

3100098010075

MONITORING DAN PENGONTROLAN PEMAKAIAN DAYA LISTRIK DENGAN MENGUNAKAN IBM PC FASILITAS LPT

TUGAS AKHIR

Disusun oleh :

AGUS SUNYOTO

NRP : 2291.100.015

R
621.398.1
Sun
m-1

1997



JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
1997



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

**MONITORING DAN PENGONTROLAN
PEMAKAIAN DAYA LISTRIK DENGAN
MENGUNAKAN IBM PC FASILITAS LPT**

TUGAS AKHIR

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

Pada

Bidang Studi Elektronika

Jurusan Teknik Elektro

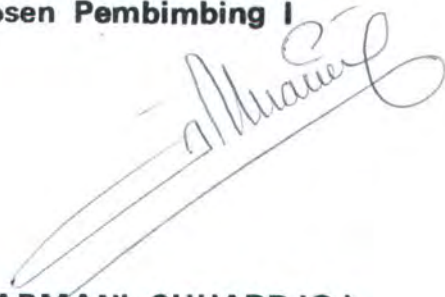
Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

S u r a b a y a

Mengetahui / Menyetujui

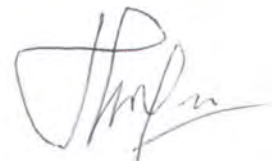
Dosen Pembimbing I



(Ir. HARMANI SUHARDJO)

NIP. 130 368 611

Dosen Pembimbing II



(PUJIONO, ST)

NIP. 132 094 793

S U R A B A Y A

Pebruari, 1997

“ Tidak ada Daya dan Kekuatan kecuali dari Allah Swt ”

ABSTRAK

ABSTRAK

Paralel port (LPT) pada IBM PC lebih banyak digunakan untuk pemakaian printer. Untuk itu dimanfaatkan LPT untuk aplikasi lain. Pada tugas akhir ini memanfaatkan LPT untuk monitoring dan pengontrolan pemakaian daya listrik. Penggunaan LPT pada alat ini untuk memudahkan pemasangan tanpa perlu membongkar komputer, bila dibandingkan dengan penggunaan interface pada slot komputer

Alat ini akan dapat melakukan monitoring besarnya daya listrik yang dipakai pada setiap saat pada suatu gedung yang dibagi menjadi blok-blok. Alat ini melakukan pengontrolan secara otomatis daya beban yang aktif dan jika dayanya lebih besar dari yang diijinkan maka secara otomatis akan mengaktifkan beban pada blok berdasarkan prioritasnya. Jadi blok-blok akan ditentukan prioritasnya.

Daya pada setiap blok diukur oleh rangkaian pengubah besaran daya ke tegangan dan beda phasanya untuk menentukan faktor daya. Kemudian hasil pembacaan akan diubah ke dalam bentuk sinyal digital. Data tersebut masuk ke LPT pada IBM PC. Proses pengolahan data akan dilakukan oleh IBM PC. Arus yang mengalir pada blok dapat diputus dengan kontrol lewat IBM PC. Pada alat ini dapat dilihat besar dan persentase pemakaian daya antar blok. Pembatasan daya dan prioritas dari blok-blok dapat diubah-ubah.



KATA PENGANTAR

KATA PENGANTAR

Syukur Alhamdulillah penyusun panjatkan kehadiran Allah Swt atas segala rahmat dan hidayah-Nya yang telah dilimpahkan sehingga terselesaikannya Tugas Akhir yang berjudul :

MONITORING DAN PENGONTROLAN PEMAKAIAN DAYA LISTRIK DENGAN MENGGUNAKAN IBM PC FASILITAS LPT

Tugas Akhir dengan beban 6 SKS ini merupakan salah satu persyaratan untuk memperoleh gelar sarjana Teknik Elektro pada Bidang Studi Elektronika, Jurusan Teknik Elektro, Fakultas Teknologi Industri, Institut Teknologi Sepuluh Nopember Surabaya.

Dengan terbatasnya waktu serta kemampuan penyusun dalam mngerjakan Tugas Akhir ini, penyusun menyadari bahwa penyusunan Tugas Akhir ini jauh dari sempurna dan masih banyak kekurangan. Penyusun mengucapkan banyak terima kasih atas saran dan kritik yang telah diberikan. Akhirnya penyusun berharap semoga Tugas Akhir ini dapat bermanfaat bagi para pembaca.

Surabaya, Februari 1997

Penyusun



UCAPAN TERIMA KASIH

UCAPAN TERIMA KASIH

Dalam mengerjakan dan menyusun Tugas Akhir ini, banyak bantuan dan bimbingan baik berupa materiil dan moril dari berbagai pihak, untuk itu penyusun mengucapkan terima kasih kepada :

1. Bapak Ir. Harmani S, selaku dosen pembimbing I dan dosen wali yang telah memberi banyak bimbingan serta saran-saran yang berguna dalam penyusunan Tugas Akhir ini.
2. Bapak Pujiono, ST, selaku dosen pembimbing II yang telah memberi banyak bimbingan serta saran-saran yang berguna dalam penyusunan Tugas Akhir ini.
3. Bapak Ir. Soetikno, selaku koordinator bidang studi Elektronika yang telah banyak memberi pembinaan.
4. Bapak Ir. Teguh Yuwono, selaku Ketua Jurusan Teknik Elektro FTI - ITS yang telah memberi bantuan terhadap lancarnya Tugas Akhir ini.
5. Bapak, Ibu serta seluruh keluarga di rumah yang telah memberikan doa dan dorongan semangat.
6. Rekan-rekan di Lab. RL, E-31, Riki beserta teman-temannya dan seluruh warga Elektronika yang telah banyak memberikan bantuan.
7. Eka Sujiatiningsih, atas banyak perhatian yang telah diberikan.
8. Seluruh rekan-rekan dan Ibu kost di Keputih Utara 69 Surabaya, yang membantu lancarnya terhadap selesainya tugas akhir ini.



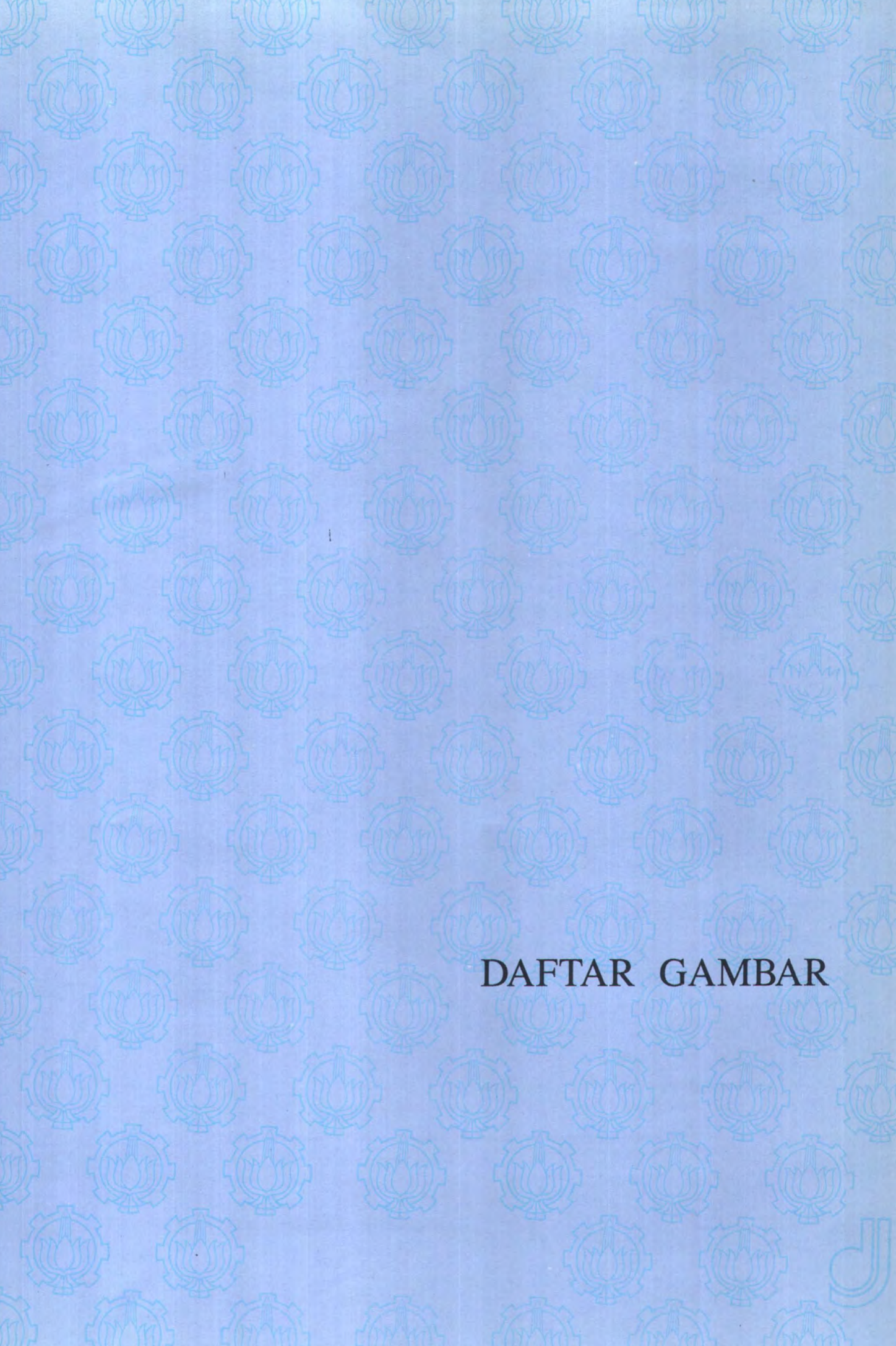
DAFTAR ISI

DAFTAR ISI

JUDUL	i
HALAMAN PENGESAHAN	ii
ABSTRAK	iii
KATA PENGANTAR	iv
UCAPAN TERIMA KASIH	v
DAFTAR ISI	vi
DAFTAR GAMBAR	ix
DAFTAR TABEL	xi
BAB I. PENDAHULUAN	1
I.1. Latar Belakang	1
I.2. Permasalahan	1
I.3. Pembatasan Masalah	2
I.4. Tujuan	2
I.5. Metodologi Penelitian	2
I.6. Sistematika	3
I.7. Relevansi	4
BAB II. TEORI PENUNJANG	5
II.1 Daya dan Faktor Daya	5

II.2 Pengubah Analog ke Digital	9
II.2.1 ADC MAX180ACPL	11
II.3 Pengubah Besaran Daya ke Tegangan	13
II.4 Multiplier MC1495L	14
II.5.Port Paralel Input Output.....	16
II.5.1.Synchronous dan Asynchronous	16
II.5.2.Perlunya teknik transmisi Asynchronous	17
II.5.3.Handshaking	18
II.5.4.Programed I/O dan Interupt I/O	19
II.5.6.Port Paralel IBM PC	21
II.6. Transistor Switching	25
BAB III. PERENCANAAN ALAT	28
III.2. Perencanaan Perangkat Keras	28
III.2.1.Pengubah daya ke Tegangan	29
III.2.2.Beda Phase	31
III.2.3.Rangkaian Multiplekser Analog	33
III.2.4.Rangkaian Switch Tegangan Jala-Jala	34
III.2.5.Rangkaian Pengubah Analof ke Digital	35
III.2.6.Hubungan ke paralel port (LPT)	38
III.3. Perencanaan Perangkat Lunak	39
III.3.1 Program Kerja ADC dan pengambilan Data	39
III.3.2.Program Pengolahan Data	40

III.3.3 Program Pembatasan Daya tiap Blok	42
III.3.4. Program Kontrol Switch Tegangan Jala-jala	43
III.3.5. Program Beda Phase	44
BAB IV PENGUJIAN DAN PENGUKURAN	45
IV.1. Kalibrasi Pengubah Daya ke Tegangan	45
IV.2. Pengujian Alat	46
IV.2.1 Pengujian Pengubah Daya ke Tegangan	46
IV.2.2. Pengujian ADC	47
IV.3. Pengukuran	48
BAB V. PENUTUP	52
V.1. Kesimpulan	52
V.2. Saran-saran	53
DAFTAR PUSTAKA	54
LAMPIRAN	
LAMPIRAN A1 - A4 : GAMBAR RANGKAIAN	
LAMPIRAN B1 - B23 : LISTING PROGRAM	



DAFTAR GAMBAR

DAFTAR GAMBAR

GAMBAR	HALAMAN
2.1. Beban Tahanan dan Daya arus bolak-balik	6
2.2. Beban Reaktif dan Daya arus bolak-balik	7
2.3. Komponen arus dalam rangkaian bolak-balik	9
2.4. Blok Diagram MAX180ACPL	11
2.5. Blok Diagram pengubah besaran Daya ke tegangan	13
2.6. Blok Diagram MC1495L	15
2.7. Programmed I/O	20
2.8. Fungsi pin-pin paralel port	22
2.9. Paralel port DB25.....	25
2.10. Bentuk gelombang transistor switching	26
3.1. Blok Diagram Rangkaian	29
3.2. Pengubah Daya ke Tegangan	30
3.3. Rangkaian Deteksi Beda Phase	32
3.4. Pulsa pada rangkaian beda phase	33
3.5. Multiplekser analog	33
3.6. Rangkaian Switch Tegangan Jala-Jala	34
3.7. Rangkaian ADC	35
3.8. Timing Diagram ADC MAX180ACPL Mode 1.....	36

3.9. Komunikasi data dari ADC ke LPT	37
3.10. Flowchart pengambilan data ADC	40
3.11. Flowchart proses pengolahan data	41
3.12. Flowchart Pembatasan Daya tiap blok	43
3.13. Kontrol Switch Tegangan Jala-jala	43
3.14. Flowchart Beda Phase	44
4.1 Tampilan Menu dan daya tiap blok	49
4.2. Tampilan Konfigurasi LPT	49
4.3. Tampilan Prioritas blok	50
4.4. Tampilan Persentase Pemakaian	50
4.5. Tampilan Batas Daya	51



DAFTAR TABEL

DAFTAR TABEL

TABEL	HALAMAN
2.1. Pin-pin dari MAX180ACPL	12
2.2. Format Data ouput MAX180ACPL	12
2.3. Alamat port LPT	21
3.1. Bit Kontrol Switch Tegangan Jala-jala	35
3.2. Pemakaian bit LPT1 port 378h	38
3.3. Pemakaian bit LPT1 port 379h	38
3.4. Pemakaian bit LPT1 port 37Ah	38
4.1. Pengujian pengubah daya ke tegangan	47
4.2. Pengujian ADC MAX180ACPL	47
4.3. Pengukuran Daya Listrik	48

BAB I

PENDAHULUAN

I.1.LATAR BELAKANG.

Pesatnya perkembangan teknologi elektronika telah memberikan suatu kemudahan bagi manusia dalam menjalankan aktivitas sehari-hari , mulai dari lingkungan rumah tangga, gedung-gedung sampai ke lingkungan industri.

Pemakaian daya listrik pada suatu gedung besar yang mempunyai banyak ruangan atau pada industri yang memakai banyak daya listrik, seringkali terjadi pemakaian daya listrik diluar keperluan semestinya, atau pemakaian daya listrik semaunya sendiri untuk hal-hal yang kurang penting, sehingga dapat terjadi pemborosan daya listrik yang akhirnya mengakibatkan pengelola atau pemiliknya menjadi besar pembayaran listriknya.

Bila ternyata beban total yang dipakai melebihi total daya listrik yang diijinkan maka secara manual harus memilih beban mana yang terasa kurang perlu harus dimatikan dan beban mana yang terasa penting yang harus tetap dialiri aliran listrik.

I.2.PERMASALAHAN

Dari latar belakang masalah tersebut, maka dalam tugas akhir ini akan berusaha untuk mengembangkan dan membuat sistem untuk dapat melakukan



BAB I

PENDAHULUAN

monitoring pemakaian daya listrik pada suatu area pada setiap saat. Sistem ini dapat mengatur beban yang mana yang harus hidup dan harus mati bila beban melebihi daya total .

I.3.PEMBATASAN MASALAH

Pada tugas akhir ini masalah dibatasi pada untuk pengukuran daya nyata pada listrik Arus bolak-balik (AC) untuk satu phase.

I.4.TUJUAN

Merencanakan dan membuat suatu sistem peralatan untuk dapat melakukan monitoring dan pengontrolan daya. Instrumen akan melakukan kontrol secara otomatis bila ternyata daya yang akan dipakai melebihi batas daya yang diijinkan dengan prioritas beban mana yang perlu hidup terlebih dahulu.

I.5.METODOLOGI

Untuk mencapai tujuan diatas, langkah-langkah yang harus ditempuh adalah sebagai berikut :

- Studi Literatur, mengenai teori yang menunjang pembuatan alat, seperti teori pengukuran daya listrik bolak-balik, teori pengubahan dari besaran daya ke tegangan, port paralel (LPT) dari komputer dan komponen penunjang lainnya.
- Studi Lapangan
- Perencanaan Dan Disain Alat

- Pengetesan Alat
- Pengambilan Data
- Pengolahan Data
- Penulisan Naskah Tugas Akhir.

I.6.SISTEMATIKA

Sistematika pembahasan pada penyusunan Tugas Akhir ini dibagi dalam beberapa bagian, yaitu :

- BAB I** : **Pendahuluan**, pada bab ini membahas tentang latar belakang, permasalahan, pembatasan masalah, tujuan , metodologi penelitian, dan sistematika.
- BAB II** : **Teori Penunjang**, pada bab ini membahas tentang teori penguku ran daya listrik dan faktor daya, paralel port (LPT), pengubah besaran daya ke tegangan , Multipler Analog, pengubah analog ke digital. dan transistor switching.
- BAB III** : **Perencanaan Alat**, pada bab ini membahas tentang perencanaan alat yang terbagi dalam 2 bagian :
 - Perencanaan perangkat keras yang terbagi dalam blok pengubah besaran daya ke tegangan, blok beda phase, blok ADC dan multipleks input, rangkaian pemutus arus.

- Perencanaan perangkat lunak , pada bab ini membahas tentang perencanaan program yang meliputi pembuatan program untuk monitoring daya listrik dengan Borland Delphi.

- **BAB IV : Pengujian dan Pengukuran**, pada bab ini membahas tentang cara mengkalibrasi alat dan melakukan pengukuran untuk melihat hasil kalibrasi. Kalibrasi meliputi kalibrasi pengubah daya ke tegangan. Untuk pengujian dilakukan uji coba peralatan untuk mengukur besar pemakaian daya listrik.
- **BAB V : Penutup** , pada bab ini berisi kesimpulan dari perencanaan dan pembuatan alat serta saran-saran untuk pengembangan selanjutnya.

I.7.RELEVANSI.

Tugas akhir ini diharapkan berguna untuk dapat memberikan bantuan terhadap hal untuk menghemat dan memonitor penggunaan daya listrik pada suatu area tertentu pada gedung misalnya.



BAB II

TEORI PENUNJANG

BAB II

TEORI PENUNJANG

II.1. Daya dan Faktor Daya

Pada rangkaian arus searah daya, sama dengan perkalian antara arus dan tegangan pada beban. Untuk arus bolak-balik satu fase, daya rangkaian sama dengan perkalian dari harga arus dan tegangan sesaat. Jika arus dan tegangan sefase, maka daya rata-rata dalam satu periode sama dengan perkalian dari arus dan tegangan efektif. Jika ada reaktansi, maka arus dan tegangan menjadi tidak sefase, maka timbulah beda fase antara arus dan tegangan.

Perkalian harga arus dan tegangan efektif dalam tegangan bolak-balik dinyatakan dalam VoltAmpere (VA). Daya nyata (actual power) dinyatakan dalam watt.

