih

Rangkaian proteksi tegangan lebih dirancang untuk menahan tegangan lebih, yang mungkin muncul pada keluaran penguat penjumlah. Tegangan lebih yang muncul dapat berupa tegangan positif maupun tegangan negatif.

Rangkaian ini dirancang dengan menggunakan sejumlah resistor, dioda dan dioda zener. Gambar 3.8 adalah rangkaian proteksi tegangan lebih yang dirancang.



Gambar 3.8 Rangkaian Proteksi Tegangan Lebih

Inti rangkaian ini terletak pada dioda zener 4V7. Sesuai dengan karakteristik dioda zener, maka rangkaian ini akan menahan tegangan yang mempunyai amplitudo dibawah 0 menjadi 0 dan diatas 5 volt menjadi 5 volt.

3.2.3.6. ADC 0804

Untuk rancang bangun alat ini digunakan ADC 0804, yang diproduksi oleh *National Semiconductor*, sebagai pengubah sinyal analog menjadi digital.

ADC 0804 mempunyai penyemat sebanyak 20 pin dan dikemas dalam bentuk DIP (*Dual In Line Package*).

Spesifikasi dari ADC 0804 (National Semiconductor, 1995) adalah :

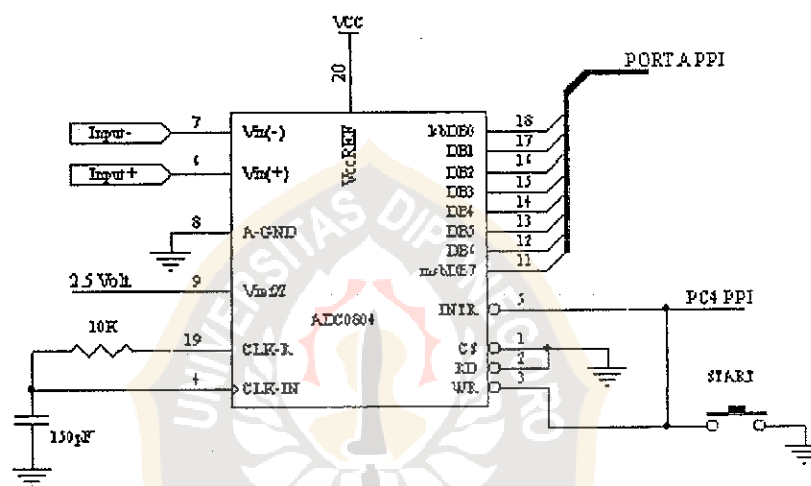
1. Kompatibel dengan turunan dari mikroprosesor 8080
2. Tidak diperlukan logik interface (antar muka)
3. Waktu akses sebesar 135 ns

4. Dapat dengan mudah diinterfacekan dengan semua mikroprosesor, atau dapat dioperasikan secara tersendiri.
5. Tegangan masukan analog diferensial
6. Level tegangan logik masukan dan keluaran kompatibel dengan tegangan MOS dan TTL
7. Tegangan referensi kerjanya sebesar 2,5 volt
8. Pembangkit *clock* detak seserpih (*On-Chip*)
9. Tegangan masukan analog antara 0V sampai dengan 5V dengan tegangan catu tunggal sebesar 5 volt DC
10. Tidak diperlukan pengaturan nol
11. Beroperasi secara ratiometrik (linier) atau $v_{\text{reff}} = 5V_{\text{DC}}$, atau $v_{\text{reff}}/2 = 2,5 V_{\text{DC}}$ atau dengan tegangan yang dapat diatur.
12. Resolusi 8 bit
13. Waktu konversi 100 μs
14. Kesalahan konversi total $\pm 1/4 \text{ LSB}$, $\pm 1/2 \text{ LSB}$, $\pm 1/8 \text{ LSB}$

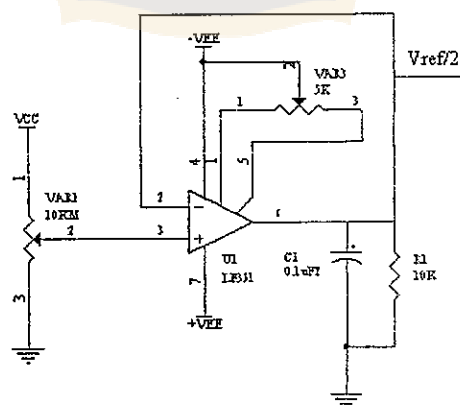
ADC0804 merupakan ADC berjenis *Successive Approximation* satu saluran masukan analog. Dalam rangkaian ini digunakan mode *free running*. Pada mode ini ADC0804 tidak memerlukan perintah SOC (*Start Of Conversion*) dari luar ADC0804, karena sinyal SOC didapatkan dari sinyal EOC (*End Of Conversion*) yang dihasilkan oleh ADC0804 dengan cara menghubungkan pin SOC dan EOC. Meskipun demikian pada mode *free running* diperlukan sinyal SOC *external* saat mulai digunakan, yaitu sinyal berlogika rendah.

Mode *free running* dipilih dengan pertimbangan bahwa yang diperlukan pada penelitian ini adalah kontinuitas data yang dapat dihasilkan oleh ADC0804 serta pertimbangan kesederhanaan sistem dan kesederhanaan proses kontrol.

Pin-pin keluaran ADC0804 dihubungkan dengan *port* A PPI yang diprogram pada mode 1 masukan. Sinyal SOC dihubungkan dengan Port C₄ PPI untuk jalur sinyal interupsi (*INTR*).



Gambar 3.9 Rangkaian ADC0804



Gambar 3.10 Rangkaian Tegangan Referensi ADC0804

Gambar 3.9 adalah rangkaian *free running* ADC0804 yang dirancang. Tegangan analog yang akan dikonversi oleh ADC0804 dapat dipilih pada range 0-5 volt, sehingga dilakukan pengaturan nilai tegangan yang masuk ke pin $V_{ref/2}$ ADC0804 pada nilai 2,5 V. Nilai tegangan pada $v_{ref/2}$ harus berupa tegangan yang stabil. Gambar 3.10 adalah rangkaian tegangan referensi yang dirancang.

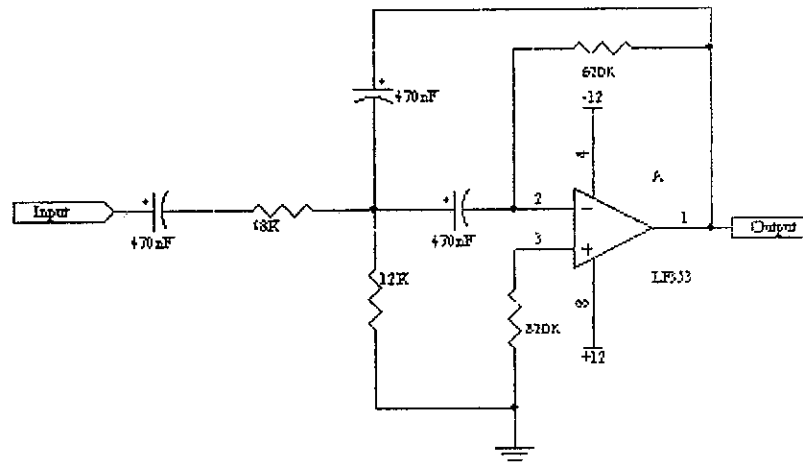
3.2.4. Penghitung Denyut Nadi

Sinyal EKG memiliki bermacam bagian (*segment*), segmen QRS adalah segmen yang membawa informasi denyut nadi. Pada manusia denyut ini berada pada frekuensi 17 Hz. Sehingga apabila segmen ini dapat di proses secara terpisah maka akan didapatkan denyut nadi.

Berikut ini rangkaian yang digunakan untuk pendeteksi denyut nadi.

3.2.4.1. Filter QRS

Sinyal QRS (denyut nadi) adalah segmen sinyal jantung dengan frekuensi 17 Hz (Tompkins,1995). Untuk mendeteksi segmen ini digunakan sebuah rangkaian filter *band-pass* dengan frekuensi center 17 Hz dan *bandwidth* 6 Hz.



Gambar 3.11 Rangkaian Filter QRS (Tompkins, 1995)

Sesuai dengan prosedur perencanaan pada Lampiran E maka dapat dilakukan penghitungan sebagai berikut:

1. Dipilih frekuensi tengah $f_r = 17 \text{ Hz}$, $|A_r| = 6$ dan Bandwidth 6 Hz
2. Dipilih $C_1 = C_3 = 100 \text{ nF}$
3. Dihitung R_2 dari persamaan:

$$R_2 = \frac{2}{B_w C} \quad (\text{Aston, 1990})$$

$$R_2 = \frac{2}{6 \cdot 100 \cdot 10^{-9}}$$

$$R_2 = 3M3$$

4. Dihitung R_1 dari persamaan:

$$R_1 = \frac{R_2}{2|A_r|} \quad (\text{Aston, 1990})$$

$$R_1 = \frac{3M3}{2 \cdot 6}$$

$$R_1 = 277k\Omega$$

Nilai resistor yang paling mendekati yang ada di pasaran adalah sebesar $280\text{k}\Omega$, sehingga $R_1 = 280\text{k}\Omega$.

5. Dihitung R_3 dari persamaan

$$Q = \frac{w_r}{B_w} \quad (\text{Aston, 1990})$$

$$Q = \frac{2 \cdot \pi \cdot 17}{6}$$

$$Q = 17,8$$

$$R_3 = \frac{R_2}{4Q^2 - 2|A_r|} \quad (\text{Aston, 1990})$$

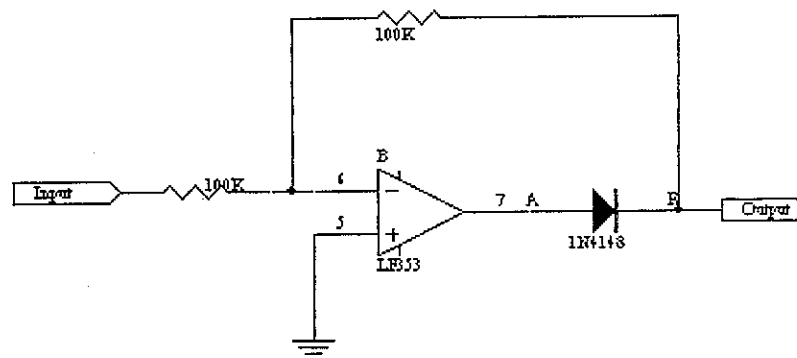
$$R_3 = \frac{3M3}{4(17,8)^2 - 2 \cdot 6}$$

$$R_3 = 2K6$$

Nilai resistor yang paling mendekati yang ada di pasaran yaitu sebesar 2670Ω , sehingga $R_3 = 2670\Omega$.

3.2.4.2. Penyearah Gelombang

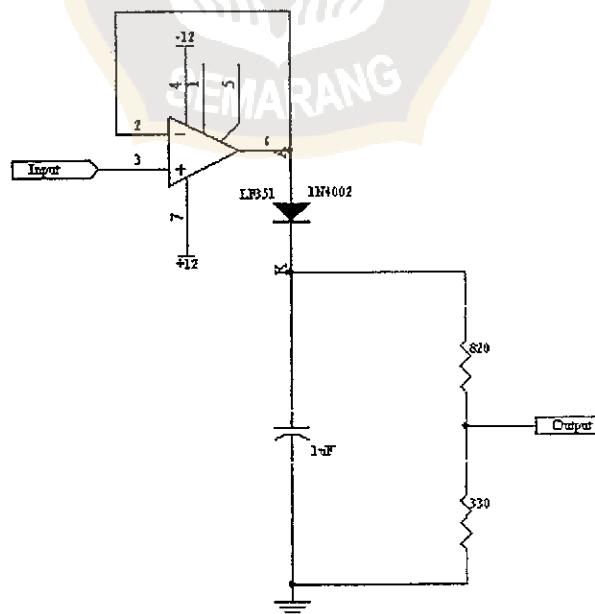
Sinyal QRS adalah sinyal AC dengan dua buah amplitudo yang berlainan pada sisi positif dan sisi negatifnya sehingga diperlukan rangkaian penyearah untuk mendapatkan puncak gelombang. Karena sinyal berupa sinyal kecil maka digunakan rangkaian penyearah berupa penguat operasional dan sebuah dioda penyearah. Gambar 3.12 adalah rangkaian penyearah yang dirancang.



Gambar 3.12 Rangkaian Penyearah Gelombang

3.2.4.3. Treshold

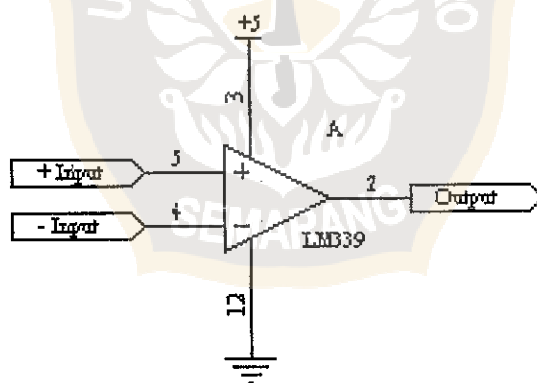
Rangkaian treshold digunakan untuk menangkap puncak bagian tertinggi sinyal QRS dan menyimpannya. Proses penyimpanan nilai puncak ditangani oleh kapasitor penyimpan $1\mu\text{F}$, dan waktu penyimpanannya didesain lebih kecil daripada periode sinyal QRS. Nilai puncak yang diperoleh akan dibuang melalui resistor yang dipasang paralel dengan kapasitor penyimpan.