Jika tegangan dinyatakan dengan fungsi sinus maka

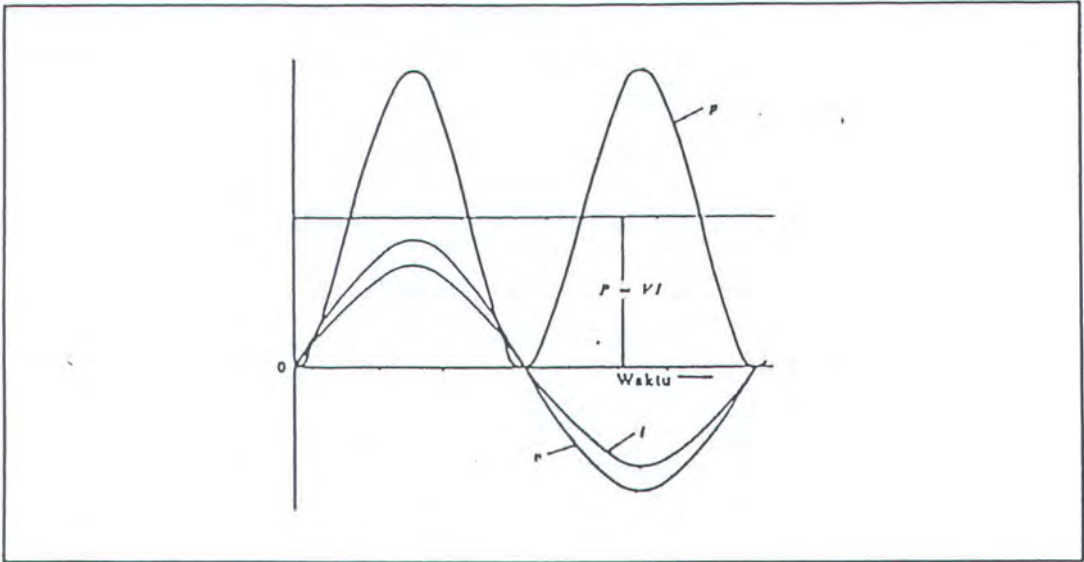
$$v = V_m \sin \omega t \dots\dots\dots(1)$$

dan R adalah tahanan pada beban, maka arus pada beban adalah:

$$i = \frac{V_m}{R} \sin \omega t \dots\dots\dots(2)$$

Dan daya (p) adalah:

$$p = V_m I_m \sin^2 \omega t \dots\dots\dots(3)$$



Gambar 2.1.¹
 Beban Tahanan dan Daya arus bolak-balik

Harga efektif dari daya (p) dalam satu periode adalah:

$$p = \frac{Vm Im}{2} = VI = I^2R = \frac{V^2}{R} \dots\dots\dots(4)$$

dimana V dan I adalah harga efektif dari v dan i.

Persamaan ini punya bentuk yang sama untuk arus searah. Dengan demikian maka p disebut sebagai daya dalam arus searah.

Apabila beban mempunyai reaktansi misalnya induktansi L , maka:

$$i = \frac{Vm}{\omega L} \text{Sin}\left(\omega t - \frac{\pi}{2}\right) = Im \text{Sin}\left(\omega t - \frac{\pi}{2}\right) \dots\dots\dots(5)$$

$$p = Vm Im \text{Sin}(\omega t) \text{Sin}\left(\omega t - \frac{\pi}{2}\right) = \frac{Vm Im}{2} \text{Sin}2\omega \dots\dots\dots(6)$$

¹ Dr.SoedjanaSapii,Dr.Osamu Nishino, "Pengukuran dan Alat-alat ukur listrik",Pradnya Paramita , Jakarta, 1975,hal 60.

Jika beban merupakan tahanan dan reaktansi , maka dapat dinyatakan sebagai berikut:

$Z = R + jx$ dimana R adalah tahanan dan x merupakan reaktansi.

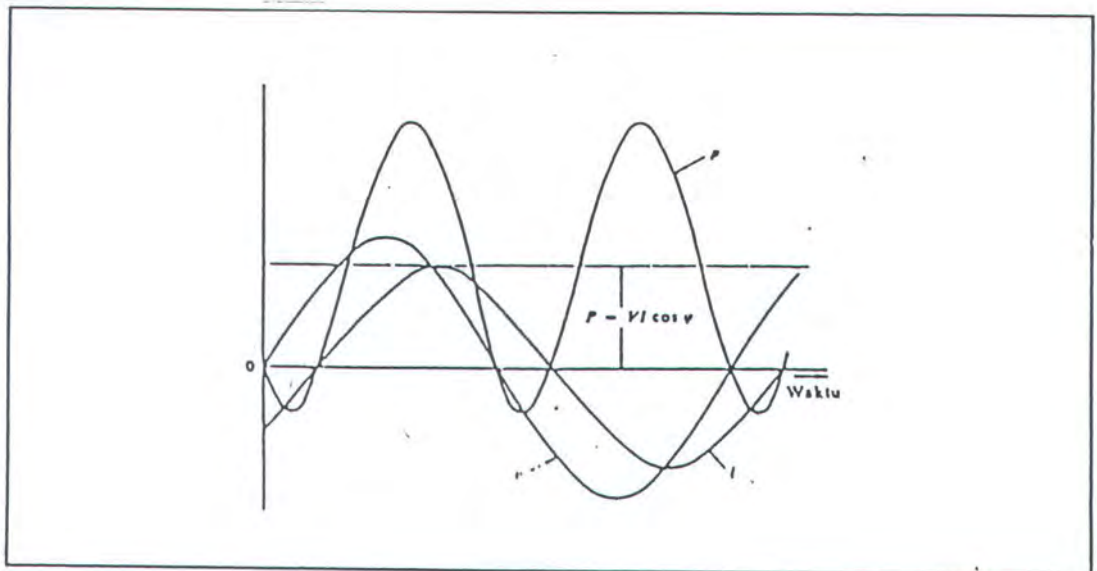
$$i = I_m \sin \omega t \dots\dots\dots(7)$$

$$p = V_m I_m \sin \omega t \sin(\omega t - \Phi) \dots\dots\dots(8)$$

$$\tan \Phi = \frac{X}{R} \dots\dots\dots(9)$$

$$p = VI \cos \omega(1 - \cos 2\omega t) - V I \sin \Phi \sin 2\omega t \dots\dots\dots(10)$$

Hubungan ini dapat dinyatakan pada gambar 2.2 dan harga rata-rata dari daya P adalah sebagai berikut:



Gambar 2.2²
Beban Reaktif dan Daya arus bolak-balik

² Ibid. , hal 61

$$P = VI \cos \Phi \dots\dots\dots(11)$$

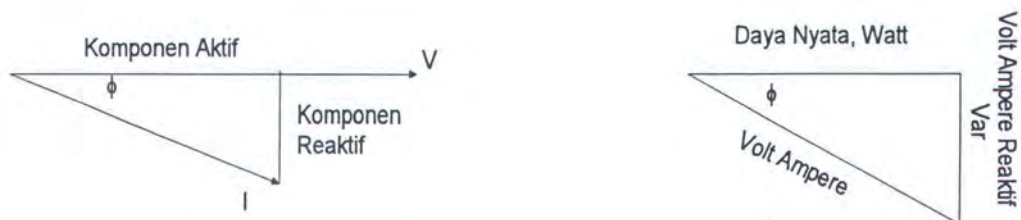
$$\cos \Phi = \frac{R}{\sqrt{R^2 + X^2}}, \text{ maka } \dots\dots\dots(12)$$

$$P = I^2 R = \frac{V^2}{R} \dots\dots\dots(13)$$

Harga efektif dari daya dalam arus bolak-balik dengan beban umum, dapat dinyatakan sebagai hasil perkalian arus dan tegangan efektif dikalikan cosinus beda fase antara tegangan dan arus yang mengalir pada beban. Daya tersebut disebut dengan daya nyata. $\cos \Phi$ disebut faktor daya, yang mempunyai harga antara 0 sampai dengan 1 pada beban yang mengandung tahanan dan reaktansi,, bergantung pada harga relatif dari tahanan dan reaktansi dalam rangkaian.. $\cos \Phi$ sama dengan 1, merupakan beban resistif dimana arus dan tegangan sefase. sedang $\cos \Phi$ sama dengan 0 dimana beda fase 90 seperti pada rangkaian bersifat rangkaian induktif atau kapasitif. Faktor daya dapat didefinisikan sebagai perbandingan antara daya nyata dalam watt dengan VoltAmpere pada tegangan bolak-balik. Harga faktor daya bergantung pada besarnya beda fase antar arus dan tegangan pada beban.

Arus yang mengalir dalam rangkaian bolak-balik dapat dianggap terdiri dari dua komponen yang sefase dengan tegangan dan komponen yang berbeda fase 90 dengan tegangan . Komponen yang sefase disebut komponen aktif karena harga ini jika dikalikan dengan tegangan memberikan daya nyata dari rangkaian . Komponen yang tidak sefase disebut komponen reaktif. Perkalian komponen reaktif dari arus

dan tegangan disebut daya reaktif atau VoltAmpere Reaktif dan diukur dengan satuan Var.



Gambar 2.3³
Komponen arus dalam rangkaian bolak-balik

Dalam gambar 2.3. cosinus sudut fase Φ adalah :

$$\text{Cos}\Phi = \frac{I_{aktif}}{I} \dots\dots\dots(14)$$

Hubungan antara daya dalam watt, voltampere dan voltampere reaktif adalah seperti pada gambar 2.3. Oleh karena itu voltampere sama dengan VI, daya nyata adalah VI Cos Φ , dan Voltampere reaktif adalah VI Sin Φ .

$$\text{oltampere} = \sqrt{\{(Dayanyata)^2 + (VAreaktif)^2\}} \dots\dots\dots(15)$$

II.2.PENGUBAH ANALOG KE DIGITAL.

Rangkaian pengubah analog ke digital (ADC) digunakan untuk merubah besaran analog ke besaran digital dalam bentuk bit-bit sehingga pengolahannya akan lebih mudah. Pengubah analog ke digital merupakan bagian utama dari sistim akuisisi data. Ada beberapa jenis sistem ADC yaitu :

³ Gunawan ,Hanapi, Ir.,Drs.,Eugene C Lister, Mesin dan Rangkaian Listrik, Penerbit Erlangga, Jakarta 1988,hal 162

- Single Ramp Integrating
- Dual Slope Integrating
- Successive Approximation Register
- Flash (parallel comparator)
- Half Flash.

Beberapa faktor yang perlu diperhatikan dalam pemilihan ADC, antara lain:

○ **Resolusi**

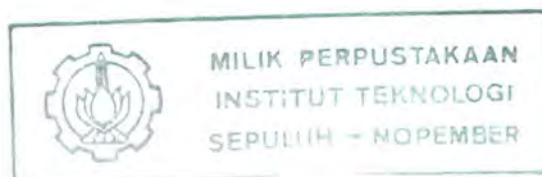
Resolusi merupakan spesifikasi terpenting dari ADC, yaitu kemampuan perubahan terkecil dari tegangan masukan untuk mengubah kode biner yang sekarang ke kode biner berikutnya pada keluaran ADC.

○ **Akurasi.**

Akurasi adalah jumlah dari semua kesalahan, misalnya kesalahan non-linearitas, skala penuh, skala nol dan lain-lain. Akurasi juga menyatakan perbedaan antara tegangan masukan analog secara teoritis terhadap tegangan masukan nyata yang menghasilkan tegangan kode biner tersebut.

○ **Waktu Konversi**

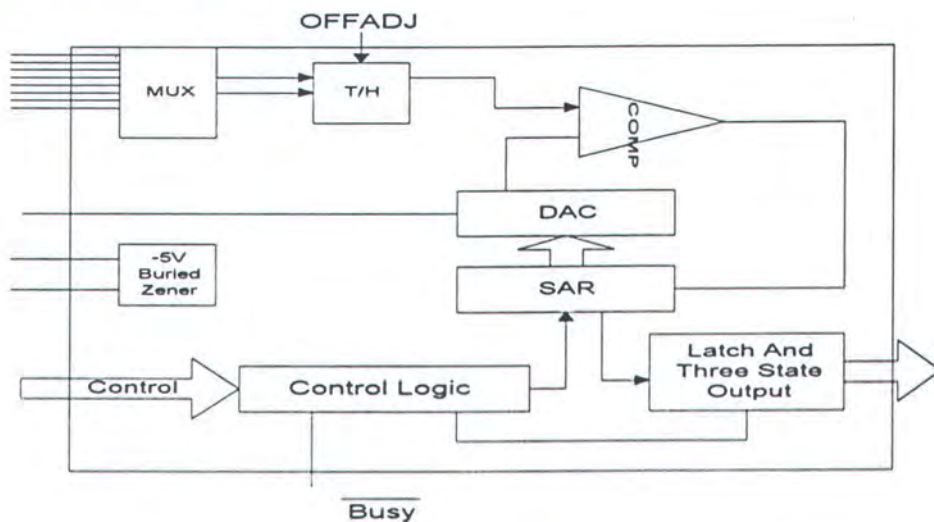
Waktu yang dibutuhkan untuk mengubah besaran analog menjadi bentuk digital untuk setiap sampelnya atau waktu yang diperlukan untuk melakukan satu konversi.



II.2.1.ADC MAX180ACPL

ADC MAX180ACPL merupakan jenis ADC 12 bit dengan dengan mempunyai 8 kanal input multipleks, mempunyai akurasi linearitas $1/2$ LSB, Track and Hold mempunyai bandwidth yang tinggi, referensi zener dengan drift yang rendah, serta kecepatan konversi yang tinggi. ADC ini merupakan keluaran dari Maxim dengan konsumsi daya yang rendah (110mW). MAX180ACPL dapat dikonfigurasi untuk bipolar dan unipolar konversi dan input single ended atau diferensial. Dapat digunakan antara referensi internal atau referensi eksternal. Offset dan referensi internal dapat diatur untuk mendapatkan gain sistem dan offset error nol.

MAX180ACPL menggunakan sistem Successive Approximation. Blok diagram adalah sebagai berikut:



Gambar 2.4.⁴
Blok Diagram MAX180ACPL

⁴ Maxim , MAX180/181 Data Sheet, MaximIntegrated products,USA, 1994,hal 1

Fungsi pin-pin dari MAX180ACPL adalah sebagai berikut:

Tabel 2.1⁵
Pin-pin dari MAX180ACPL

Nama	Nomor Pin	Keterangan
AIN0 - AIN7	1 - 8	Input Analog (0 s/d +5V) Unipolar (-2.5 s/d +2.5) Bipolar
REFIN	9	Referensi Input
AGND	10	Analog ground
REFOUT	11	-5V output Referensi
REFADJ	12	-5V Pengatur referensi. Hubung ke VDD bila tak perlu
OFFADJ	13	Pengatur offset. Hubung ke VDD bila tak perlu
MODE	14	Mode Interface
Vss	15	Supply negatif
D11 - D8	16 - 19	Data output , MSB = D11
DGND	20	Digital Ground
D7 - D0	21 - 28	Data output, LSB = D0
CLKIN	29	Input clock
HBEN	30	High Byte Enable Input
RD	31	Read input
WR	32	Write input
CS	33	Chip select input
Busy	34	Busy output
DIFF	35	Single Ended Mode. DIFF=0 . Differential Mode. DIFF=1
BIP	36	Unipolar mode. BIP=0, Bipolar BIP=1
A0 - A2	37 - 39	Alamat Kanal Multiplekser
VDD	40	Supply Positif (+5 V)

Format output data daru ADC MAX180ACPL adalah sebagai berikut :

Tabel 2.2.⁶
Format Data ouput MAX180ACPL

HBEN	Pin 16	Pin 17	Pin 18	Pin19	Pin 21	Pin 22	Pin 23	Pin 24	Pin 25	Pin 26	Pin 27	Pin 28
0	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
1	D11	D10	D9	D8	0	0	0	0	D11	D10	D9	D8

⁵ Ibid, hal 5

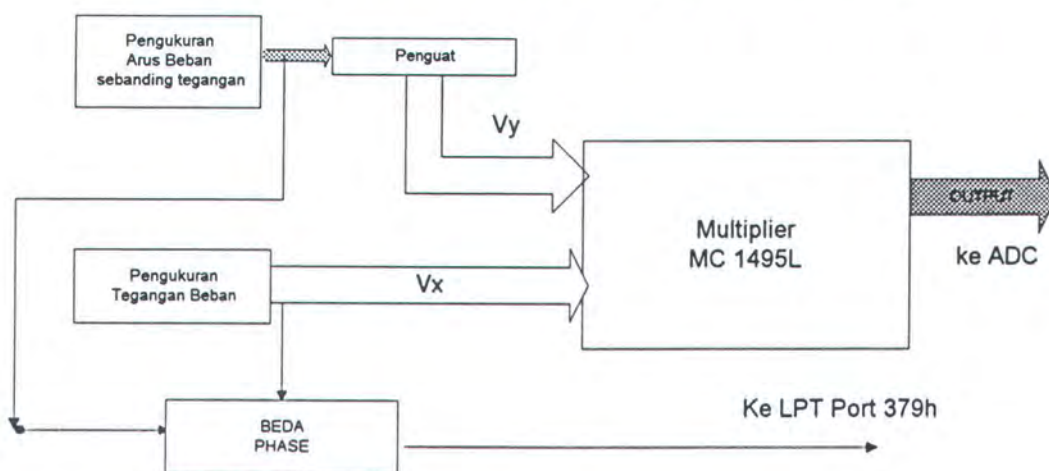
⁶ Ibid hal 14

II.3. PENGUBAH BESARAN DAYA KE TEGANGAN.

Blok Rangkaian ini digunakan untuk mengubah besarnya daya pada beban yang dialiri tegangan bolak-balik menjadi besaran tegangan searah. Untuk mengetahui besarnya daya nyata tegangan bolak-balik digunakan *four-quadrant multiplier*⁷ yang mana keluarannya berupa tegangan.

Pada alat ini mampu mengukur arus yang mengalir pada beban dan tegangan pada beban secara terus menerus dan bersamaan. Selanjutnya kedua harga tersebut (arus dan tegangan) dikalikan dengan sebuah IC *four-quadrant multiplier* yaitu IC MC 1495L. Dengan catatan harga arus diubah ke dalam tegangan. Dimana keluarannya berupa tegangan yang menyatakan besarnya perkalian harga arus dan tegangan pada beban yang dilewati tegangan jala-jala.

Blok diagram pengubah besaran daya ke tegangan adalah sebagai berikut:



Gambar 2.5.
Blok Diagram pengubah besaran Daya ke tegangan

⁷ Lemon.L, , Wattmeter, Elcktor Electronics, April 1991, hal 32

II.4.MULTIPLIER MC1495L

Komponen ini dirancang untuk digunakan untuk menghasilkan output yang linear dari dua input tegangan (Vx dan Vy).

MC 1495L adalah monolithic, four-quadrant multiplier yang beroperasi pada prinsip transkonduktansi⁸. Input tegangan maksimum adalah :

$$V_{X(max)} < I_{13} R_y \dots\dots\dots(16)$$

$$V_{Y(max)} < I_3 R_x \dots\dots\dots(17)$$

dimana I₁₃ dan I₃ adalah arus yang mengalir pada pin 13 dan pin 3.

Persamaan dasar dari diferensial output dari arus adalah:

$$I_A - I_B = \frac{2V_x \cdot V_y}{R_x \cdot R_y \cdot I_3} \dots\dots\dots(18)$$

dimana I_A dan I_B adalah arus yang mengalir pada pin 14 dan 2 dan V_X dan V_Y adalah input tegangan.

Tegangan output adalah sebagai berikut :

$$V_O = k \cdot V_x \cdot V_y \dots\dots\dots(19)$$

dimana $k = \frac{2 \cdot R_L}{R_x \cdot R_y \cdot I_3} \dots\dots\dots(20)$

Prosedur penyetelan untuk offset nol dan pengaturan faktor skala⁹:

⁸ Motorola, Motorola Linear/Interface Devices, Motorola Semiconductor, hal 11-42

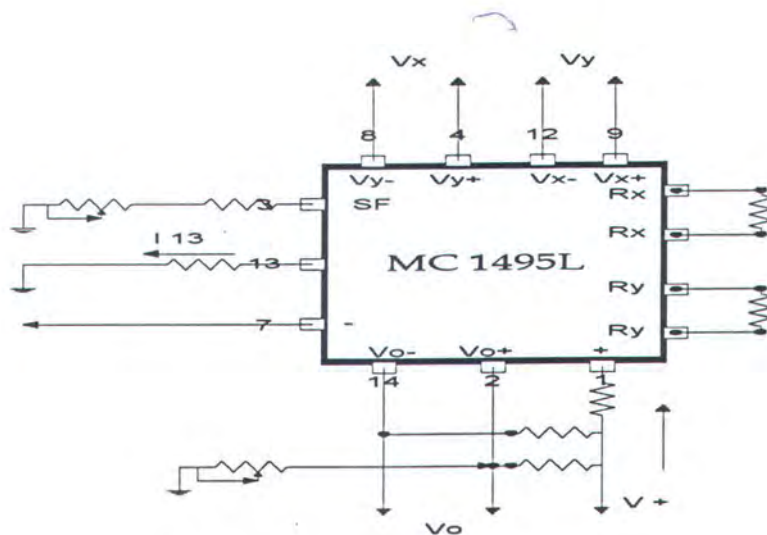
⁹ Ibid ,hal 11-46

□ Offset Input X

Input Y (pin 4) dihubungkan gelombang sinus (frekwensi 50Hz, 5 Vpp), kemudian Input X (pin 9) dihubungkan ground. Potensiometer offset X diatur sehingga diperoleh tegangan sinyal pada output sama dengan nol. ($V_o = 0$).

□ Offset Input Y.

Input X (pin 9) dihubungkan gelombang sinus (frekwensi 50Hz, 5 Vpp), kemudian Input Y (pin 9) dihubungkan ground. Potensiometer offset Y diatur sehingga diperoleh tegangan sinyal pada output sama dengan nol. ($V_o = 0$).



Gambar 2.6.
Blok Diagram MC1495L

□ Offset output.

Input X dan Y dihubungkan ke ground. Potensio offset output diatur sampai tegangan DC pada output sama dengan nol.

□ Skala Faktor.

Untuk pengaturan skala faktor , disesuaikan dengan skala faktor yang diinginkan. Tegangan DC dihubungkan ke input X dan Y, kemudian diatur potensiometer offset skala faktor sehingga diperoleh tegangan output yang disesuaikan dengan skala faktor yang dikehendaki.

II.5.PORT PARALEL INPUT OUTPUT ¹⁰

Antar muka (interface) mempunyai arti sebagai penghubung antara komputer sebagai pusat pengolah data dengan dunia luar, dimana dalam tugas akhir ini merupakan penghubung antara komputer yang bertugas mengambil data, menyimpan, mengolah dan mengirimkannya ke peralatan monitoring daya sebagai dunia luarnya.

Dibawah ini akan dijelaskan beberapa prinsip-prinsip dasar yang harus dimengerti dari perangkat input output paralel IBM PC.

II.5.1.SYNCHRONOUS DAN ASYNCHRONOUS

Pada saat akan menghubungkan suatu perangkat luar dengan sistem komputer, mungkin yang akan menjadi pertanyaan yang paling pokok adalah apakah transfer informasinya berupa asynchron atau synchron.

¹⁰ Roy W.Body, Hardware, Software And Aplication, Mc Hill. Intel Microprocessor

Pembahasan dimulai mengenai definisi dari *asynchronous* dan *synchronous* dan melihat perbedaannya.

❖ Synchronous berarti transfer data yang bisa diprediksi (clocked)

❖ *Asynchronous* berarti transfer data yang tak terprediksi (unclocked)

Dalam transfer data atau operasi input output pada *synchronous*, semua yang terjadi didalamnya mempunyai satu kesatuan sistem clock dan biasanya berada dibawah kontrol CPU. Sedangkan pada *asynchronous* semua informasi yang ditransfer tidak berada dalam satu sistem clock tetapi dapat terjadi tiap saat dengan kecepatan dan interval yang mungkin tidak beraturan.

Pemilihan transfer informasi menggunakan *synchronous* atau *asynchronous* seringkali bergantung pada jenis peripheral yang akan ditambahkan pada sistem. Sebagai contoh, jika peralatan yang akan ditambahkan itu adalah fast RAM dengan karakteristik pewaktuan yang baik, maka semua transfer yang terjadi dapat disatukan dalam satu sistem clock. Lain halnya jika kita menggunakan peralatan mekanik seperti printer dimana data-data yang akan ditransferkan mempunyai kecepatan dan interval yang tidak beraturan maka sistem *asynchronous* adalah hal yang paling tepat untuk digunakan.

II.5.2. PERLUNYA TEKNIK TRANSMISI ASYNCRHONOUS

Didalam transfer data *asynchronous* terdapat beberapa cara pengaturan aliran informasi antara sumber informasi dan tujuan informasi. Teknik yang paling banyak digunakan adalah dikenal dengan teknik *handshaking* yang berisikan

kumpulan sinyal-sinyal penyinkron yang terpisah yang digunakan untuk menentukan letak dari transfer data. Sinyal-sinyal *handshaking* ini dapat sepenuhnya terpisah dari data dan menggunakan jalur sinyal yang terpisah, atau dapat diselipkan secara langsung dalam aliran data.

II.5.3.HANDSHAKING

Contoh pemakaian teknik *handshaking* yang mudah kita temui adalah pada peralatan pencetak (printer) dimana terdapat sinyal-sinyal pengatur aliran data dari komputer ke printer. Sebagai alasan utama penggunaan teknik ini adalah perbedaan kecepatan yang sangat mencolok antara kecepatan komputer sebagai pengirim data dengan printer yang akan mencetaknya kedalam karakter-karakter yang sesuai.

Pada port paralel I/O terdapat tiga sinyal standart dari *handshaking* yaitu sinyal ACKNLG, BUSY dan STROBE. Ketiga sinyal ini berfungsi sebagai sinyal-sinyal pengontrol aliran data yang menjembatani perbedaan kecepatan dari printer dan komputer. Cara kerja ketiga sinyal ini adalah sebagai berikut:

- ❖ Jika semua data yang terdapat didalam buffer printer telah tercetak, maka printer akan mengirimkan sinyal ACKNLG untuk memberitahu komputer bahwa printer siap untuk menerima data berikutnya.
- ❖ Komputer yang menerima sinyal ACKNLG ini segera akan mengirimkan data dengan memberikan pula pulsa STROBE. Pulsa STROBE yang diterima printer secara langsung menyebabkan printer mengirimkan sinyal BUSY sebagai pertanda bahwa data telah diterima dan buffer telah penuh.

- ❖ Kejadian ini berulang terus menerus hingga semua data dari komputer telah tercetak oleh printer.

II.5.4. PROGRAMMED I/O DAN INTERRUPT I/O

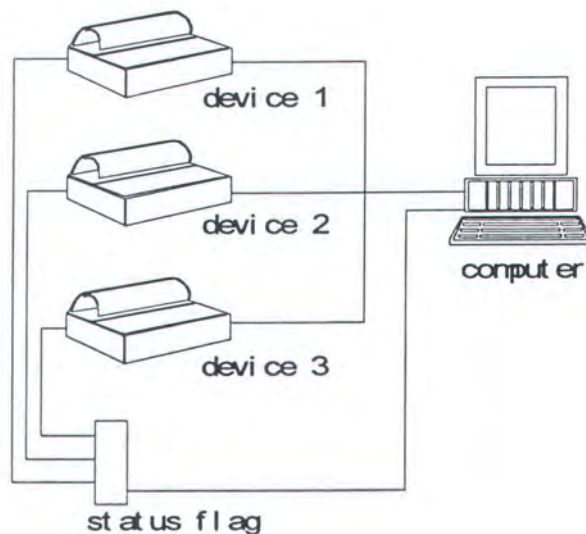
Suatu sinyal *handshaking* ke CPU, seperti sinyal BUSY pada printer dapat diterima dengan dua cara. Cara pertama yaitu dengan menggunakan teknik *programmed I/O* atau dikenal pula sebagai teknik polling dimana tanggapan dari sinyal *handshaking* ini terdapat pada program dibawah kontrol program utama. Biasanya komputer diletakkan pada kondisi pengerjaan yang berulang-ulang menunggu sinyal *handshaking* (*wait loop*). Jika sinyal yang ditunggu-tunggu ini terdeteksi oleh komputer maka komputer akan keluar dari *looping* dan segera mengerjakan program sebagai tanggapan dari adanya sinyal *handshaking* tadi.

Jika sinyal *handshaking* dari printer dapat terjadi kapan saja dengan interval yang tidak beraturan maka teknik *handshaking* ini termasuk dalam teknik *asynchronous*. Penggunaan teknik *programmed I/O* ini dapat meminimalkan pemakaian hardware.

Pemakaian lebih dari satu perangkat luar yang akan dihubungkan dengan komputer, mengharuskan program untuk peralatan tersebut dalam arti yang mana yang harus dilayani dengan mencari suatu sinyal permintaan. Sebagai contoh, pada gambar 2.7 terdapat tiga peralatan yang dihubungkan dengan komputer. Perangkat tersebut dipilih dengan mengirimkan satus *flag* dan memeriksa masing-masing bit secara berurutan. Jika ditemukan sebuah bit aktif maka bit ini akan digunakan

sebagai penunjuk alamat dari suatu service routine. Peralatan disusun berdasarkan tingkat prioritas pilihan.

Teknik programmed I/O atau polling ini memang mudah diimplementasikan tetapi mempunyai beberapa kekurangan. Yang pertama teknik ini mengharuskan komputer untuk memeriksa terus menerus kondisi status flag sehingga menyita semua waktu komputer, semua perangkat harus diperiksa walaupun pada dasarnya sedang tidak membutuhkan pelayanan. Kedua, waktu respon yang lambat karena komputer melakukan pemeriksaan secara berurutan dan bagi perangkat lunaknya sendiri harus memberikan alamat routine pelayanan yang sesuai begitu permintaan ditemukan.



Gambar 2.7
Programmed I/O

Interrupt I/O merupakan cara lain untuk menerima sinyal *handshaking* selain dari programmed I/O yang dijelaskan diatas. Teknik ini mengatasi kelemahan dari teknik programmed I/O dimana didalam interrupt I/O sinyal *handshaking*

berubah menjadi sinyal interrupt request dari komputer. Saat interrupt terjadi, pengerjaan komputer langsung mengarah pada interrupt service routine yang memberikan program pelayanan pada perintah yang mengirimkan interupsi tadi. Dibawah kontrol interrupt I/O ini, komputer melayani perangkat luar hanya pada saat dibutuhkan saja, diluar itu komputer dapat mengerjakan pekerjaan yang lain. Pada umumnya penggunaan interrupt I/O meningkatkan respon waktu.

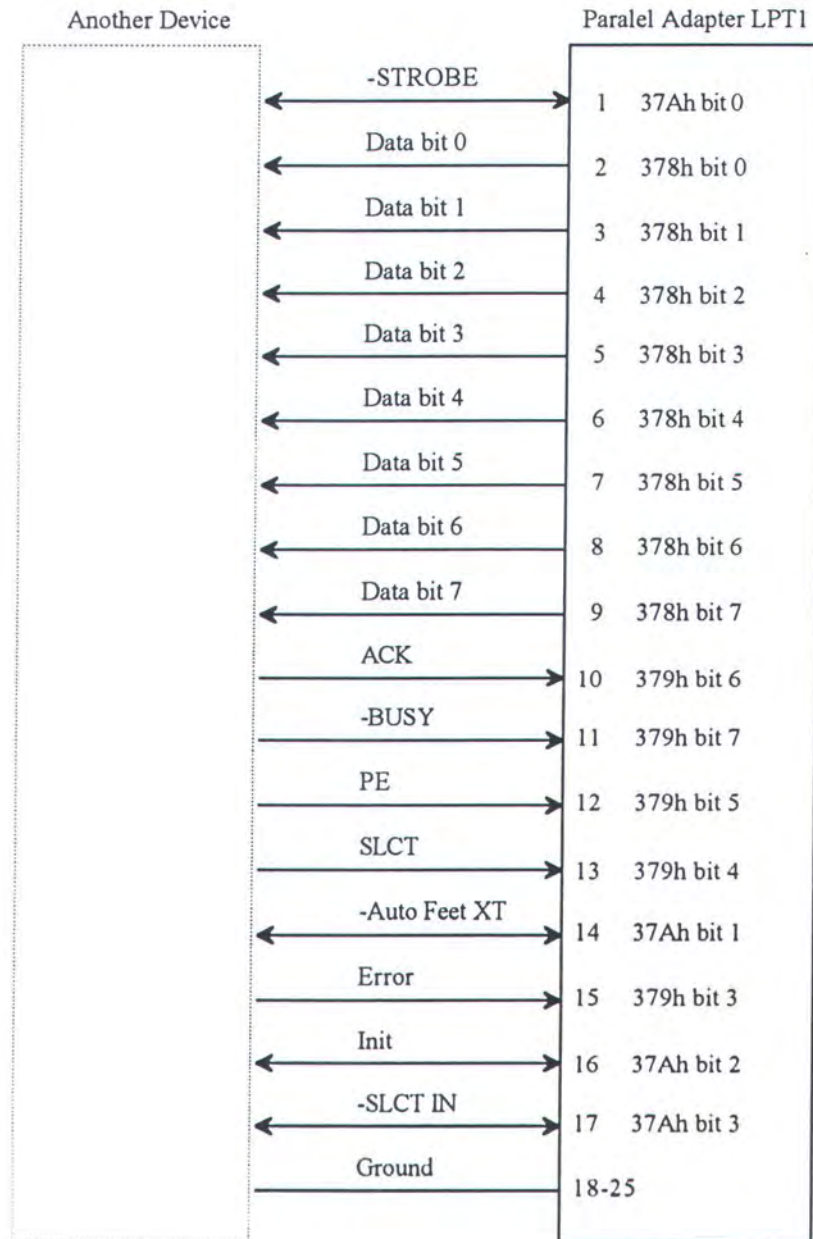
II.5.5.PORT PARALEL IBM PC (LPT)

Setiap IBM PC memiliki standart port paralel yang biasanya digunakan untuk pemakaian printer.

Secara fisik, paralel port berupa konektor 25 pin tipe D yang terletak pada adapter paralel port (biasanya terpisah dengan motherboard). Pada gambar 2.8 memperlihatkan printer port sebagai port paralel berisikan register data latch, printer status dan printer control. Ketiga register ini diakses dengan mudah menggunakan instruksi IN dan OUT. Alamat register-register ini diperlihatkan pada tabel dibawah:

Tabel 2.3.
Alamat port LPT

REGISTER	ALAMAT
Printer control port 1, 2, 3	3BEH, 37AH, 27AH
Printer status port 1, 2, 3	3BDH, 379H, 279H
Data latch port 1, 2, 3	3BCH, 378H, 278H



Gambar 2.8
Fungsi Pin-pin paralel port (LPT)

Fungsi Pin-pin Adapter Paralel adalah sebagai berikut :

Ada 3 address yang digunakan oleh paralel adapter pada LPT1 yakni address 378h, 379h dan 37Ah sedang LPT2 menggunakan address 3BCh, 3BDh dan 3BEh

Spesifikasi setiap address pada LPT1 sebagai berikut :

- ♦ Address 378h hanya dapat digunakan sebagai output, sedangkan jumlah bit yang dapat digunakan sebanyak 8 bit mulai pin 2 sebagai bit 0 sampai pin 9 sebagai bit 7 pada Paralel Adapter.
- ♦ Address 379h hanya dapat digunakan untuk input data, tetapi jumlah data yang dapat diinputkan hanya 5 bit, yaitu :

bit 3 pada pin 15

bit 4 pada pin 13

bit 5 pada pin 12

bit 6 pada pin 10

bit 7 pada pin 11

khusus untuk bit 7 data yang diterima merupakan komplemen dari data yang diinputkan pada pin 11, misal inputnya 0 maka data yang diterima 1, sedangkan jika input 1 maka data yang diterima adalah 0.

Terkhusus juga untuk bit 6 terhubung ke interrupt hardware pada PC yakni IRQ 5 dengan no interrupt INT 0Ch untuk LPT1 dan IRQ 6 untuk LPT2 dengan no interrupt INT 0Dh, dalam kondisi normal interrupt ini passif sedang untuk mengaktifkannya yakni dengan mengirim data ke address 37Ah

dengan data 0001000b atau bit ke 4 dikirim data 1, sedang interrupt akan terjadi jika ada pulsa low ke adres 379h bit 6.

- ♦ Adres 37Ah sebenarnya difungsikan hanya sebagai output dengan jumlah data sebanyak 4 bit yakni :

bit 0 pada pin 1

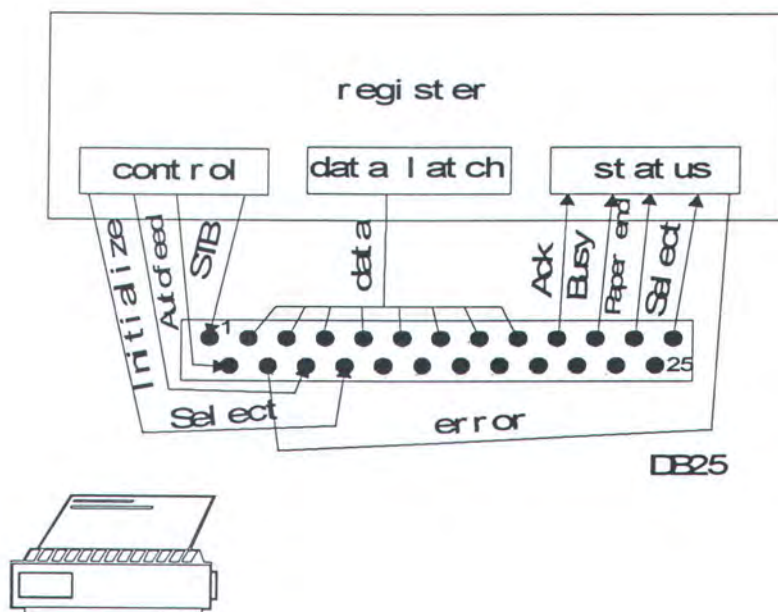
bit 1 pada pin 14

bit 2 pada pin 16

bit 3 pada pin 17

Tetapi setelah mengamati hardware yang ada pada paralel adapter sebenarnya adres ini dapat digunakan sebagai input, hanya saja sebelum memanfaatkannya sebagai input terlebih dahulu harus diinisialisasi dengan data 04h, hal ini dimaksudkan agar data yang diambil benar karena pada rangkaian outputnya pada paralel adapter digunakan rangkaian transistor yang bersifat "open collector" sehingga transistor harus dimatikan terlebih dahulu dengan memberikan data 04h pada adres ini.

Sebagaimana pada adres 379h ada beberapa bit yang bersifat komplemen yakni bit 0, bit 1 dan bit 3.