Gambar 3.13 Rangkaian Treshold

3.2.4.4. Komparator

Rangkaian komparator berfungsi untuk menghasilkan *trigger* bagi *monostable multivibrator*. Sinyal *trigger* yang dihasilkan merupakan hasil perbandingan masukan yang keluar dari rangkaian *threshol*d dan dari rangkaian penyearah gelombang. Sinyal QRS yang berasal dari penyearah gelombang adalah sinyal QRS terkini sedangkan yang berasal dari *threshol*d adalah fraksi puncak sinyal QRS periode sebelumnya. Jadi proses perbandingan dilakukan antara sinyal QRS terkini dan fraksi puncak sinyal QRS periode sebelumnya. Fraksi puncak sinyal QRS periode sebelumnya dibentuk dari nilai tegangan yang muncul pada resistor $330k\Omega$ pada rangkaian *threshol*d. Gambar 3.14 adalah rangkaian komparator yang dirancang.

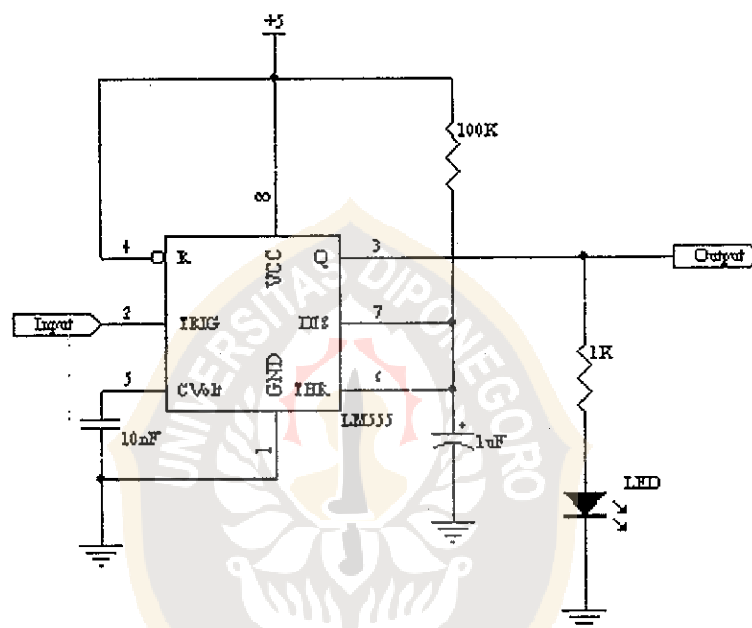


Gambar 3.14 Rangkaian Komparator

3.2.4.5. Monostable Multivibrator

Rangkaian ini akan menghasilkan sinyal keluaran setiap ada sinyal *trigger* yang masuk. Karena komparator memberikan sinyal keluaran berupa *trigger*

setiap ada denyut nadi, maka *monostable multivibrator* akan aktif setiap ada denyut nadi. *Monostable multivibrator* dirangkai agar dapat memberikan sinyal keluaran berupa sinyal satu tembakan(*one-shot*), yaitu sinyal *low-high-low*, dengan waktu *high* 110 ms. Gambar 3.15 adalah rangkaian *monostable* yang dirancang.



Gambar 3.15 Rangkaian Monostable Multivibrator

Sesuai dengan persamaan pada Lampiran G.1 , maka waktu tunda yang didapatkan adalah sesuai dengan persamaan berikut ini :

$$T_{\text{high}} = 1,1 \cdot R_A \cdot C \quad (\text{Boylestad, 1992})$$

$$T_{\text{high}} = 1,1 \cdot 100\text{k}\Omega \cdot 1\mu\text{F}$$

$$T_{\text{high}} = 0,11 \text{ detik}$$

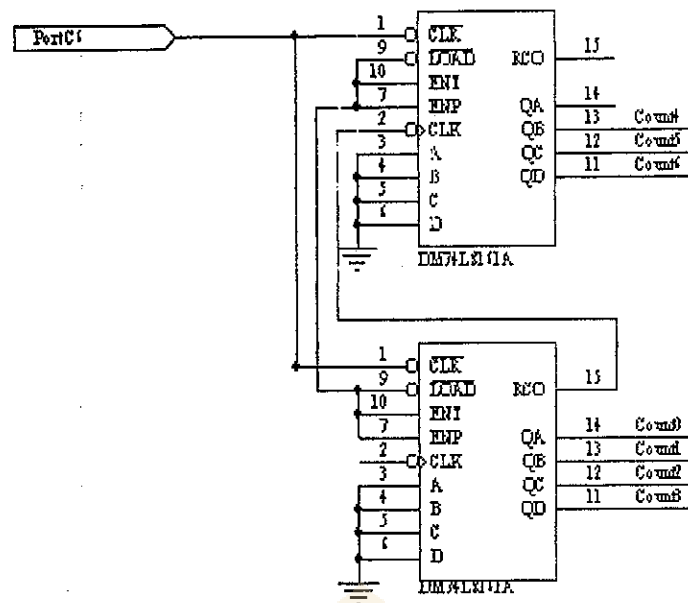
Jadi waktu tunda yang didapatkan adalah sebesar 110 millidetik.

3.2.4.6. Pencacah 7 Bit

Pencacah yang digunakan merupakan pencacah *biner (binary) 7 bit*. Rangkaian ini akan melakukan proses perhitungan dari nilai minimum 0 sampai maksimum nilai $2^7 - 1$ atau sebesar 127. Dalam hal ini ini digunakan IC pencacah tipe 74LS161 yang merupakan pencacah biner 4 *bit*.

Pada prinsipnya IC 74LS161 merupakan pencacah 4 *bit* yang dilengkapi dengan beragam fungsi tambahan yaitu *clear*, *load* dan *carry out*. *Clear* digunakan untuk mengenolkan keluaran (posisi nol), *load* digunakan untuk memasukkan nilai awal penghitungan yang dikehendaki, dan *carry out* digunakan untuk keperluan penyusunan secara seri. Pada susunan seri pencacah tipe ini dapat melakukan proses pencacahan sampai beberapa *bit* yang dikehendaki.