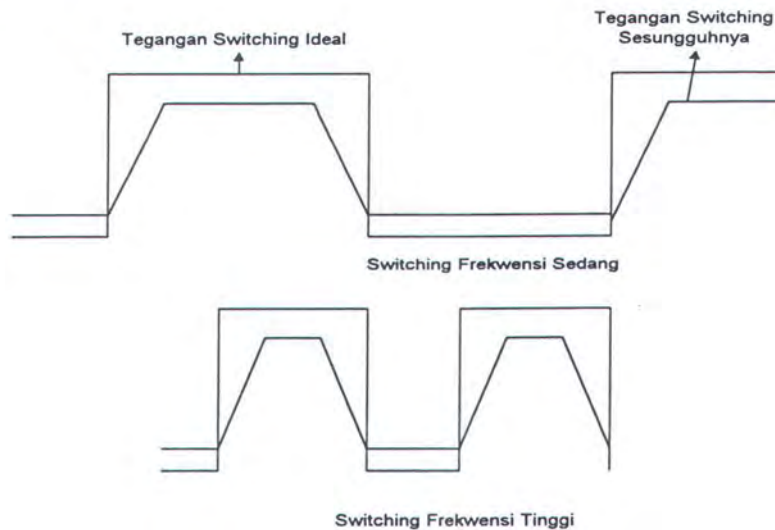
Gambar 2.9
Paralel port DB25

II.6. TRANSISTOR SWITCHING

Banyak sekali jenis-jenis transistor switching, mulai yang terbuat dari germanium-germanium modern sampai darlington monolitik yang canggih. Bahkan sering ditemukan bahwa lebih dari satu jenis transistor dapat dipakai untuk fungsi yang sama.

Pada gambar 2.10 terlihat sepotong segmen gelombang switching ideal dan juga terlihat bentuk gelombang sesungguhnya yang dihasilkan oleh sebuah transistor switching. Bentuk gelombang sempurna segiempat selalu ingin dicapai namun tidak pernah berhasil. Kehebatan gelombang switching ideal adalah bahwa kehilangan energinya selalu nol. Bentuk gelombang switching ideal ini selalu dalam

keadaan yang benar-benar on atau off dan perubahan dari dua keadaan ini tidak memerlukan waktu.



Gambar 2.10
Bentuk gelombang transistor switching

Bila gelombang dalam keadaan on, ia akan mengalirkan arus secara penuh tanpa adanya penurunan tegangan diantara kedua terminalnya. Bila keadaan off, akan timbul tegangan di kedua terminalnya tanpa ada arus bocor yang mengalir. Gelombang trapesium merupakan bentuk gelombang yang sesungguhnya adalah gelombang switching yang harus kita terima dari transistor switching.

Bentuk gelombang switching yang sesungguhnya memiliki empat daerah kehilangan daya. Bila switch dalam keadaan off, ia akan membocorkan sedikit arus. Karena tegangan emiter-kolektor adalah paling tinggi saat off, walaupun sedikit kehilangan ini tetap penting. Ketika switch dalam keadaan on, sesungguhnya ia juga tidak benar-benar on, tegangan jenuh kolektor-emiter, $V_{ce(sat)}$, tertahan antara

terminal kolektor-emiter. Walaupun tegangan tersebut relatif rendah, terjadinya justru pada saat aliran arus kolektor paling besar, dengan demikian kehilangan ini sangat terasa.

Daerah terarsir dari gambar tersebut menunjukkan penyimpangan keadaan ideal yang lain lagi. Perpindahan keadaan (transisi) on dan off memerlukan waktu. Selain dalam keadaan transisi operasi switching ini berubah menjadi elemen perangkat linier, yaitu penguat kelas A. hal ini akan membuang-buang daya karena cukup banyak tegangan dan arus yang melaluinya. Untungnya masa transisi ini sebentar sekali. Tetapi bila frekuensi switching yang dipakai terlalu tinggi, kehilangan daya pada saat transisi menjadi dominan.



BAB III

PERENCANAAN ALAT

BAB III

PERENCANAAN ALAT

III.1.UMUM

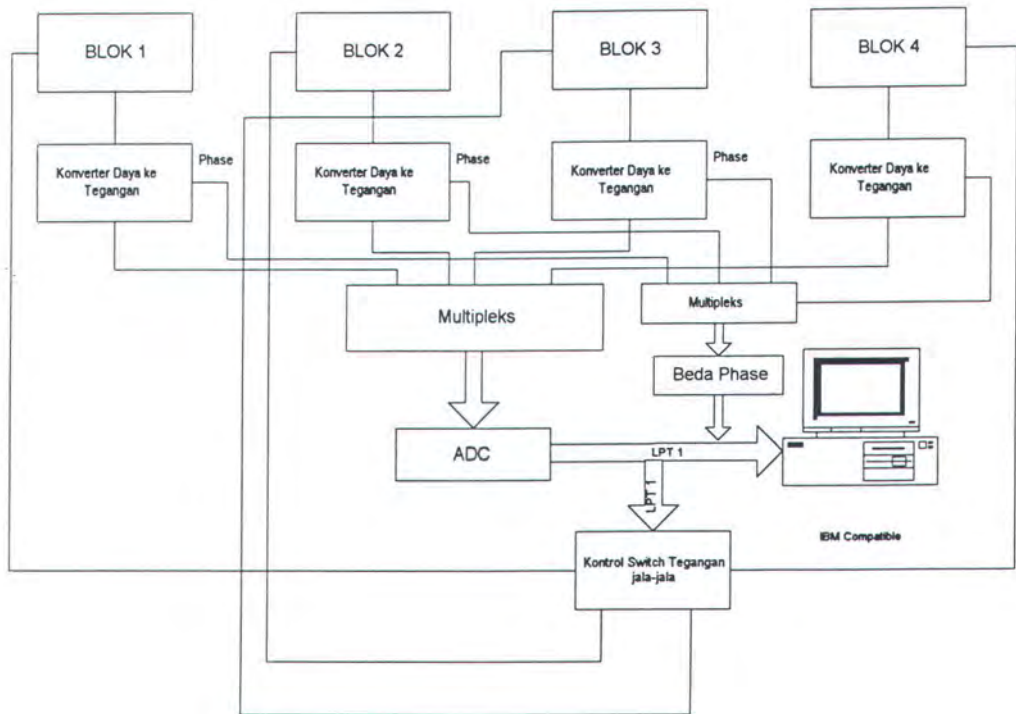
Secara umum pada perencanaan alat dibagi menjadi dua bagian utama , yaitu pembahasan perangkat keras dan perangkat lunak.

Perencanaan perangkat keras meliputi perencanaan rangkaian pengubah daya ke tegangan, rangkaian beda phase, rangkaian multipleks analog , rangkaian pengubah sinyal analog ke sinyal digital (ADC) dan rangkaian switch tegangan jala-jala.

Perencanaan perangkat lunak meliputi perencanaan program yang mengatur kerja dari sistem rangkaian secara keseluruhan. Perangkat lunak ini mengatur proses kerja multipleks analog, pengambilan data ADC, pengolahan data dan kontrol rangkaian switch tegangan jala-jala, serta beda phase.

III.2.PERENCANAAN PERANGKAT KERAS.

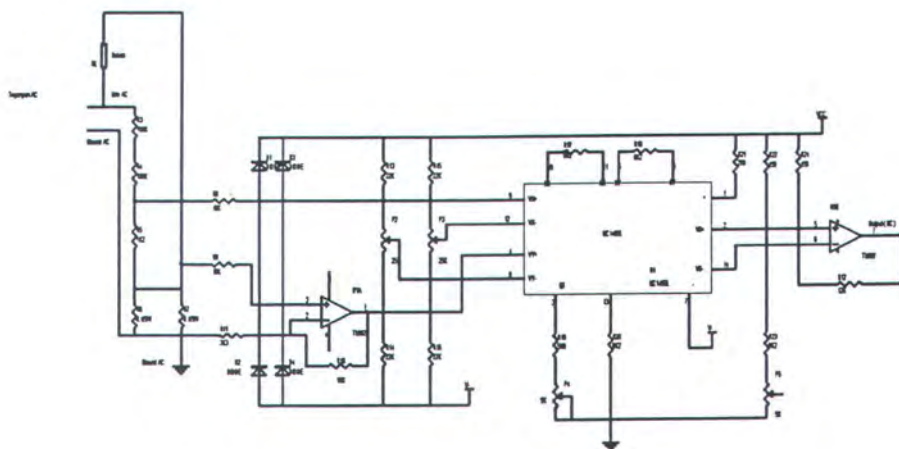
Perencanaan perangkat keras dapat dilihat secara keseluruhan pada blok diagram.



Gambar 3.1
Blok Diagram Rangkaian

III.2.1. PENGUBAH DAYA KE TEGANGAN.

Rangkaian pengubah daya ke tegangan adalah sebagai berikut: Beban RL merupakan beban yang akan diukur besar dayanya . Tahanan R_6 (0.1 ohm/5W) dan R_7 (0.1 ohm/5W) dihubungkan secara paralel, yang mengalir arus pada beban. Tahanan efektif dari dua tahanan yang terhubung secara paralel itu adalah 0.05 ohm/ 10W. Kedua tahanan itu dipakai menjadikan besarnya arus yang mengalir pada beban menjadi besaran tegangan yang sesuai dengan perubahan arus pada beban , yang selanjutnya dikuatkan oleh op-amp (TL082) sebelum masuk keinput dari *four-quadrant multiplier*. Op-Amp TL082 digunakan sebagai Non Inverting Amplifier.



Gambar 3.2.
Pengubah Daya ke Tegangan

Besarnya penguatan adalah sebagai berikut :

$$A_v = \frac{R_f}{R_i} + 1 = \frac{R_{10}}{R_{11}} + 1 \dots\dots\dots (1)$$

$$A_v = \frac{18k}{3.3k} + 1 = 6.45 \dots\dots\dots (2)$$

R₃ - R₄ dan R₅ dihubungkan secara paralel dengan beban (R_L) merupakan pembagi tegangan. Tahanan R₈ dihubungkan ke keluaran pembagi tegangan sebelum masuk ke V_x input dari multiplier. Dua tahanan yang dihubungkan seri digunakan pada cabang di atasnya dari pembagi tegangan untuk tetap mengalir dengan baik tegangan maksimum yang bisa dihubungkan pada sebuah tahanan 0.125 W. Tegangan line yang diigunakan sekitar 220V, ini menjadi aman dengan menggunakan dua resistor yang dihubungkan seri dengan tegangan jala-jala yang bisa naik sampai 250 Volt.

Diode D1 sampai D4 melindungi input Op-amp dan input multiplier oleh kutup yang terbalik dari tegangan supply.

Operasi dasar dari multiplier MC 1495L adalah

$$V_o = k V_x V_y \dots\dots\dots(3)$$

$$\text{dimana } k = \frac{2R_L}{R_X R_Y I_3} \dots\dots\dots(4)$$

Dalam rangkaian , R_L adalah dua tahanan 150 ohm yaitu R_{22} dan R_{24} pada output dari multiplier , dimana R_X dan R_Y dibentuk oleh tahanan yang dihubungkan pada pin 10 - 11 dan pin 5 - 6 dari IC MC 1495L. Arus I_3 mengalir dari pin 3 ke ground. Dan dapat diatur besarnya faktor skala (k) sesuai besarnya Cosinus beda fase. Tahanan variabel P_4 dan P_5 setiap supply mengoffset kompensasi tegangan pada input $V_x -$ dan $V_x +$. Tegangan ini untuk mengeset diferensial tegangan pada input relevan multiplier.

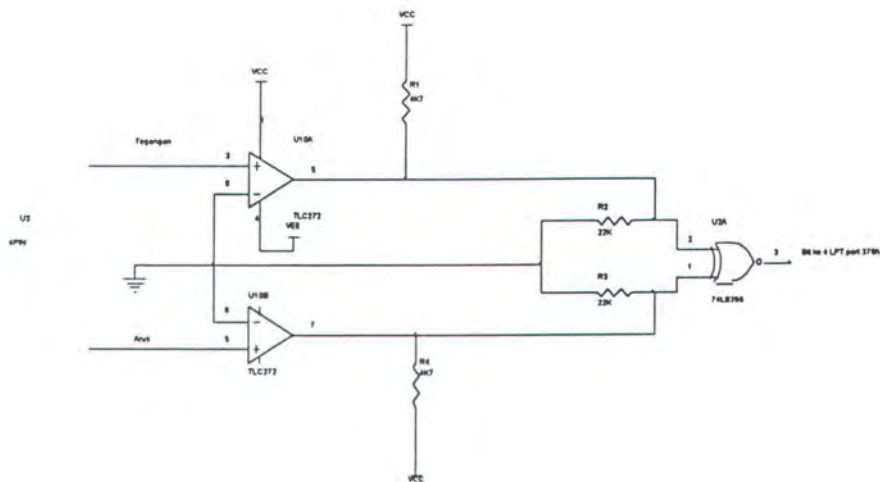
Op-amp kedua dari rangkaian, yaitu pada output multiplier, menguatkan sinyal output pada multiplier.

Sejak rangkaian dihubungkan langsung dengan beban yang akan diukur besarnya daya, meminta banyak perawatan dan perhatian untuk mencegah resiko dari electrical shock. Untuk mencegah itu maka dibutuhkan kawat yang besar untuk menghubungkan dengan tegangan jala-jala..

III.2.2.BEDA PHASE

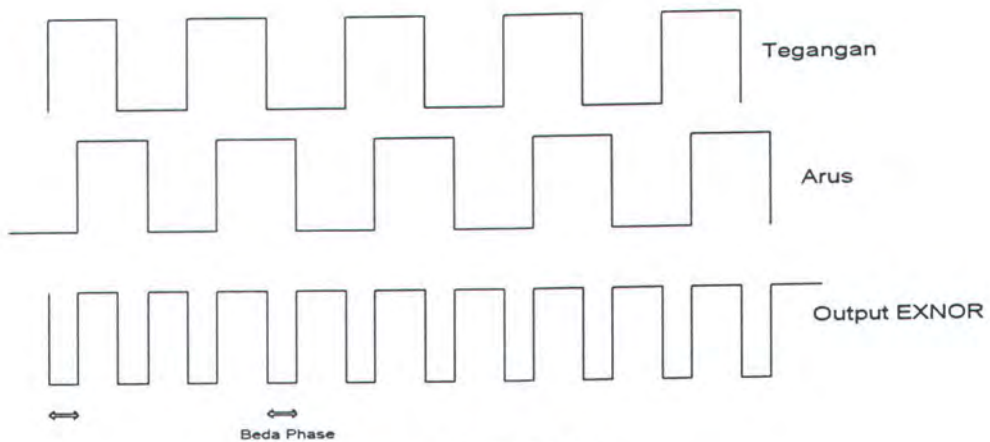
Rangkaian ini terdiri dari dua zero crossing detektor dengan op-amp IC TLC272. Input dari Zero crossing detector adalah dari rangkaian penerima arus

dan tegangan pada beban yang dialiri tegangan bolak-balik . Output dari IC TLC272 berupa sinyal digital, bila input positif maka sinyal akan high (+5V) dan bila input negatif maka sinyal akan low (0 V). Sinyal tersebut diterima oleh gerbang EXNOR (74LS266) yang akan low jika input mempunyai polaritas yang berbeda, dan akan high bila input mempunyai polaritas yang sama. Lebar sinyal low dari EXNOR tersebut menyatakan besarnya beda phase antara arus dan tegangan bolak-balik. Selanjutnya output dari EXNOR tersebut masuk ke paralell port untuk kemudian diproses dengan perangkat lunak.



Gambar 3.3
Rangkaian beda phase

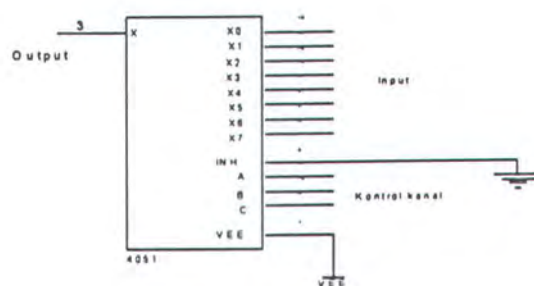
Gambar 3.3.
Rangkaian Deteksi Beda Phase



Gambar 3.4.
Pulsa pada rangkaian beda phase

II.2.3.RANGKAIAN MULTIPLEKSER ANALOG.

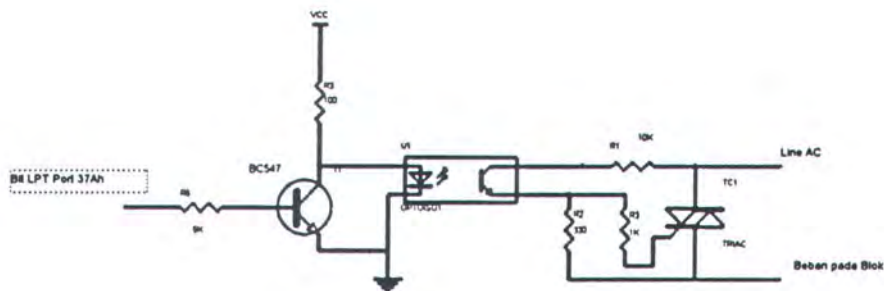
Rangkaian multiplekser analog seperti terlihat pada gambar 3.5 IC yang digunakan untuk multiplekser analog yaitu IC CMOS 4051. IC 4051 memiliki delapan jalur input multiplekser dengan keluaran sesuai dengan pemilih sinyal input. Rangkaian multiplekser ini digunakan untuk melihat beda phase setiap blok, sedang daya pada setiap blok menggunakan multiplekser internal dalam ADC MAX180ACPL.



Gambar 3.5
Multiplekser analog

III.2.4.RANGKAIAN SWITCH TEGANGAN JALA-JALA.

Rangkaian untuk menghubungkan dan memutuskan tegangan jala-jala yang menuju ke beban menggunakan Triac. Kondisi Triac ditentukan oleh MOC 3040, apabila MOC 3040 dialiri arus mengakibatkan kondisi triac keadaan ON, dan sebaliknya bila tidak dialiri arus listrik maka kondisi triac OFF. Keadaan tersebut dikontrol bit-bit pada port 37Ah (untuk LPT1). Apabila bit pada port 37Ah (untuk LPT1) '1' maka transistor switching dalam keadaan On, sehingga arus akan mengalir ke ground dan tidak ada arus yang mengalir pada MOC 3040, yang selanjutnya mengakibatkan Triac keadaan Off dan tegangan jala-jala tidak akan terhubung ke beban, demikian sebaliknya bila bit pada port 37Ah (untuk LPT1) '0' maka transistor switching dalam keadaan Off, sehingga ada arus yang mengalir pada MOC 3040, yang selanjutnya mengakibatkan Triac keadaan On dan tegangan jala-jala akan terhubung ke beban tegangan. Untuk pengendali triac selengkapnya adalah sebagai berikut (untuk LPT1).



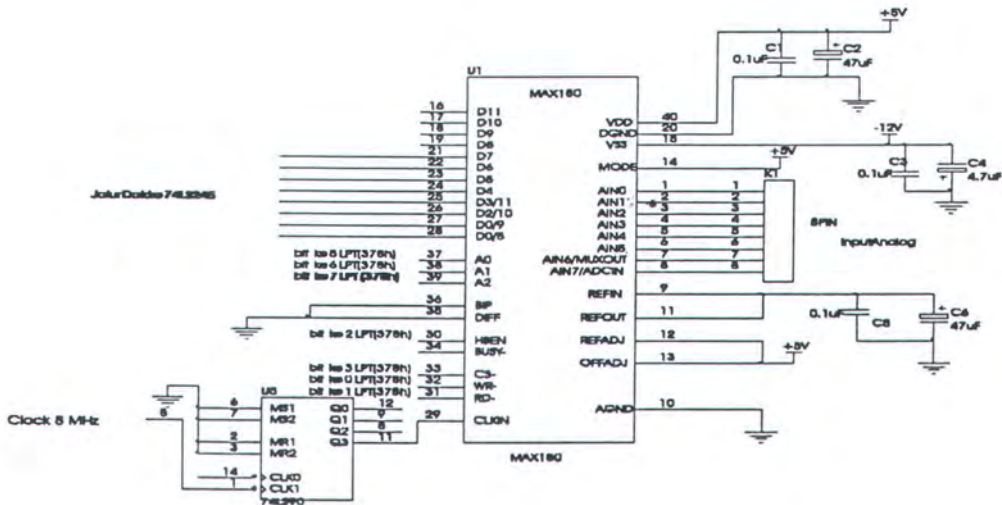
Gambar 3.6
Rangkaian Switch Tegangan Jala-Jala

Tabel 3.1.
Bit Kontrol Switch Tegangan Jala-jala.

Port 37Ah Bit ke	Beban Blok ke
0	1
1	2
2	3
3	4

III.2.5. RANGKAIAN PENGUBAH ANALOG KE DIGITAL

ADC yang dipakai adalah Succesive Approximation Register (SAR) type MAX180ACPL 12 bit. Rangkaian seluruhnya dikendalikan oleh LPT.



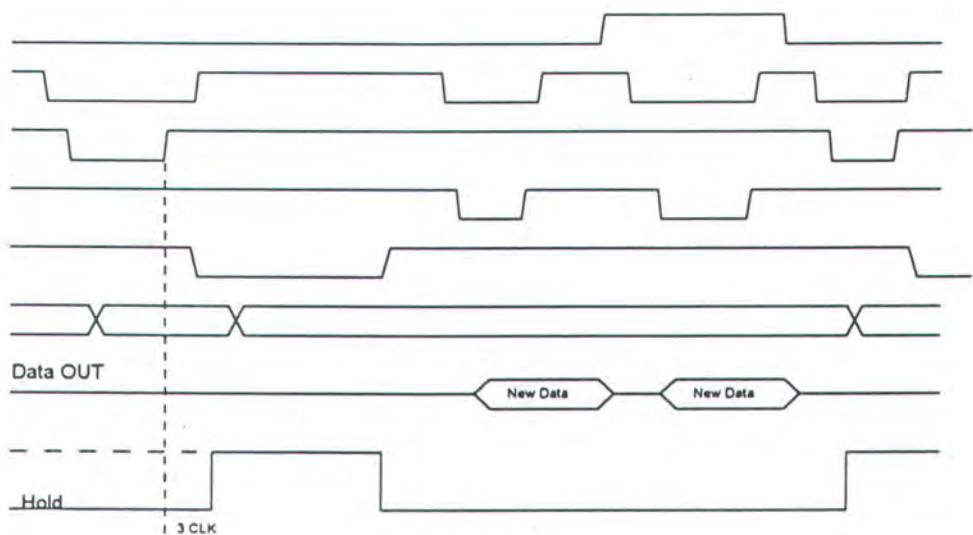
Gambar 3.7.
Rangkaian ADC

Pada rangkaian ADC MAX180ACPL ini digunakan clock dengan frekwensi 1.6 MHz. Clock dengan frekwensi sebesar itu diperoleh dari rangkaian k dengan menggunakan kristal frekwensi 8 MHz. Dengan rangkaian pembagi

frekwensi yang menggunakan IC 74LS90 , maka frekwensi 8 MHz dibagi 5 sehingga menjadi 1.6 MHz.

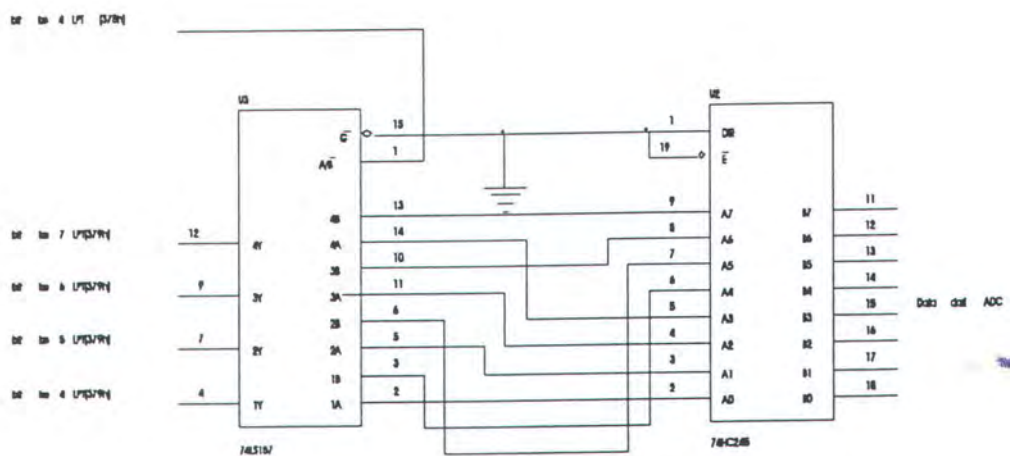
Resolusi dari ADC MAX180ACPL adalah $V_{DD} / 4096$. V_{DD} pada rangkaian ini adalah 5 volt sehingga resolusinya $5 \text{ Volt} / 4096 = 1.22 \text{ mVolt}$ perbit. Pada rangkaian tersebut menggunakan mode 1 (Input/output port mode timing two byte read). Timing diagram dari mode tersebut pada gambar 3.8.

Pin 14 (Mode) dari MAX180ACPL dihubungkan ke V_{DD} sehingga merupakan mode 1. Pin 35 dan 36 (BIP dan DIFF) dihubungkan ke ground yang berarti Unipolar dan non differnsial input. Tegangan referensi menggunakan internal referensi dari ADC MAX180ACPL dengan menghubungkan REFFIN dan REFOUT. REFADJ dan OFFADJ tidak digunakan, pada rangkaian keduanya dihubungkan dengan V_{DD} . Pin CS, WR, RD, dan HBEN semua dikontrol oleh LPT (pada LPT1 port 378h). Dekoder alamat untuk multipleks input (A0,A1,A2) juga dikendalikan oleh LPT .



Gambar 3.8
Timing Diagram ADC MAX180ACPL Mode 1

Pada ADC ini pembacaan data 12 bit dari ADC dilakukan dengan dua kali pembacaan, yang semua itu dikontrol oleh HBEN. Bila HBEN = 0, maka data keluar dari ADC 8 bit LSB (D0-D7). Bila HBEN = 1, maka data keluar dari ADC 4 bit MSB (D8-D11).



Gambar 3.9
Jalur komunikasi data dari ADC ke LPT

Data keluaran ADC masuk ke buffer 8 bit IC 74LS245. Pembacaan data oleh LPT (pada LPT1 port 379h), dilakukan dua kali karena memang port ini mempunyai 4 bit input. Jadi data 8 bit LSB keluaran ADC dibaca dua kali setiap 4 bit, dengan menggunakan IC 74LS157 yang merupakan dekoder 8 to 4. Pengontrolan IC 74LS157 dilakukan oleh port LPT1 378h. Data keluaran ADC sebanyak 12 bit oleh LPT dibaca sebanyak 3 kali dengan memanfaatkan IC 74LS157.

III.2.5.HUBUNGAN KE PARALEL PORT (LPT)

Paralel port untuk LPT 1 terdiri dari 3 alamat yaitu 378h, 379h, 37Ah.

Ketiga alamat itu digunakan sebagai berikut :

Alamat **378h** dipakai sebagai output adalah :

Tabel 3.2.
Pemakaian bit LPT1 port 378h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
A		D		C			
A2	A1	A0	A/B	CS	HBEN	RD	WR

Alamat **379h** dipakai sebagai input adalah:

Tabel 3.3.
Pemakaian bit LPT1 port 379h

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Data dari ADC							
D7	D6	D5	D4	Phase	1	1	1

Alamat dipakai **37Ah** sebagai output adalah:

Tabel 3.4.
Pemakaian bit LPT1 port 37Ah

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
					Switch	Tegangan	Jala-Jala
1	1	1	1	SW4	SW3	SW2	SW1

untuk LPT2 , port (378h) = port (3BCh)

port (379h) = port (3BDh)

port (37Ah)= port (3BEh).

II.3.PERENCANAAN PERANGKAT LUNAK

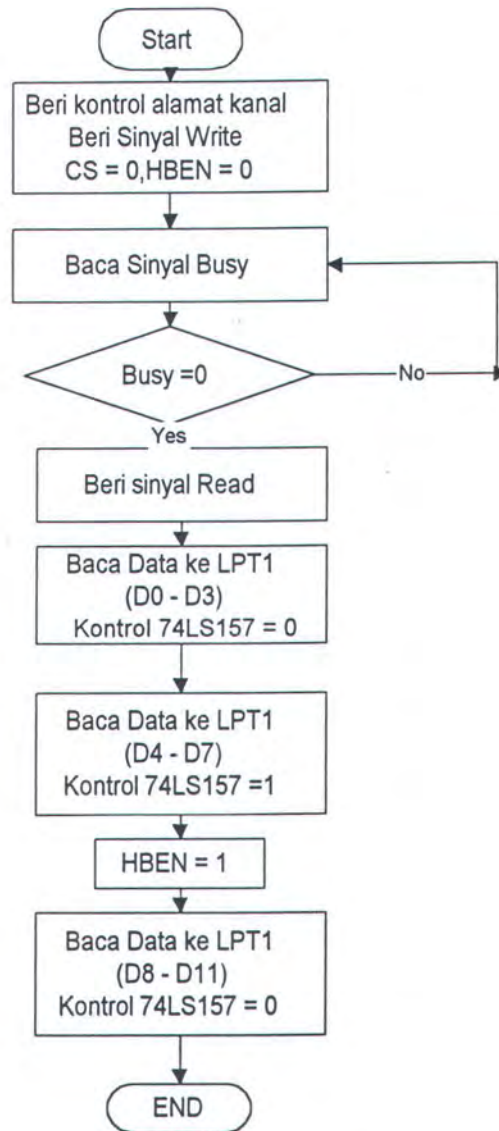
Perangkat lunak yang digunakan untuk mengatur proses kerja dengan menggunakan Borland Delphi. Secara garis besarnya perangkat lunak ini meliputi :

- Program kerja ADC dan pengambilan data.
- Program pengolahan data
- Program kontrol switch tegangan jala-jala.
- Program pembatasan daya setiap blok.
- Program Beda Phase

III.3.1.PROGRAM KERJA ADC DAN PENGAMBILAN DATA.

Pada gambar memperlihatkan flowchart dari kerja ADC untuk pengambilan data besarnya daya yang mengalir pada setiap blok.

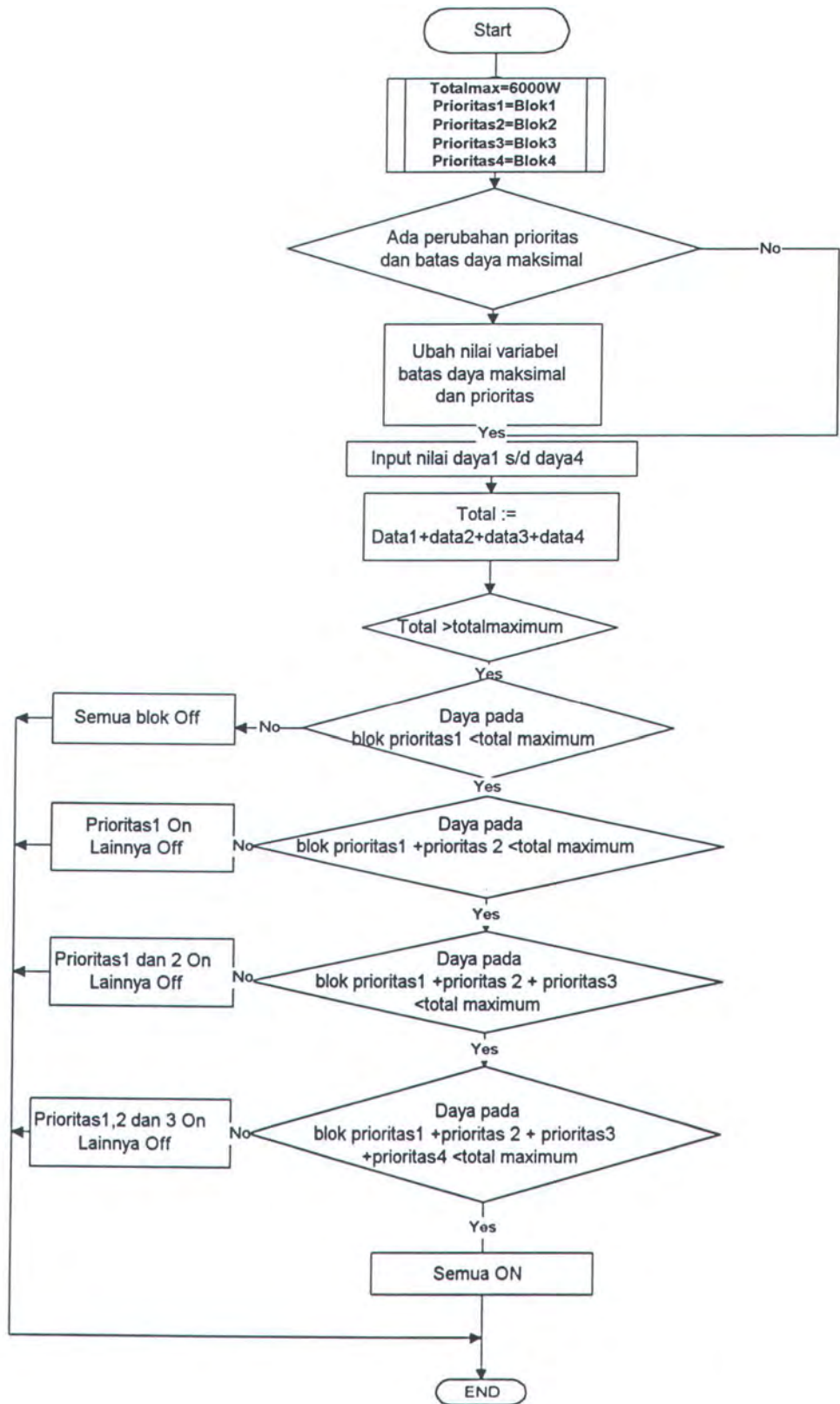
Data setiap blok diambil oleh ADC dengan memanfaatkan multiplekser analog internal ADC. Pengaturan data dari blok mana yang sedang diambil ditentukan oleh alamat port 378h. Data keluaran dari ADC disimpan oleh PC dalam variabel yang berbeda. Data 12 bit keluaran dari ADC akan dibaca oleh LPT sebanyak 3 langkah yaitu pembacaan D0 - D3, D4 - D7 dan terakhir D8 - D11.



Gambar 3.10
Flowchart pengambilan data ADC

III.3.2. PROGRAM PENGOLAHAN DATA

Flowchart dari program proses pengolahan data yang diterima dari ADC adalah seperti pada gambar 3. 11 .



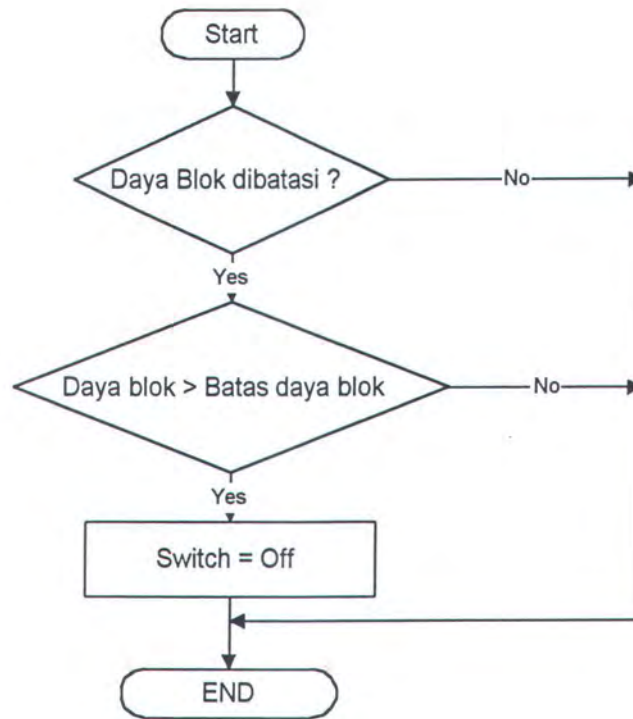
Gambar 3.11
Flowchart proses pengolahan data

Pada awal program dijalankan akan setiap variabel seperti nilai prioritas dan nilai daya maksimum diberi nilai awal. Kemudian bila ada perubahan nilai-nilai tersebut diatas, maka akan dibandingkan daya yang mengalir lebih besar dari batas daya maksimum. Bila ternyata lebih besar maka akan dilihat daya saat itu yang mengalir pada blok prioritas ke 1. Bila masih dibawah batas daya maksimum maka blok prioritas ke 1 dinyalakan. Kemudian dilihat besar daya blok prioritas ke 1 ditambah daya blok prioritas ke 2. Bila masih dibawah batas daya maksimum maka kedua blok itu dinyalakan. Seterusnya dilihat besar daya blok prioritas ke 1 ditambah daya blok prioritas ke 2 ditambah daya prioritas ke 3, bila masih dibawah batas maksimum maka ketiga prioritas dinyalakan .

III.3.3. PROGRAM PEMBATAHAN DAYA TIAP BLOK

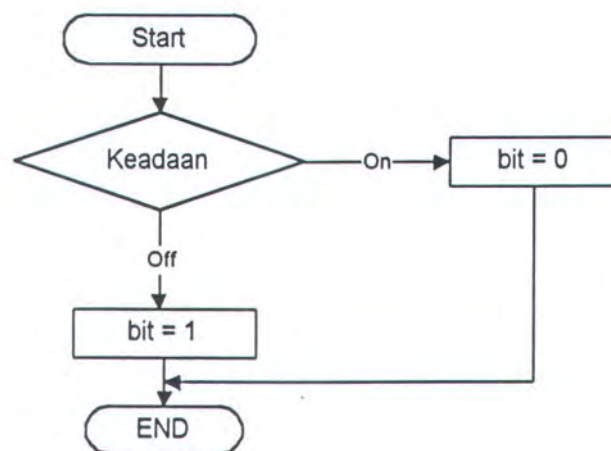
Gambar flowchart dari pembatasan daya setiap blok adalah pada gambar 3.12 . Dalam sistem, setiap blok dapat dibatasi atau tidak dibatasi dayanya. Jadi setiap blok nilai batas dayanya dapat variabel.

Bila ternyata daya yang mengalir lebih besar dari batas daya blok yang dimasukan maka blok tersebut akan dimatikan dengan memberikan output high bit kontrol switch.



Gambar 3.12
Flowchart Pembatasan Daya tiap blok

III.3.4. PROGRAM KONTROL SWITCH TEGANGAN BOLAK-BALIK



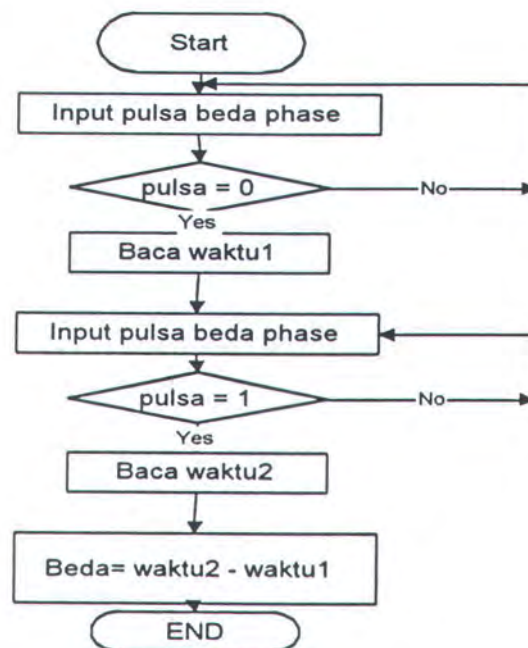
Gambar 3. 13
Kontrol Switch Tegangan Jala-jala

Flowchart dari program kontrol switch tegangan jala-jala seperti pada gambar 3.13 Pada gambar terlihat, bila diinginkan blok dimatikan dengan memberi sinyal high pada output bit kontrol blok tersebut. Bila diinginkan blok dinyalakan dengan memberi sinyal low pada output bit kontrol blok tersebut

III.3.5.PROGRAM BEDA PHASE

Pada gambar 3.14 merupakan gambar flowchart dari perhitungan beda phase. Apabila polaritas tegangan berbeda maka keluaran XNOR akan transisi dari tinggi ke rendah, pada saat itu dibaca waktu di komputer. Pada transisi dari rendah ke tinggi , akan dibaca juga waktunya di komputer. Selisih kedua waktu itu merupakan lebar beda phase. Perhitungan beda phase adalah sebagai berikut :

$$\text{Cos}\Phi = \text{Cos}\left(\frac{\text{bedawaktu}}{10} \times 180\right) \dots\dots\dots(5)$$



Gambar 3.14
Flowchart Beda Phase



BAB IV

PENGUJIAN DAN PENGUKURAN

BAB IV

PENGUJIAN DAN PENGUKURAN

Dalam suatu sistem rangkaian perlu dilakukan kalibrasi, pengujian dan pengukuran terhadap beberapa unit rangkaian. Sedangkan bagian rangkaian yang perlu dilakukan pengujian adalah bagian pengubah besaran daya ke tegangan. Sedangkan pengujian meliputi rangkaian pengubah besaran daya ke tegangan dan, rangkaian ADC. Untuk pengukuran dilakukan terhadap beban dengan hasilnya terlihat pada PC.