Masukan *clock* pada rangkaian pencacah didapatkan dari sinyal keluaran rangkaian deteksi QRS yang dioperasikan secara logika *AND* dengan keluaran rangkaian *timer* 15 detik. Pada rancangan masukan *reset* pada pencacah dihubungkan dengan *Port C₇* pada PPI 8255. Gambar 3.16 adalah rangkaian pencacah 7 bit yang dirancang.

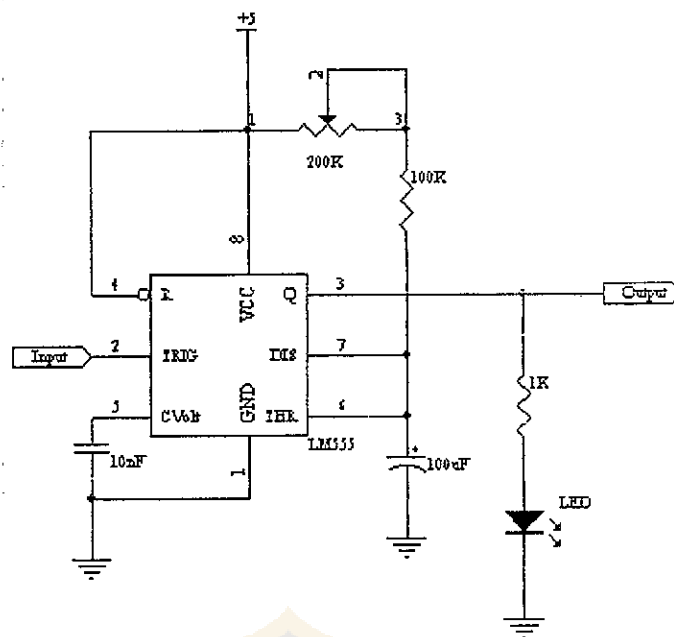


Gambar 3.16 Rangkaian Pencacah 7 Bit

3.2.4.7 Timer 15 Detik

Dalam dunia medis proses penghitungan detak nadi menggunakan satuan denyut per menit. Hal ini direalisasikan dalam perangkat keras dengan melakukan penghitungan denyut nadi tiap 15 detik dan kemudian hasil denyut yang diperoleh dikalikan dengan 4 sehingga didapatkan hasil denyut nadi per menit.

Untuk keperluan penghitungan denyut nadi, maka dirangkai *timer* yang dapat memberikan tundaan waktu sebesar 15 detik. Rangkaian ini disusun dengan basis IC tipe 555 yang dirangkai menurut rangkaian *monostable multivibrator*. Gambar 3.17 adalah rangkaian tundaan waktu 15 detik yang dirancang.



Gambar 3.17 Rangkaian Timer 15 Detik

Berdasarkan persamaan pada Lampiran G.1 maka dapat dilakukan penghitungan sebagai berikut :

$$T_{\text{high}} = 1,1.R_A.C$$

(Boylestad, 1990)

$$T_{\text{high}} = 1,1.(100\text{k}\Omega + VR).100\mu\text{F}$$

$$T_{\text{high}} = 110.(100\text{k}\Omega + VR)\mu\text{s}$$

Jika $T_{\text{high}} = 15$ detik maka harga VR dapat dicari sebagai berikut

$$15 = 110.(100\text{k}\Omega + VR).10^{-6}$$

$$VR = (15.10^6)/110 - 100\text{k}\Omega$$

$$VR = 36,36 \text{ k}\Omega$$

Jadi besarnya hambatan VR yang diperlukan adalah sebesar 36,36 k Ω

Sinyal *trigger* yang diperlukan untuk memulai tundaan waktu adalah berupa pulsa berlogika rendah. Dalam hal ini ini sinyal *trigger* diperoleh dari *Port C PPI 8255 bit ke 7*.

3.2.5. Port Printer Adapter

Dalam hal ini yang digunakan sebagai jalur I/O adalah *port* printer adapter, dengan demikian beragam jenis komputer dapat digunakan untuk keperluan pengontrolan modul.

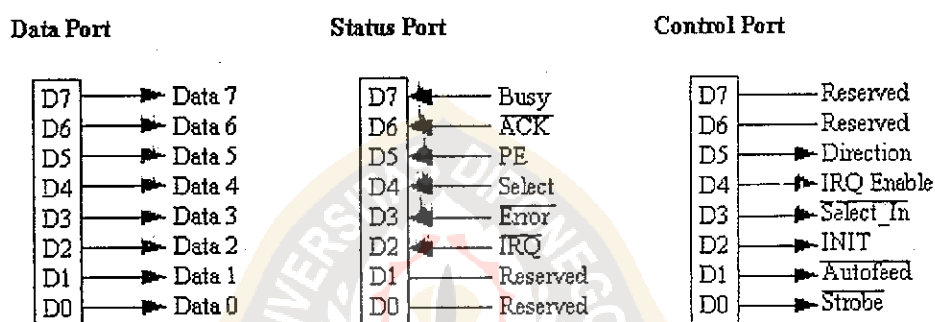
Port kontrol digunakan untuk mengirimkan sinyal kontrol ke PPI (D3 – D0). Untuk memudahkan pemrograman, maka D3 – D0 pada *port* kontrol didesain untuk memiliki sifat *inverting* semuanya. Untuk menjamin bahwa sinyal yang dikirimkan oleh komputer melalui *port* ini cukup kuat untuk menggerakkan PPI maka pada setiap keluaran *port* ini dihubungkan resistor 10 k Ω sebagai resistor *pull-up*.

Port Status yang digunakan adalah pada D6. *Bit* ini digunakan untuk mengirimkan sinyal *interrupt* ke komputer. *Bit* ini dihubungkan ke PC3 pada PPI yang merupakan jalur sinyal EOC yang dihasilkan oleh ADC0804.

Port Data dihubungkan dengan D0 – D7 pada PPI. *Port* ini digunakan untuk memindahkan informasi antara PPI dan komputer. Untuk mengaktifkan transfer data dua arah pada *port* printer adapter dilakukan dengan mengatur D4 pada *port* kontrol printer adapter. Jika D4 *port* kontrol printer adapter bernilai 1 maka *port* data printer adapter merupakan *port* masukan, bila bernilai 0 maka *port* data printer adapter merupakan *port* keluaran.

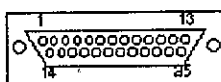
Gambar 3.18 adalah diskripsi *port* kontrol, *port* status dan *port* data pada printer adapter. Tidak semua bit pada ketiga *port* tersebut dapat digunakan dalam perancangan perangkat keras, hal ini karena *bit-bit* tersebut digunakan secara internal oleh komputer. *Bit-bit* yang dapat digunakan untuk kepentingan perancangan perangkat keras adalah seperti yang terdapat pada gambar 3.19

Gambar 3.19 adalah deskripsi pin-pin pada *port* printer adapter, yang dikemas dalam konektor/terminal yang disebut dengan DB25.