IV.1. KALIBRASI PENGUBAH DAYA KE TEGANGAN

Pada kalibrasi bagian ini diperlukan Multimeter dan sinyal generator gelombang sinus.

- ❑ Pada awal dilakukan pengaturan variabel resistor pada LM317 dan LM337 agar power supply mencapai +7.5 Volt dan - 7.5 Volt.
- ❑ Selanjutnya output sinus sinyal generator diatur pada tegangan $V_{p-p} = 3V$ dan frekwensi 50 Hz, serta offset DC = 0 Volt. Output sinyal generator ini dihubungkan pada pin3 dari IC TL082 dan pada rangkaian R_{10} (resistor feedback op-amp) dihubung singkat, atau sama dengan menghubungkan output op-amp dengan negatif input. Pin 9 (V_{X+}) dari IC MC1495L dihubungkan

- ground. Variabel resistor (P_5) yang merupakan V_x offset , diatur sehingga tegangan AC pada output menjadi minimum.
- Demikian sebaliknya , pin 9 (V_x) IC MC1495L diberi input sinyal sinus dan pin 3 (input negatif) TL082 dihubungkan ground. Variabel resistor (P_4) diatur sehingga output pada rangkaian pengubah daya ke tegangan, tegangan AC menjadi minimum.
 - Variabel resistor (P_7) diatur untuk memperoleh tegangan DC menjadi nol.
 - Sinyal generator dan ground dilepas dari MC1495L dan TL082, serta R_{10} pada dihubungkan op-amp kembali.
 - Rangkaian dihubungkan ke beban sebuah lampu 100 Watt dengan tegangan AC 220 Volt. Diinginkan sensitivitas 1 milivolt/ 1 watt. Sehingga variabel resistor P_6 diatur untuk memperoleh tegangan output 100 milivolt.

IV.2. PENGUJIAN ALAT

Pengujian dilakukan untuk melihat setiap blok rangkaian berjalan dengan baik. Pengujian dilakukan pada blok rangkaian pengubah daya ke tegangan dan blok rangkaian ADC MAX180ACPL

IV.2.1. PENGUJIAN PENGUBAH DAYA KE TEGANGAN.

Sensitivitas rangkaian ini diinginkan adalah 1 milivolt / 1 watt. Pengujian dilakukan dengan menghubungkan beban AC yang resistif pada rangkaian,

kemudian diukur besarnya tegangan output (DC). Pada pengujian diperoleh sebagai berikut:

Tabel 3.1
Pengujian pengubah daya ke tegangan

Tegangan eban (Volt)	Arus beban (Ampere)	<i>Volt X Ampere</i>	Tegangan DC Output (miliVolt)
222	0.18	39.96	38
222	0.27	59.94	59
222	0.45	99.9	100
222	0.72	159.8	162

IV.2.2. PENGUJIAN ADC

Pengujian pada blok rangkaian ADC MAX180ACPL dilakukan untuk mengetahui ketelitian dan lineritas ADC tersebut. Resolusi ADC adalah 1.22 milivolt / bit. Pengujian dilakukan dengan cara memberi input tegangan DC pada ADC dan dilihat hasilnya pada PC dengan menggunakan program ADC. Hasil pengujian adalah sebagai berikut:

Tabel 3.2
Pengujian ADC MAX180ACPL

Tegangan input ADC (milivolt)	Hasil Penunjukan ADC (milivolt)
40	38
80	79

100	99
120	119
160	158
200	198
240	238
280	279
320	318
340	339

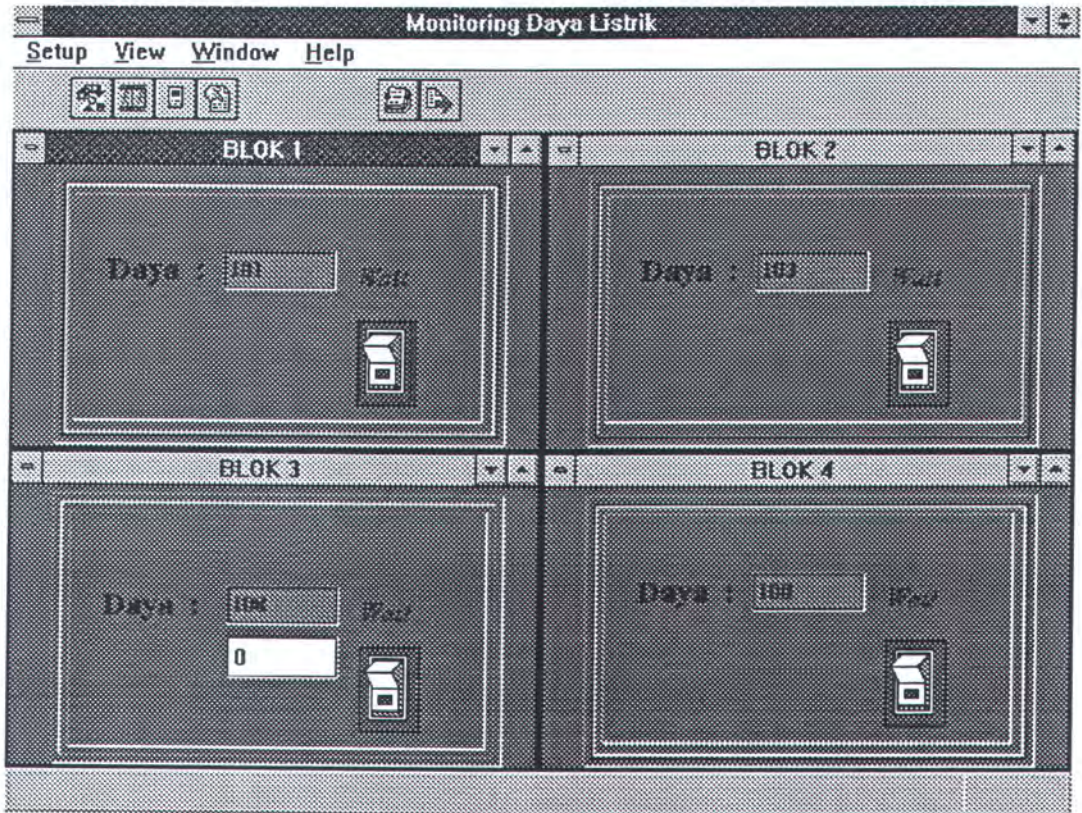
IV.3. PENGUKURAN.

Pengukuran dilakukan terhadap seluruh sistem secara menyeluruh dengan menghubungkan beban AC dan besarnya daya listrik akan terlihat pada layar monitor komputer. Demikian hasil program pada komputer dapat dilihat pada gambar 3.1

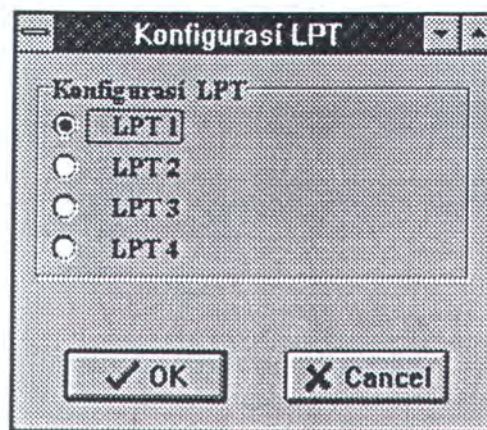
Tabel 3.3
Pengukuran Daya Listrik

Beban AC (watt)	Hasil pada komputer (Watt)
40	39
100	101
140	141
200	200

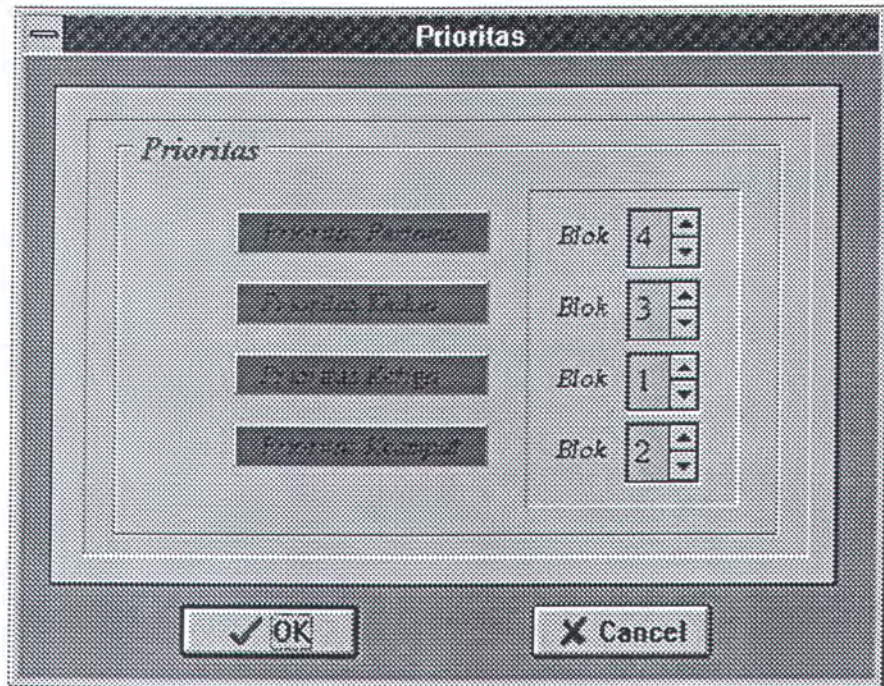
Hasil dari Program adalah sebagai berikut :



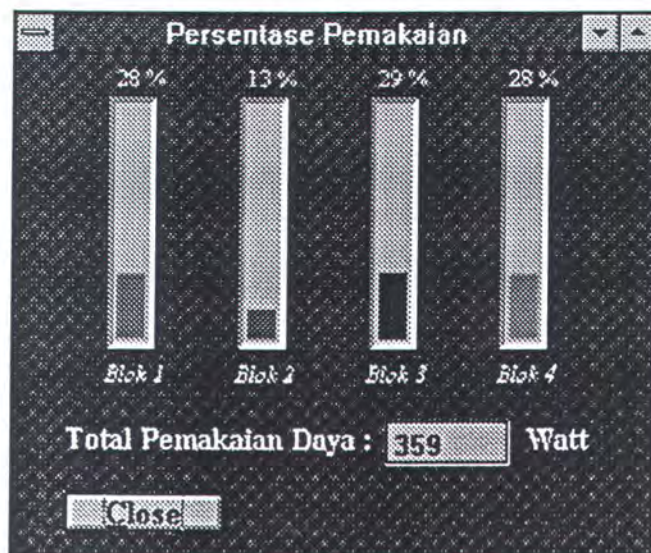
Gambar 4.1
Tampilan Menu dan Daya tiap blok



Gambar 4.2.
Tampilan konfigurasi LPT



Gambar 4.3
Tampilan Prioritas Blok



Gambar 4.4
Tampilan Persentase Pemakaian Daya

BATAS DAYA

Batas Daya Total

BATAS DAYA TOTAL : *Watt*

Blok 1

Batas Daya

Blok 2

Batas daya

Blok 3

Batas Daya

Blok 4

Batas daya

Gambar 4.5
Tampilan Batas Daya



BAB V

PENUTUP

BAB V

PENUTUP

V.1. KESIMPULAN

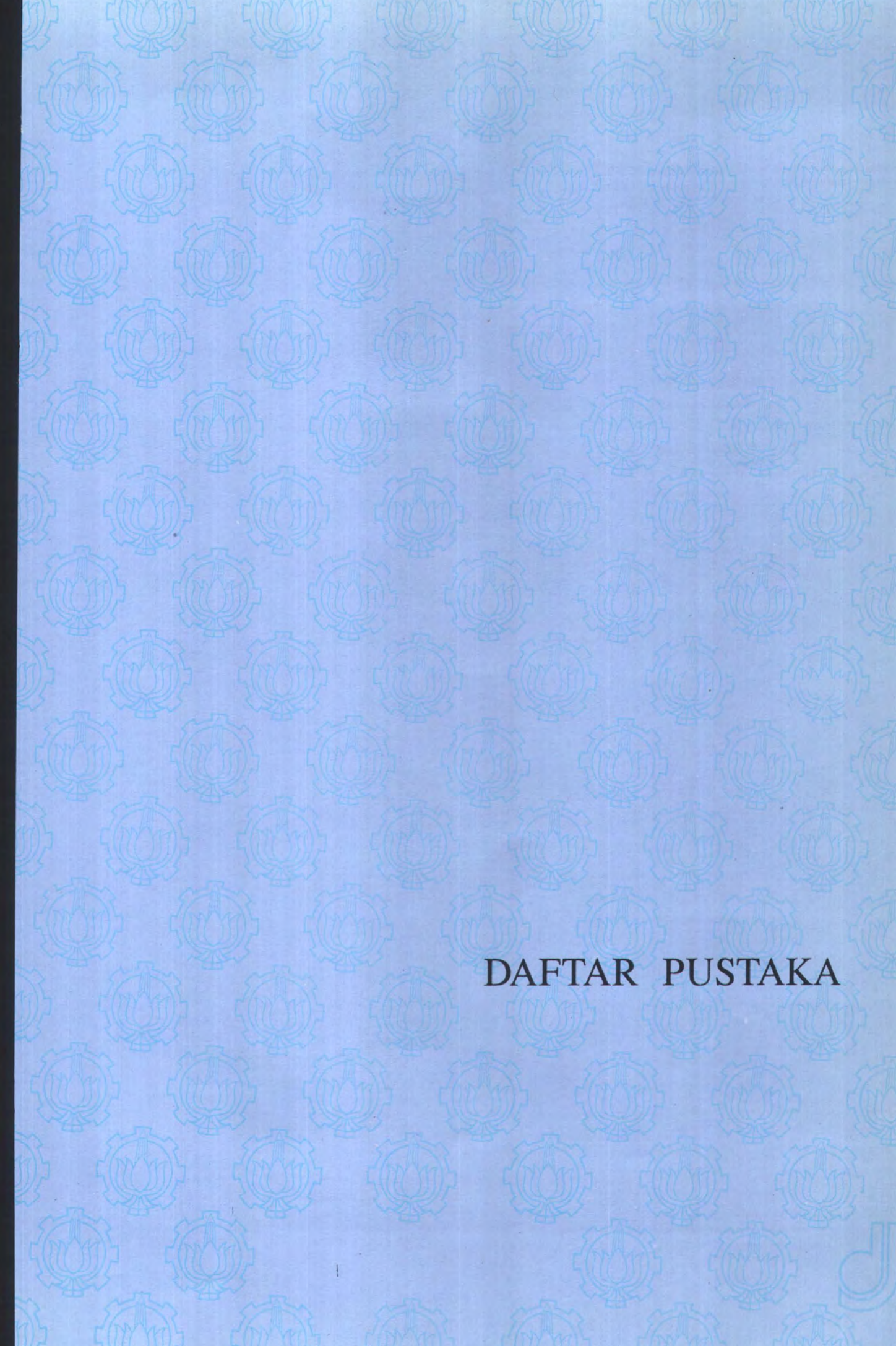
Sesudah melakukan pembahasan mulai dari bab-bab sebelumnya dalam tugas akhir ini, maka dapat dibuat kesimpulan :

1. Secara keseluruhan dan secara umum hasil pengukuran dari peralatan yang dibuat dapat bekerja untuk beban dengan daya yang cukup besar.
2. Diperlukan proses kalibrasi dan pengaturan rangkaian pengubah daya ke tegangan yang mempunyai ketelitian dan kesabaran yang tinggi unruk memperoleh hasil yang tepat.
3. Proses pengukuran dan pengontrolan daya listrik dapat dilakukan dengan IBM PC melalui paralel port (LPT), yang mana memudahkan pemasangan alat ke IBM PC tanpa harus membongkar komputer dahulu.
4. *Multiplier analog* dapat digunakan untuk mengukur besarnya daya listrik, dan dengan menggunakan multiplier analog dapat lebih menyederhanakan hardware.
5. Paralel port (LPT) mempunyai jumlah bit input/output yang tidak terlalu banyak, sehingga bila digunakan untuk aplikasi peralatan luar yang memanfaatkan jumlah bit yang banyak, diperlukan komponen luar untuk hal tersebut.

V.2. SARAN-SARAN

Saran-saran yang diharapkan dapat berguna untuk pengembangan dan penggunaan lebih lanjut dari tugas akhir ini adalah :

1. Diperlukan Low Pass filter yang baik dari keluaran pengubah daya ke tegangan agar noise yang timbul dapat dihilangkan.
2. Ketepatan pengukuran juga tergantung dari ketelitian pengubah analog ke digital, sehingga diperlukan rangkaian akuisisi data yang baik.