Gambar 3.18 Deskripsi *Port-port* pada Printer Adapter

(Anderson, ---)



View is looking at
Connector side of
DB-25 Male Connector.

Pin Description

Pin	Description	
1	$\overline{\text{Strobe}}$	PC Output
2	Data 0	PC Output
3	Data 1	PC Output
4	Data 2	PC Output
5	Data 3	PC Output
6	Data 4	PC Output
7	Data 5	PC Output
8	Data 6	PC Output
9	Data 7	PC Output
10	$\overline{\text{ACK}}$	PC Input
11	Busy	PC Input
12	Paper Empty	PC Input
13	Select	PC Input
14	$\overline{\text{Auto Feed}}$	PC Output
15	$\overline{\text{Error}}$	PC Input
16	Initialize Printer	PC Output
17	Select Input	PC Output

Pin Assignments

Note: 8 Data Outputs
4 Misc Other Outputs

5 Data Inputs

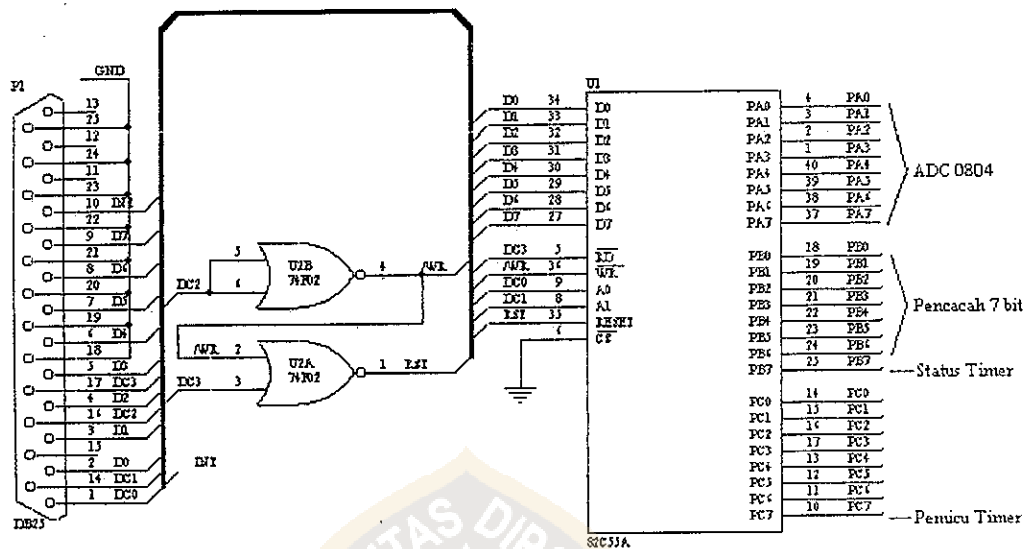
Note: Pins 18-25 are
Ground

Gambar 3.19 Konfigurasi Pin-pin DB25
(Anderson,---)

3.2.6. Programmable Periferal Interface (PPI 8255)

Chip ini dioperasikan pada mode 1 masukan untuk *port A* dan mode 0 masukan untuk *port B* sedangkan *port C* pada mode keluaran sehingga *control word* yang dikirimkan ke PPI adalah B2H. *Port A* dihubungkan dengan D0 – D7 ADC0804, sehingga *port* ini sebagai jalur masuk data hasil konversi ADC0804. *Port B* dihubungkan dengan data keluaran pencacah 7 bit (D0-D6), sedangkan D7 digunakan untuk mengambil status *timer 15* detik. *Port C bit* ke 7 digunakan untuk mengirimkan pulsa yang berguna untuk *reset* pencacah 7 bit dan sinyal *trigger* untuk *timer 15* detik. Sinyal *interrupt* pada *port A* diperoleh dari sinyal

EOC ADC0804, dan digunakan untuk mengirimkan permintaan *interrupt* ke CPU melewati *interrupt* IRQ7 pada adapter printer.



Gambar 3.20 Port Printer Adapter dan PPI 8255

Sinyal-sinyal kontrol pada PPI8255 yaitu A0, A1, RD dan WR masing-masing dihubungkan dengan *port* kontrol printer adapter, sehingga operasi baca dan tulis harus melibatkan *port* kontrol printer adapter.

3.3. Perangkat Lunak

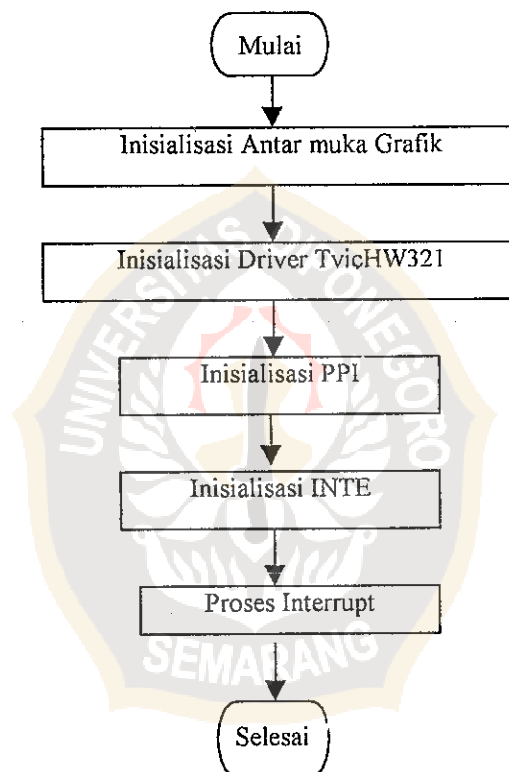
Software dirancang untuk bekerja pada sistem operasi berbasis Windows dengan menggunakan program Delphi versi 3.0. Pemilihan ini dengan pertimbangan kemudahan pembuatan antarmuka grafik serta untuk mengikuti perkembangan teknik pemrograman yang dewasa ini berbasis windows.

Proses penanganan *port* masukan – keluaran pada sistem operasi berbasis windows relatif sulit jika dibandingkan dengan pada sistem operasi berbasis DOS.

Sehingga diperlukan *driver* khusus, yaitu TvicHW321 yang dibuat oleh Victor Ishikeev, pemrogram dari Rusia.

3.3.1. Algoritma Secara Umum

Hal pertama yang dilakukan oleh program adalah melakukan proses inisialisasi .



Gambar 3.21 Diagram Alir Program Secara Umum

Inisialisasi yang pertama dilakukan adalah inisialisasi antar muka grafik yang berupa proses penciptaan tampilan program. Inisialisasi kedua adalah inisialisasi *Driver* yang menangani proses *port* masukan dan *port* keluaran berbasis windows yaitu TvicHW321.

Inisialisasi berikutnya adalah inisialisasi PPI dengan mengirimkan *control Word* B2H dan inisialisasi INTE untuk mengaktifkan *interrupt* pada *port A* dengan mengirimkan *control word* 89H.

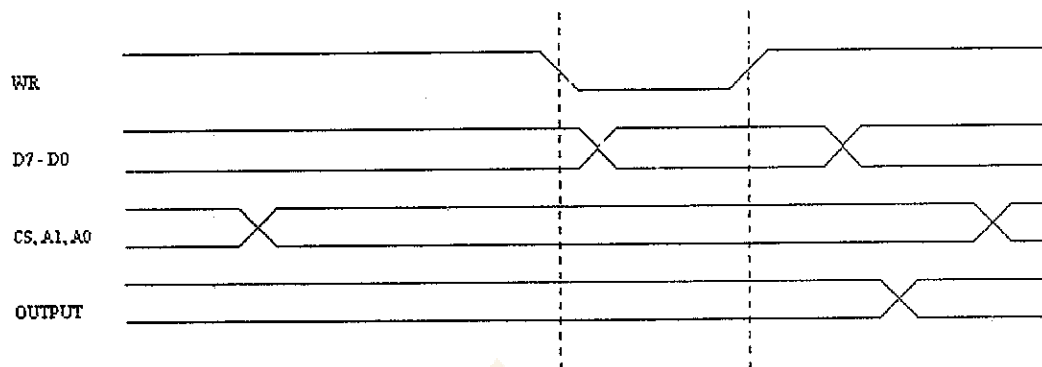
Proses selanjutnya adalah proses penanganan *interrupt* yang meliputi didalamnya proses pengambilan data dari ADC0804 dan pencacah 7 bit yang keduanya dilakukan melalui PPI8255 dan adapter *printer*, serta proses penampilan data yang diperoleh pada layar komputer. Proses ini akan berlangsung terus-menerus sampai program mendapat perintah untuk keluar ke sistem operasi.

Proses pengambilan data pada ADC0804 dan rangkaian pencacah 7 bit oleh komputer melewati adapter *printer* tidak dapat dilakukan dengan menggunakan satu buah operasi masukan pada komputer. Hal ini karena PPI 8255 bekerja sebagai jembatan antara *printer* adapter dan rangkaian pencacah 7 bit serta rangkaian ADC0804, sehingga perintah pembacaan data oleh komputer ke rangkaian ADC0804 maupun rangkaian pencacah 7 bit harus diterjemahkan kedalam proses pembacaan komputer ke PPI 8255. Hal tersebut memerlukan teknik pemrograman tertentu dan sebagai konsekuensinya pada proses baca-tulis oleh komputer ke PPI 8255 memerlukan waktu tambahan. Gambar 3.21 adalah algoritma pemrograman secara umum yang dirancang.

3.3.2. Algoritma Inisialisasi PPI

Operasi inisialisasi PPI adalah merupakan proses tulis ke PPI dengan alamat tujuan *port* kontrol PPI, dengan *control word* B2H. Sehingga kode yang disusun pada prinsipnya merupakan urutan sinyal yang sesuai dengan *timing*

diagram proses tulis ke PPI. Gambar 3.22 adalah *timing diagram* operasi tulis ke PPI, sedangkan Gambar 3.23 adalah diagram alir proses inisialisasi PPI yang dirancang.



Gambar 3.22 Timing Diagram Operasi Tulis PPI
(Hogenboom, 1996)

Bahasa pemrogramman Inisialisasi PPI dengan Borland Delphi 3.0 adalah sebagai berikut:

with VicHW321 do

begin

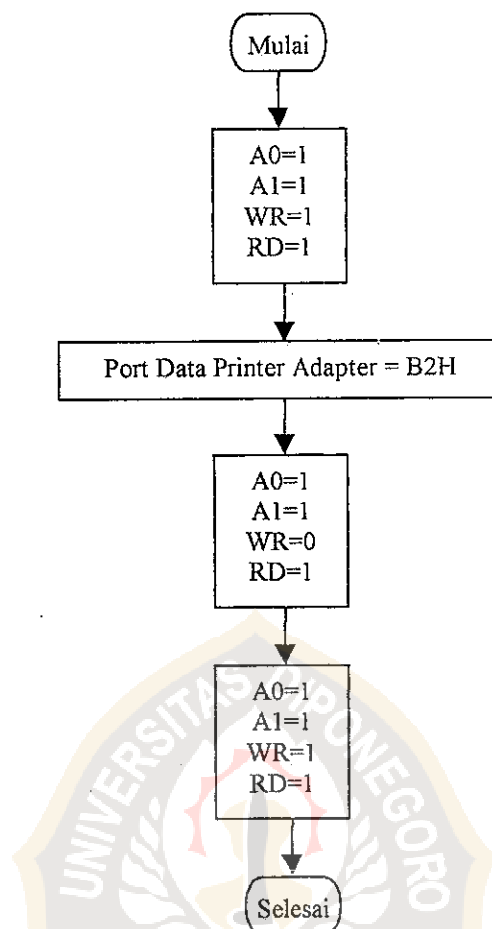
Port[\$378]:= \$B2; // data that will be sent 1011 0010

Port[\$37A]:= \$00; // address is pointed to port control PPI and WR is setted to high

Port[\$37A]:= \$04; // set WR low and keep the address, the formula is 0 or 4

Port[\$37A]:= \$00; // set WR high again and data should have reached control register

end;

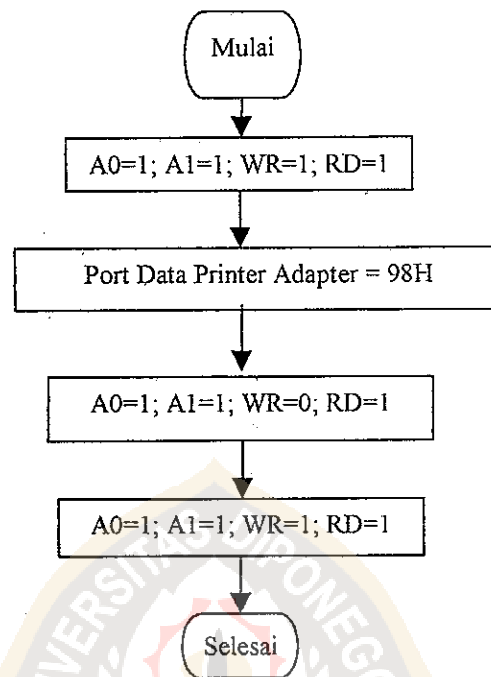


Gambar 3.23 Diagram Alir Inisialisasi PPI

3.3.3. Algoritma Inisialisasi Interrupt.

Pada prinsipnya Inisialisasi Interrupt sama dengan inisialisasi PPI8255, yaitu operasi tulis ke *port* kontrol PPI8255, dengan perbedaan terdapat pada *control word* yang dikirimkan ke *port* kontrol PPI8255, yaitu 89H. Inisialisasi *interrupt* merupakan proses tulis *single bit* ke PC4 PPI8255, dimana PC4 merupakan *bit enable-disable interrupt port A* PPI8255 mode 1. Jika bernilai 0 maka interupsi melalui *port A* tidak dapat dilakukan sedangkan bila bernilai 1

maka interupsi dapat dilakukan melalui *port* A. Gambar 3.24 adalah diagram alir inisialisasi *interrupt* yang dirancang.