DAFTAR PUSTAKA

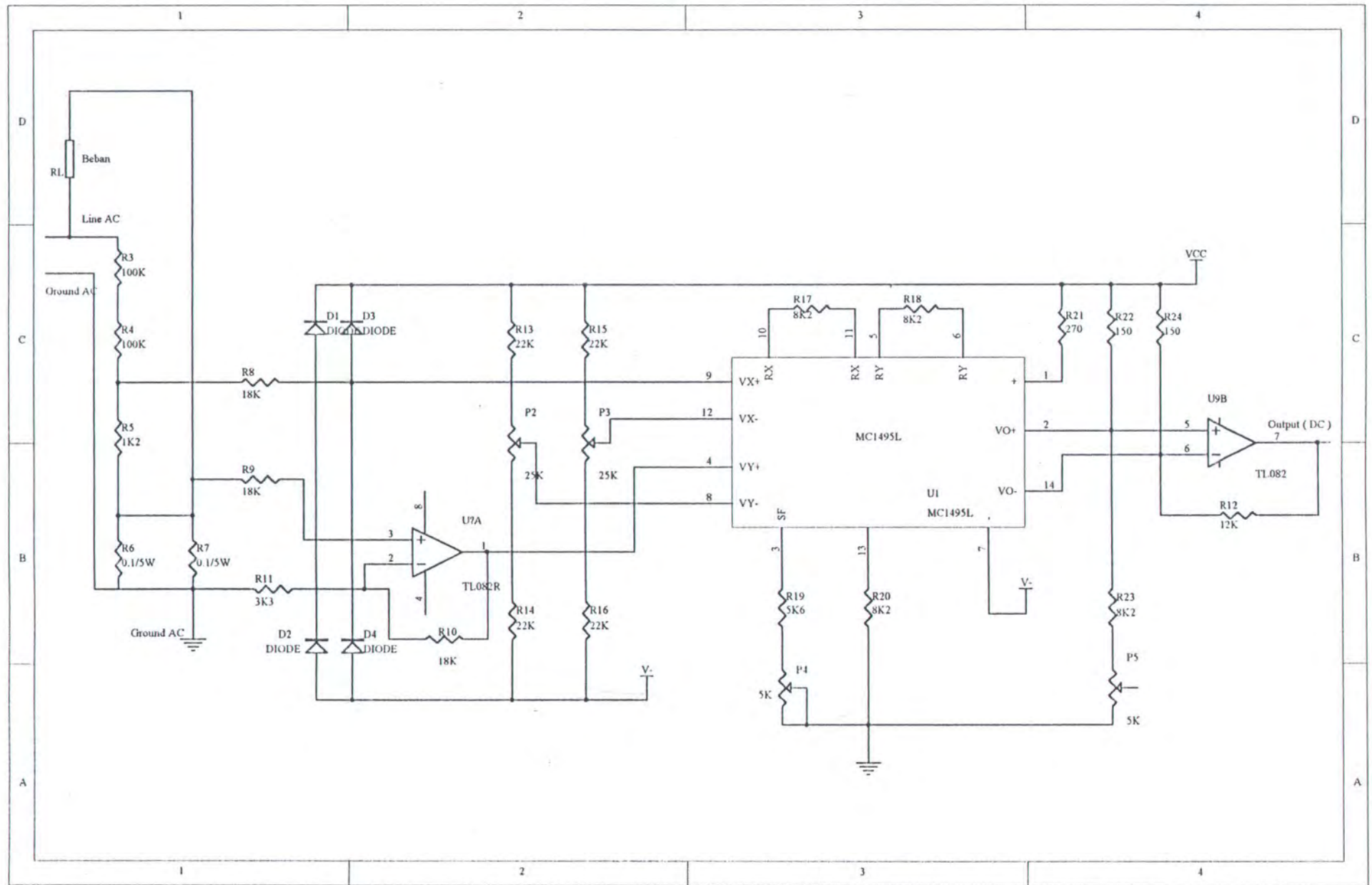
DAFTAR PUSTAKA

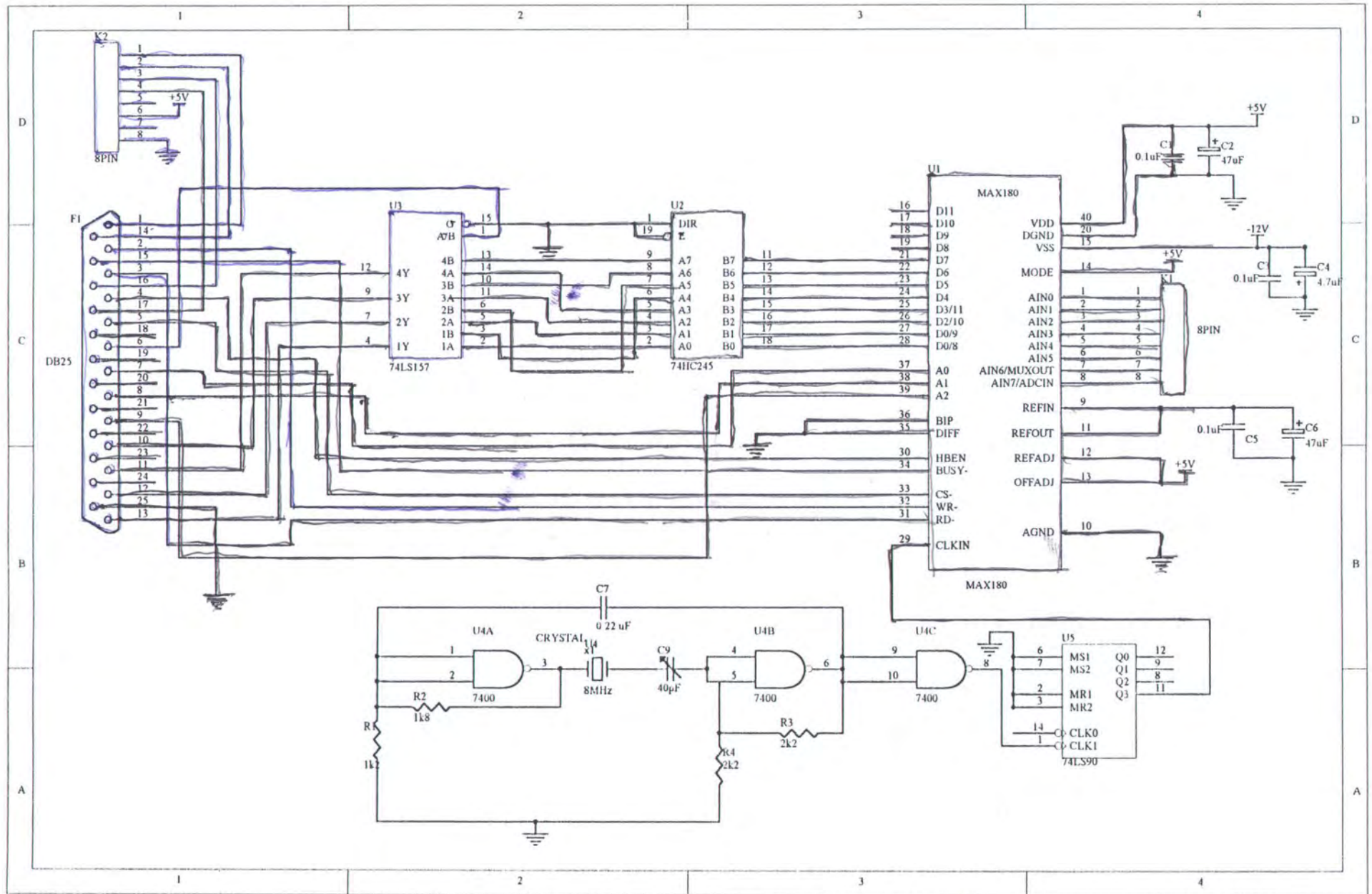
1. Gunawan, Hanapi,Ir.Drs., Eugene C Lister, 1988, **Mesin dan Rangkaian Listrik**, Penerbit Erlangga, Jakarta
2. Lemon , 1991, **Wattmeter**, Elektor Electronics, April, hal 32 - 35
3. Maxim, 1994, **Max 180ACPL Data Sheet**, Maxim Integrated Product
4. Motorola, **Motorolla Linear/Interface Devices**, Motorola Semiconductor, hal 11-42.
5. Roy W.Body, **Hardware Software And Aplication**, Mc Hill. Intel Microprocessor.
6. Soedjana Sapii,Dr., Osamu Nishino,Dr.,1975, **Pengukuran dan Alat-alat ukur listrik**, Pradnya Paramita , Jakarta.
7. Xavier Pacheco & Steve Teixeira, 1995, **Delphi Developer's Guide**, SAMS Publishing.

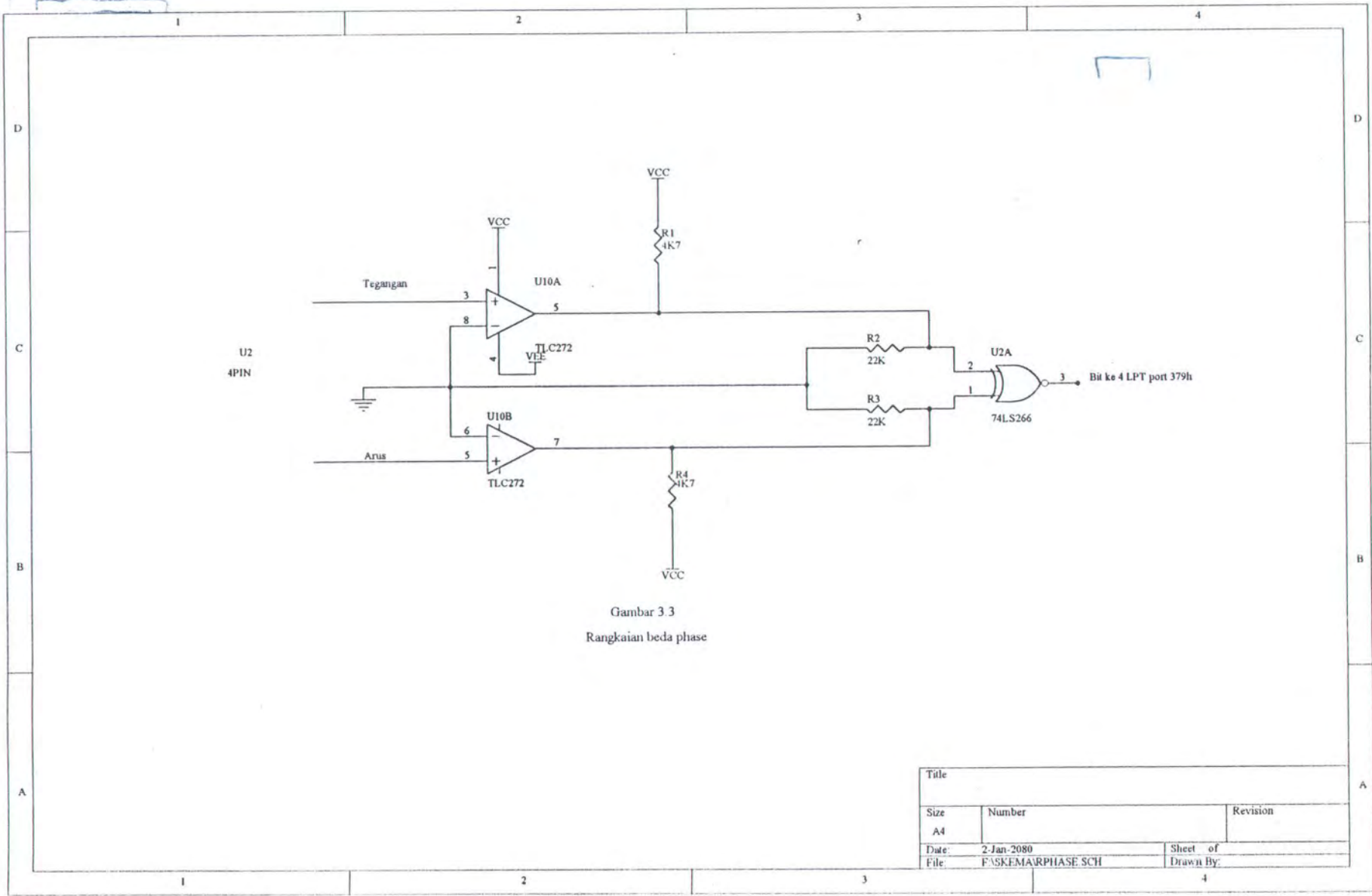


LAMPIRAN A

GAMBAR RANGKAIAN



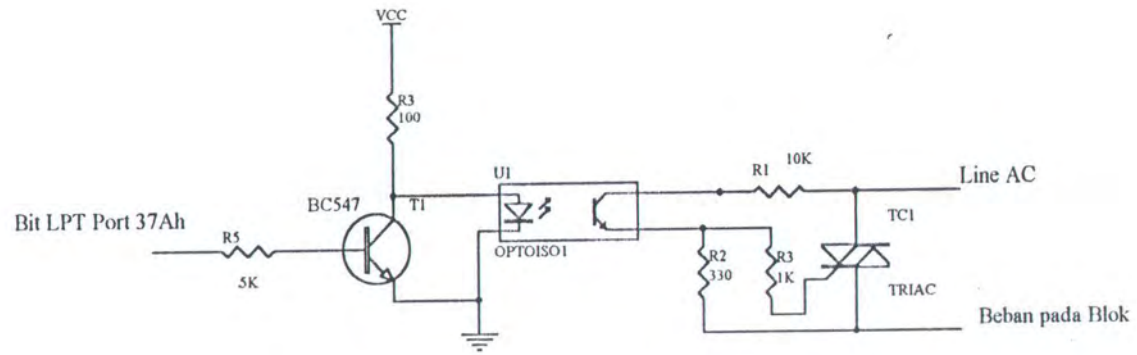




Gambar 3.3
Rangkaian beda phase

Title		
Size A4	Number	Revision
Date:	2-Jan-2080	Sheet of
File:	F:\SKEMA\RPHASE SCH	Drawn By:

SWITCH TEGANGAN AC





LAMPIRAN B

LISTING PROGRAM

Main. pas

```
unit Main;
interface
uses WinTypes, WinProcs, SysUtils, Classes, Graphics, Forms, Controls, Menus,
  StdCtrls, Dialogs, Buttons, Messages, ExtCtrls, VBXCtrl, Gauge, Grids,
  Calendar;
type
TMainForm = class(TForm)  MainMenu1: TMainMenu;
  Panel1: TPanel; StatusLine: TPanel; File1: TMenuItem;
  FileNewItem: TMenuItem; FileOpenItem: TMenuItem; Panel2: TPanel;
  FileCloseItem: TMenuItem; Window1: TMenuItem; Help1: TMenuItem;
  N1: TMenuItem; FileExitItem: TMenuItem; WindowCascadeItem: TMenuItem;
  WindowTileItem: TMenuItem; WindowArrangeItem: TMenuItem;
  HelpAboutItem: TMenuItem; FileSaveItem: TMenuItem;
  Edit1: TMenuItem; CutItem: TMenuItem; WindowMinimizeItem: TMenuItem;
  SpeedPanel: TPanel; CutBtn: TSpeedButton; ExitBtn: TSpeedButton;
  SpeedButton1: TSpeedButton; SpeedButton2: TSpeedButton;
  Timer1: TTimer; SpeedButton3: TSpeedButton;
  SpeedButton4: TSpeedButton; procedure FormCreate(Sender: TObject);
  procedure FileNewItemClick(Sender: TObject);
  procedure WindowCascadeItemClick(Sender: TObject);
  procedure UpdateMenuItems(Sender: TObject);
  procedure WindowTileItemClick(Sender: TObject);
  procedure WindowArrangeItemClick(Sender: TObject);
  procedure FileCloseItemClick(Sender: TObject);
  procedure FileOpenItemClick(Sender: TObject);
  procedure FileExitItemClick(Sender: TObject);
  procedure FileSaveItemClick(Sender: TObject);
  procedure FileSaveAsItemClick(Sender: TObject);
  procedure CutItemClick(Sender: TObject);
  procedure CopyItemClick(Sender: TObject);
  procedure PasteItemClick(Sender: TObject);
  procedure WindowMinimizeItemClick(Sender: TObject);
  procedure FormDestroy(Sender: TObject);
  procedure SpeedButton1Click(Sender: TObject);
  procedure SpeedButton2Click(Sender: TObject);
  procedure Timer1Timer(Sender: TObject);
  procedure SpeedButton3Click(Sender: TObject);
  procedure FormShow(Sender: TObject);
  procedure CutBtnClick(Sender: TObject);
  procedure SpeedButton4Click(Sender: TObject);
private
  procedure CreateMDIChild(const Name: string);
  procedure ShowHint(Sender: TObject);
public
end;
function day1:real; function day2:real; function day3:real;
function day4:real; function psn1:longint; function psn2:longint;
function psn3:longint; function psn4:longint; function j:longint; function fdy:real;
var
  MainForm: TMainForm;
implementation
{$R *.DFM}
uses ChildWin,max1,max2,max3,max4,max5,max6,max7,max8;
```

```

Screen.OnActiveFormChange := nil;
end;
procedure TMainForm.SpeedButton1Click(Sender: TObject);
begin
mainform.enabled:=false;FORM5.SHOW;
end;
procedure TMainForm.SpeedButton2Click(Sender: TObject);
begin
mainform.enabled:=false;form6.show;
end;
procedure TMainForm.Timer1Timer(Sender: TObject);
var busy,data1,data2,data3,data1r,data1a,data2r,data3r,data3a,cek:integer;
    kanal,data,s,n,data1a1,data1a2,data2a1,data2a2,data3a1,data3a2:integer;
    msw1,msw2,msw3,msw4 : integer;
    jml,jhasil,cekt,rhasil:real;
    sel,tang1,tang2,wkt1,wkt2,beda:tdatetime;
    phas,phase:integer;
    j,m,d,mid,j1,m1,d1,md1,j2,m2,d2,md2,md:word;
    j3,m3,d3,md3:word;
    hasil: array [1..1001] of real;
label ulang,terus,balik,tnol,tbh,a1,a2,a3,a4,a5;
begin
begin
tang1:=time;decodetime(tang1,j,m,d,mid);mid:=mid+10;
a2:
phase:=port[$379];phas:=phase and $08;
if phas=$08 then
begin
wkt1:=time;
a3: phase:=port[$379]; phas:=phase and $08;
if phas=$0 then
begin
wkt2:=time;
a5:
phase:=port[$379]; phas:=phase and $08;
if phas=$08 then
begin
sel:=time; goto a4;
end else
goto a5; end
else
begin
tang2:=time; decodetime(tang2,j2,m2,d2,md2);
if md2<=mid then
goto a3
else
begin
fdaya:=0; goto a1;
end; end; end;
tang2:=time; decodetime(tang2,j2,m2,d2,md2);
if md2>=mid then
begin
fdaya:=0; goto a1;
end else
goto a2;
a4: decodetime(sel,j1,m1,d1,md1); decodetime(wkt2,j2,m2,d2,md2);

```

```

    Child: TMDIChild;
begin
    Child := TMDIChild.Create(Application);
    Child.Caption := Name;
end;
procedure TMainForm.FileNewItemClick(Sender: TObject);
begin
    mainform.enabled:=false;FORM5.SHOW;
end;
procedure TMainForm.FileOpenItemClick(Sender: TObject);
begin
    mainform.enabled:=false;form6.show;
end;
procedure TMainForm.FileCloseItemClick(Sender: TObject);
begin
    mainform.enabled:=false;form8.show;
end;
procedure TMainForm.FileExitItemClick(Sender: TObject);
begin
    Close;
end;
procedure TMainForm.CutItemClick(Sender: TObject);
begin
    mainform.enabled:=false;form7.show;
end;
procedure TMainForm.WindowCascadeItemClick(Sender: TObject);
begin
    Cascade;
end;
procedure TMainForm.WindowTileItemClick(Sender: TObject);
begin
    Tile;
end;
procedure TMainForm.WindowArrangeItemClick(Sender: TObject);
begin
    ArrangeIcons;
end;
procedure TMainForm.WindowMinimizeItemClick(Sender: TObject);
var
    I: Integer;
begin
    for I := MDIChildCount - 1 downto 0 do
        MDIChildren[I].WindowState := wsMinimized;
    end;
end;
procedure TMainForm.UpdateMenuItems(Sender: TObject);
begin
    FileCloseItem.Enabled := MDIChildCount > 0;
    CutItem.Enabled := MDIChildCount > 0;
    CutBtn.Enabled := MDIChildCount > 0;
    WindowCascadeItem.Enabled := MDIChildCount > 0;
    WindowTileItem.Enabled := MDIChildCount > 0;
    WindowArrangeItem.Enabled := MDIChildCount > 0;
    WindowMinimizeItem.Enabled := MDIChildCount > 0;
end;
procedure TMainForm.FormDestroy(Sender: TObject);
begin

```

```

var daya:array[0..9] of real;
    pr10,pr20,pr30,pr40:integer;
    mttl,fdaya:real;
    pr1a,pr2a,pr3a,pr4a:integer;
    jp:longint;
function daya1:real;
begin
daya1:=daya[0];
end;
function daya2:real;
begin
daya2:=daya[1];
end;
function daya3:real;
begin
daya3:=daya[2];
end;
function daya4:real;
begin
daya4:=daya[3];
end;
function psn1:longint;
begin
psn1:=round(daya1);
end;
function psn2:longint;
begin
psn2:=round(daya2);
end;
function psn3:longint;
begin
psn3:=round(daya3);
end;
function psn4:longint;
begin
psn4:=round(daya4);
end;
function j:longint;
begin
j:=jp+1;
end;
function fdy:real;
begin
fdy:=fdaya;
end;

procedure TMainForm.FormCreate(Sender: TObject);
begin
    Application.OnHint := ShowHint; Screen.OnActiveFormChange := UpdateMenuItems;
end;
procedure TMainForm.ShowHint(Sender: TObject);
begin
    StatusLine.Caption := Application.Hint;
end;
procedure TMainForm.CreateMDIChild(const Name: string);
var