Gambar 3.24 Diagram Alir Inisialisasi Interrupt

Bahasa pemrograman inisialisasi *interrupt* dengan Borland Delphi 3.0 adalah sebagai berikut:

```

// Aktivasi INT pada port A PPI dengan Enable INTE A
with VicHW321 do
begin
  Port[$378]:= $09; // data that will be sent 0000 1001
  Port[$37A]:= $00; // address is pointed to port C PPI and WR is setted to high
  Port[$37A]:= $04; // set WR low and keep the adress, the formula is 0 or 4
  Port[$37A]:= $00; // set WR high again and data should have reached control register
end;
  
```

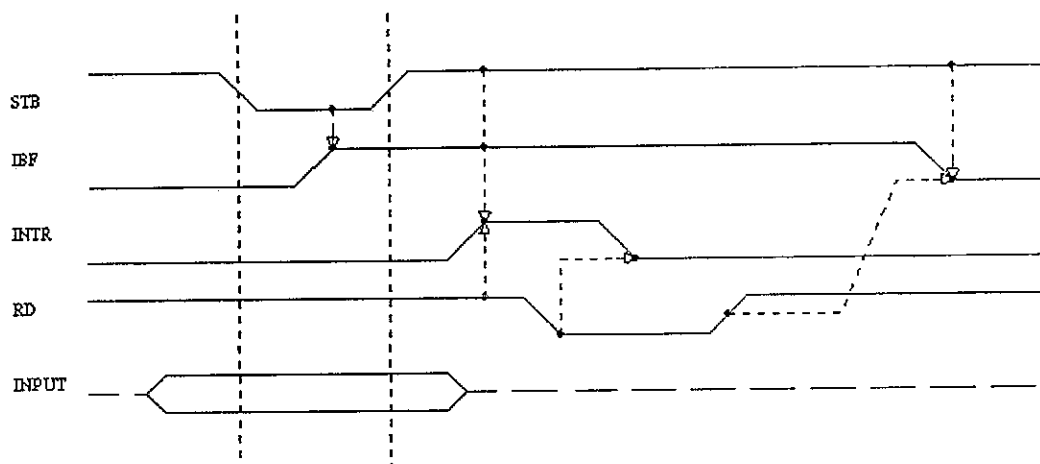
3.3.4. Algoritma Pembacaan Data Dari PPI

Pada rancangan perangkat keras *port A* PPI8255 dihubungkan dengan keluaran ADC0804 dan *port B* PPI8255 dihubungkan dengan keluaran pencacah 7 *bit*, sehingga operasi yang dilakukan pada PPI8255 pada saat program berjalan adalah operasi baca. Terdapat dua operasi baca yang dilakukan yaitu operasi baca mode 1 (pada *port A* PPI8255) dan operasi baca mode 0 (pada *port B* PPI8255).

3.3.4.1. Algoritma Pembacaan Data ADC (*Port A*)

Proses pembacaan nilai dari *Port A* PPI adalah proses pengambilan nilai keluaran ADC0804. Proses pembacaan *port A* PPI8255 dimulai dengan diterimanya sinyal EOC (*End Of Conversion*) dari ADC0804, dimana sinyal ini adalah indikator bahwa ADC0804 telah selesai mengkonversikan nilai tegangan analog yang berada pada masukan analog ADC0804. Sinyal ini selanjutnya diterima sebagai sinyal *interrupt* pada *port printer* adapter, yaitu pada IRQ 7, sehingga program untuk membaca nilai ADC (melalui *port A* PPI8255) terdapat dalam *routine interrupt* IRQ 7 pada komputer, yang ditangani oleh driver TvicHW321 (Ishikeev, 1999).

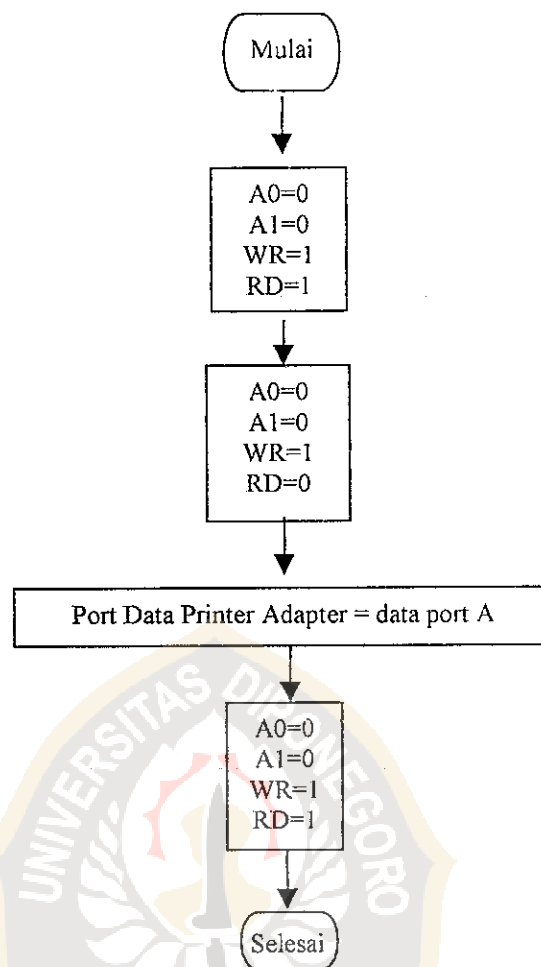
Gambar 3.25 adalah *timing diagram* proses pembacaan pada Mode 1 PPI8255 yang direalisasikan menjadi kode program untuk membaca *port A* PPI8255. Gambar 3.26 adalah diagram alir proses baca *port A* PPI8255.



Gambar 3.25 Timing Diagram Operasi Baca Mode 1
PPI8255 (Hogenboom, 1996)

Bahasa pemrograman pembacaan *port* A PPI8255 dengan Borland Delphi 3.0 adalah sebagai berikut:

```
// proses pengambilan nilai dari port A PPI
with VichW321 do
begin
    Port[$37A]:= $03; // point to address port A PPI and set RD high
    Port[$37A]:= $0B; // set RD low and keep the adress point to port A PPI
    LPTReadMode:= True;
    dataInt:= port[$378]; // data is ready on port $378, and have come into CPU Register
    Port[$37A]:= $03; // set RD high, address unchanged
end;
```



Gambar 3.26 Diagram Alir Proses Baca
Port A PPI8255

3.3.4.2. Algoritma Pembacaan Data Pencacah 7 Bit

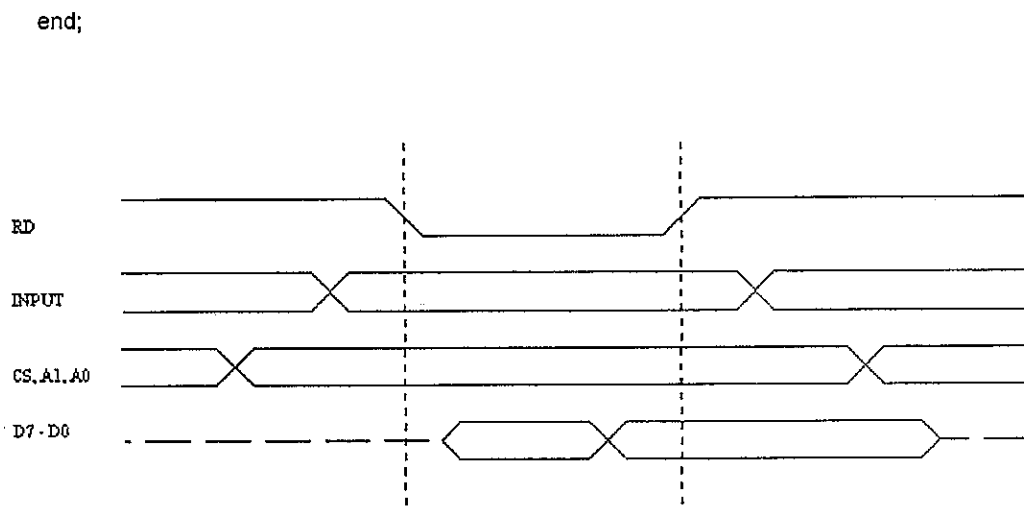
Keluaran pencacah 7 bit terhubung dengan *port* B PPI8255, sehingga untuk mengambil nilai pencacah, maka dilakukan operasi baca *port* B PPI8255. Proses baca *port* B PPI8255 direalisasikan dengan cara menyusun kode program baca *port* B dan menyisipkannya pada proses baca *port* A. Karena nilai keluaran pencacah 7 bit akan valid setelah 15 detik maka kode diatur untuk dieksekusi setiap interval waktu tertentu. Interval waktu ini ditentukan oleh posisi

penggambaran nilai *Port A* pada layar komputer. Jika penggambaran nilai *port A* PPI8255 pada layar komputer sudah mencapai tepi paling kanan maka kode baca *port B* PPI8255 dieksekusi.

Nilai yang diperoleh pada proses baca *port B* PPI8255 tidak akan selalu ditampilkan pada layar komputer sebagai jumlah denyut nadi melainkan hanya pada saat bit ke-7 pada data hasil baca *port B* PPI8255 bernilai 0. Nilai 0 ini menunjukkan bahwa *timer* 15 detik telah menyelesaikan prosesnya. Nilai denyut nadi yang diperoleh melalui pembacaan *port B* PPI8255 dikalikan dengan empat untuk mendapatkan jumlah denyut per menit. Proses penampilan nilai denyut nadi diikuti dengan proses pengiriman sinyal *clear counter* yang sekaligus merupakan sinyal *start timer* 15 detik. Hal ini menjamin bahwa proses proses penghitungan denyut nadi hanya terjadi tiap 15 detik, dan memastikan bahwa penghitungan dimulai dari nilai 0. Gambar 3.27 adalah gambar *timing diagram* pembacaan *port mode 0* PPI8255, dan gambar 3.28 adalah diagram alir pembacaan data dari *port B* PPI8255.

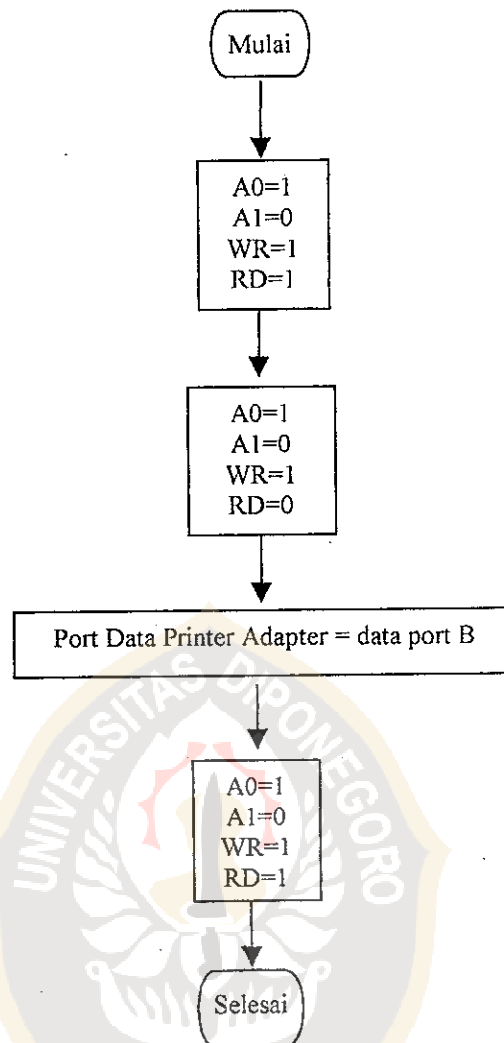
Bahasa pemrogramman pembacaan *port B* PPI8255 dengan Borland Delphi 3.0 adalah sebagai berikut:

```
// proses pengambilan counter detak jantung
with VichW321 do
begin
    Port[$37A]:= $02;           // point to address port B PPI and set RD high
    Port[$37A]:= $0A;          // set RD low and keep the adress point to port B PPI
    LPTReadMode:= True;       // LPT Read mode allowed
    StatusCount:= port[$378];  // data is ready on port $378, and have come into CPU Register
    Port[$37A]:= $02;
```



Gambar 3.27 Timing Diagram Operasi Baca Mode 0
PPI8255 (Hogenboom, 1996)





Gambar 3.28 Diagram Alir Proses Baca
Port B PPI8255