```

```

md:=md2-md1;  fdaya:=md;
al:
end;
s:=0; cekt:=0;
for kanal:=0 to 3 do
begin
rhasil:=0;  jhasil:=0 ;
for n:=1 to 1000 do
begin
hasil[n]:=0;
port[$378]:=$0b or s;  port[$378]:=$03 or s;  port[$378]:=$02 or s;
port[$378]:=$03 or s;  port[$378]:=$0b or s;
ulang:
port[$378]:=$03 or s;  port[$378]:=$01 or s;  data1:=port[$379];  port[$378]:=$11 or s;
data2:=port[$379];  port[$378]:=$01 or s;  port[$378]:=$03 or s;
port[$378]:=$0b or s;  port[$378]:=$0f or s;  port[$378]:=$07 or s;
port[$378]:=$05 or s;  data3:=port[$379];  port[$378]:=$07 or s;
port[$378]:=$0f or s;  data1a:=data1 and $f0;  data1a1:=data1a and $70;
data1a2:=data1a and $80;  data1a2:=not data1a2;data1a2:=data1a2 and $80;
data1r:=data1a1 or data1a2 ;  data1r:=data1r shr 4;  data2:=data2 and $f0;
data2a1:=data2 and $70;  data2a2:=data2 and $80;
data2a2:=not data2a2;data2a2:=data2a2 and $80;  data2r:=data2a1 or data2a2;
data3a:=data3 and $f0;  data3a1:=data3a and $70;  data3a2:=data3a and $80;
data3a2:=not data3a2;data3a2:=data3a2 and $80;  data3r:=data3a1 or data3a2 ;
data3r:=data3r shl 4;  data:=data1r or data2r or data3r;  hasil[n]:=(5000/4096) * data;
jhasil:=jhasil+hasil[n];  end;
rhasil:=jhasil/1000;  s:=s+$20;
daya[kanal]:=rhasil;
if kanal=0 then
begin
if daya[0]<=20 then
begin
daya[0]:=0;goto tnol;
end;  end;
if kanal=1 then
begin
if daya[1]<=20 then
begin
daya[1]:=0;goto tnol;
end;  end;
if kanal=2 then
begin
if daya[2]<=36 then
begin
daya[2]:=0;goto tnol;
end;  end;
if kanal=3 then
begin
if daya[3]<=20 then
begin
daya[3]:=0;goto tnol;
end;  end;tbh:
if kanal=0 then daya[0]:=daya[0] + 6 ;
if kanal=1 then daya[1]:=daya[1] +12;
if kanal=2 then daya[2]:=daya[2] + 13;
if kanal=3 then daya[3]:=(daya[3]+4)/2.24+3;

```



```

tnol:
  end;
jnd:=daya[0]+daya[1]+daya[2]+daya[3];
jp:=round(jnd);  mttl:=bsttl+25000;
pr1a:=pr1+ 1;  pr2a:=pr2+ 2;
pr3a:=pr3+ 3;  pr4a:=pr4+ 4;
if jnd >= mttl then
  begin
  case pr1a of
    1:cekt:=cekt+daya[0];
    2:cekt:=cekt+daya[1];
    3:cekt:=cekt+daya[2];
    4:cekt:=cekt+daya[3];
  end;
  if cekt >= mttl then
    begin
    port[$37a]:=$4;
    if form1.biswitch1.pon=true then
      form1.biswitch1.pon:=false;
    if form2.biswitch1.pon=true then
      form2.biswitch1.pon:=false;
    if form3.biswitch1.pon=true then
      form3.biswitch1.pon:=false;
    if form4.biswitch1.pon=true then
      form4.biswitch1.pon:=false;
    goto balik;  end;
    case pr2a of
      1:cekt:=cekt+daya[0];
      2:cekt:=cekt+daya[1];
      3:cekt:=cekt+daya[2];
      4:cekt:=cekt+daya[3];
    end;
    if cekt >=mttl then
      begin
      case pr1a of
        1: begin
          port[$37a]:=$4 or sw1;
          if form2.biswitch1.pon=true then
            form2.biswitch1.pon:=false;
          if form3.biswitch1.pon=true then
            form3.biswitch1.pon:=false;
          if form4.biswitch1.pon=true then
            form4.biswitch1.pon:=false;
          end;
        2: begin
          port[$37a]:=$4 or sw2;
          if form1.biswitch1.pon=true then
            form1.biswitch1.pon:=false;
          if form3.biswitch1.pon=true then
            form3.biswitch1.pon:=false;
          if form4.biswitch1.pon=true then
            form4.biswitch1.pon:=false;
          end;
        3: begin
          port[$37a]:=$0 or sw3;
          if form1.biswitch1.pon=true then

```

```

form1.biswitch1.pon:=false;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end;
4: begin
port[$37a]:=$4 or sw4;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end; end ; goto balik; end;
case pr3a of
1:cekt:=cekt+daya[0];
2:cekt:=cekt+daya[1];
3:cekt:=cekt+daya[2];
4:cekt:=cekt+daya[3];
end;
if cekt >= mttl then
begin
case pr1a of
1: begin
case pr2a of
1:begin
port[$37a]:=$4 or sw1 or sw1;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw1 or sw2;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw1 or sw3;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4: begin
port[$37a]:=$4 or sw1 or sw4;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ; end ; end ;

```

```

2: begin
  case pr2a of
    1: begin
      port[$37a]:=$4 or sw2 or sw1;
      if form3.biswitch1.pon=true then
        form3.biswitch1.pon:=false;
      if form4.biswitch1.pon=true then
        form4.biswitch1.pon:=false;
      end;
    2: begin
      port[$37a]:=$4 or sw2 or sw2;
      if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
      if form3.biswitch1.pon=true then
        form3.biswitch1.pon:=false;
      if form4.biswitch1.pon=true then
        form4.biswitch1.pon:=false;
      end ;
    3: begin
      port[$37a]:=$0 or sw2 or sw3;
      if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
      if form4.biswitch1.pon=true then
        form4.biswitch1.pon:=false;
      end ;
    4: begin
      port[$37a]:=$4 or sw2 or sw4;
      if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
      if form3.biswitch1.pon=true then
        form3.biswitch1.pon:=false;
      end      end ;      end ;
  3: begin
    case pr2a of
      1: begin
        port[$37a]:=$0 or sw3 or sw1;
        if form2.biswitch1.pon=true then
          form2.biswitch1.pon:=false;
        if form4.biswitch1.pon=true then
          form4.biswitch1.pon:=false;
        end ;
      2: begin
        port[$37a]:=$0 or sw3 or sw2;
        if form1.biswitch1.pon=true then
          form1.biswitch1.pon:=false;
        if form4.biswitch1.pon=true then
          form4.biswitch1.pon:=false;
        end ;
      3: begin
        port[$37a]:=$0 or sw3 or sw3;
        if form1.biswitch1.pon =true then
          form1.biswitch1.pon:=false;
        if form2.biswitch1.pon=true then
          form2.biswitch1.pon:=false;
        if form4.biswitch1.pon=true then
          form4.biswitch1.pon:=false;

```

```

    end ;
4: begin
    port[$37a]:=$0 or sw3 or sw4;
    if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
    if form1.biswitch1.pon=true then
    form1.biswitch1.pon:=false;
    end
    end ;
end ;
4: begin
case pr2a of
1:begin
    port[$37a]:=$4 or sw4 or sw1;
    if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
    if form3.biswitch1.pon=true then
    form3.biswitch1.pon:=false;
    end ;
2: begin
    port[$37a]:=$4 or sw4 or sw2;
    if form3.biswitch1.pon=true then
    form3.biswitch1.pon:=false;
    if form1.biswitch1.pon=true then
    form1.biswitch1.pon:=false;
    end ;
3:begin
    port[$37a]:=$0 or sw4 or sw3;
    if form1.biswitch1.pon=true then
    form1.biswitch1.pon:=false;
    if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
    end ;
4: begin
    port[$37a]:=$4 or sw4 or sw4;
    if form1.biswitch1.pon=true then
    form1.biswitch1.pon:=false;
    if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
    if form3.biswitch1.pon=true then
    form3.biswitch1.pon:=false;
    end
    end ;    end ;    end ;    end:    goto balik;    end;
case pr4a of
1:cekt:=cekt+daya[0];
2:cekt:=cekt+daya[1];
3:cekt:=cekt+daya[2];
4:cekt:=cekt+daya[3];
end;
if cekt >= mttl then
begin
case pr1a of
1:begin
    case pr2a of
1:begin
    case pr3a of
1:begin

```

```

port[$37a]:=$4 or sw1 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw1 or sw2 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw1 or sw3;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$4 or sw1 or sw4 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;          end;          end;
2:begin
case pr3a of
1:begin
port[$37a]:=$4 or sw1 or sw2;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw1 or sw2 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw1 or sw2 or sw3;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$4 or sw1 or sw2 or sw4 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;          end;          end;
3:begin

```

```

case pr3a of
1:begin
  port[$37a]:=$0 or sw1 or sw3;
  if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
  if form4.biswitch1.pon=true then
    form4.biswitch1.pon:=false;
  end ;
2: begin
  port[$37a]:=$4 or sw1 or sw2 or sw3;
  if form4.biswitch1.pon=true then
    form4.biswitch1.pon:=false;
  end ;
3:begin
  port[$37a]:=$0 or sw1 or sw3;
  if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
  if form4.biswitch1.pon=true then
    form4.biswitch1.pon:=false;
  end ;
4:begin
  port[$37a]:=$4 or sw1 or sw3 or sw4;
  if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
  end ;          end;          end;
4:begin
case pr3a of
1:begin
  port[$37a]:=$4 or sw1 or sw4 ;
  if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
  if form3.biswitch1.pon=true then
    form3.biswitch1.pon:=false;
  end ;
2: begin
  port[$37a]:=$4 or sw1 or sw2 or sw4 ;
  if form3.biswitch1.pon=true then
    form3.biswitch1.pon:=false;
  end ;
3:begin
  port[$37a]:=$0 or sw1 or sw3 or sw4;
  if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
  end ;
4:begin
  port[$37a]:=$4 or sw1 or sw4 ;
  if form2.biswitch1.pon=true then
    form2.biswitch1.pon:=false;
  if form3.biswitch1.pon=true then
    form3.biswitch1.pon:=false;
  end ;          end; end;          end;          end;
2:begin
case pr2a of
1:begin
case pr3a of
1:begin

```

```

port[$37a]:=$4 or sw1 or sw2 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw1 or sw2 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw1 or sw2 or sw3;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$4 or sw1 or sw2 or sw4 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;          end;          end;
2:begin
case pr3a of
1:begin
port[$37a]:=$4 or sw1 or sw2;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw2 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw2 or sw3;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false ;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$4 or sw2 or sw4 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;          end;          end;
3:begin

```

```

case pr3a of
  1:begin
    port[$37a]:=$0 or sw1 or sw2 or sw3;
    if form4.biswitch1.pon=true then
      form4.biswitch1.pon:=false;
    end ;
  2: begin
    port[$37a]:=$4 or sw2 or sw3;
    if form1.biswitch1.pon=true then
      form1.biswitch1.pon:=false;
    if form4.biswitch1.pon=true then
      form4.biswitch1.pon:=false;
    end ;
  3:begin
    port[$37a]:=$0 or sw2 or sw3;
    if form1.biswitch1.pon=true then
      form1.biswitch1.pon:=false;
    if form4.biswitch1.pon=true then
      form4.biswitch1.pon:=false;
    end ;
  4:begin
    port[$37a]:=$4 or sw2 or sw3 or sw4 ;
    if form1.biswitch1.pon=true then
      form1.biswitch1.pon:=false;
    end ;          end;          end;
4:begin
  case pr3a of
    1:begin
      port[$37a]:=$4 or sw1 or sw2 or sw4 ;
      if form3.biswitch1.pon=true then
        form3.biswitch1.pon:=false;
      end ;
    2: begin
      port[$37a]:=$4 or sw2 or sw4 ;
      if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
      if form3.biswitch1.pon=true then
        form3.biswitch1.pon:=false;
      end ;
    3:begin
      port[$37a]:=$0 or sw2 or sw3 or sw4;
      if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
      end ;
    4:begin
      port[$37a]:=$4 or sw2 or sw4 ;
      if form2.biswitch1.pon=true then
        form2.biswitch1.pon:=false;
      if form3.biswitch1.pon=true then
        form3.biswitch1.pon:=false;
      end ;          end;          end;          end;          end ;
3:begin
  case pr2a of
    1:begin
      case pr3a of
        1:begin

```



```

port[$37a]:=$0 or sw1 or sw3;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$0 or sw1 or sw2 or sw3;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw1 or sw3;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$0 or sw1 or sw3 or sw4 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
end ;          end;          end;
2:begin
case pr3a of
1:begin
port[$37a]:=$0 or sw1 or sw2 or sw3;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$0 or sw3 or sw2 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw2 or sw3;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$0 or sw3 or sw2 or sw4 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
end ;
end;          end;
3:begin
case pr3a of
1:begin
port[$37a]:=$0 or sw1 or sw3;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;

```

```

    if form4.biswitch1.pon=true then
    form4.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$0 or sw2 or sw3;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw3;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form4.biswitch1.pon=true then
form4.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$0 or sw3 or sw4 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
end ;          end;          end;
4:begin
case pr3a of
1:begin
port[$37a]:=$0 or sw1 or sw3 or sw4 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$0 or sw3 or sw2 or sw4 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw3 or sw4;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$0 or sw3 or sw4 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
end ; end;          end;          end;          end;
4:begin
case pr2a of
1:begin
case pr3a of

```

```

1:begin
port[$37a]:=$4 or sw1 or sw4 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw1 or sw2 or sw4 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw1 or sw3 or sw4;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$4 or sw1 or sw4 ;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ; end; end;
2:begin
case pr3a of
1:begin
port[$37a]:=$4 or sw1 or sw2 or sw4;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ;
2: begin
port[$37a]:=$4 or sw4 or sw2 ;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
end ;
3:begin
port[$37a]:=$0 or sw4 or sw2 or sw3;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
end ;
4:begin
port[$37a]:=$4 or sw2 or sw4 ;
if form1.biswitch1.pon=true then
form1.biswitch1.pon:=false;
if form3.biswitch1.pon=true then
form3.biswitch1.pon:=false;
end ; end; end;
3:begin
case pr3a of
1:begin
port[$37a]:=$0 or sw1 or sw3 or sw4;
if form2.biswitch1.pon=true then
form2.biswitch1.pon:=false;

```

```

    end ;
2: begin
    port[$37a]:=$4 or sw4 or sw2 or sw3;
    if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
    end ;
3:begin
    port[$37a]:=$0 or sw4 or sw3;
    if form2.biswitch1.pon=true then
        form2.biswitch1.pon:=false;
    if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
    end ;
4:begin
    port[$37a]:=$4 or sw3 or sw4 ;
    if form1.biswitch1.pon=true then
        form1.biswitch1.pon:=false;
    if form2.biswitch1.pon=true then
        form2.biswitch1.pon:=false;
    end ; end; end;
4:begin
    case pr3a of
    1:begin
        port[$37a]:=$4 or sw1 or sw4 ;
        if form2.biswitch1.pon=true then
            form2.biswitch1.pon:=false;
        if form3.biswitch1.pon=true then
            form3.biswitch1.pon:=false;
        end ;
    2: begin
        port[$37a]:=$4 or sw2 or sw4 ;
        if form1.biswitch1.pon=true then
            form1.biswitch1.pon:=false;
        if form3.biswitch1.pon=true then
            form3.biswitch1.pon:=false;
        end ;
    3:begin
        port[$37a]:=$0 or sw3 or sw4;
        if form1.biswitch1.pon=true then
            form1.biswitch1.pon:=false;
        if form2.biswitch1.pon=true then
            form2.biswitch1.pon:=false;
        end ;
    4:begin
        port[$37a]:=$4 or sw4 ;
        if form1.biswitch1.pon=true then
            form1.biswitch1.pon:=false;
        if form2.biswitch1.pon=true then
            form2.biswitch1.pon:=false;
        if form3.biswitch1.pon=true then
            form3.biswitch1.pon:=false;
        end ;end; end; end; end; end; end; goto balik; end; end;
    balik: end;
    procedure TMainForm.SpeedButton3Click(Sender: TObject);
    begin
        mainform.enabled:=false;

```

```

form7.show;end;
procedure TMainForm.FormShow(Sender: TObject);
begin
port[$37a]:=$b;
jp:=20000;end;
procedure TMainForm.CutBtnClick(Sender: TObject);
begin tile;end;
procedure TMainForm.SpeedButton4Click(Sender: TObject);
begin
mainform.enabled:=false ;
form8.show;end;end.

```

MAX 1.Pas

```

unit Max1;
interface
uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, VBXCtrl, Switch, StdCtrls, ExtCtrls;
type
  TForm1 = class(TForm)  Bevel1: TBevel;  Panel1: TPanel;  Bevel2: TBevel;  Label1: TLabel;
    Edit1: TEdit;  Label2: TLabel;  BiSwitch1: TBiSwitch;  Timer1: TTimer;
    procedure Timer1Timer(Sender: TObject);
    procedure BiSwitch1Off(Sender: TObject);
    procedure BiSwitch1On(Sender: TObject);
    procedure FormShow(Sender: TObject);
  private
    public; end;
    function sw1:integer;
  var
    Form1: TForm1;
  implementation
    uses main,childwin,max2,max3,max4,max5;
    {$R *.DFM}
    var s1,rbs1,rttl:integer;
    function sw1:integer;
    begin
    sw1:=s1;
    end;
    procedure TForm1.Timer1Timer(Sender: TObject);
    var dy1:longint;
    begin
    dy1:=round(daya1);edit1.text:=floattostrf(dy1,ffgeneral,4,0);rbs1:=bs1;rttl:=bttl;
    if rbs1<rttl then
    begin
    if dy1>=rbs1 then
    begin
    s1:=$0;
    port[$37a]:=$0 or sw2 or sw3 or sw4;
    if biswitch1.pon=true then
    biswitch1.pon:=false;
    end;end;end;
    procedure TForm1.BiSwitch1Off(Sender: TObject);
    begin
    s1:=$0;port[$37a]:=$0 or sw2 or sw3 or sw4;
    end;

```

```

procedure TForm1.BiSwitch1On(Sender: TObject);
begin
s1:=$1;port[$37a]:=$1 or sw2 or sw3 or sw4;
end;
procedure TForm1.FormShow(Sender: TObject);
begin
s1:=$1;rbs1:=25001;rttl:=25000;
end;end.

```

Max 1.Pas

```

unit Max1;
interface
uses
  SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
  Forms, Dialogs, VBXCtrl, Switch, StdCtrls, ExtCtrls;
type
TForm1 = class(TForm)
  Bevel1: TBevel; Panel1: TPanel; Bevel2: TBevel; Label1: TLabel;
  Edit1: TEdit; Label2: TLabel; BiSwitch1: TBiSwitch; Timer1: TTimer;
  procedure Timer1Timer(Sender: TObject);
  procedure BiSwitch1Off(Sender: TObject);
  procedure BiSwitch1On(Sender: TObject);
  procedure FormShow(Sender: TObject);
  private
  public
  end;
  function sw1:integer;
var
  Form1: TForm1;
implementation
  uses main,childwin,max2,max3,max4,max5;
  {$R *.DFM}
  var s1,rbs1,rttl:integer;
  function sw1:integer;
  begin
  sw1:=s1;
  end;
  procedure TForm1.Timer1Timer(Sender: TObject);
  var dyl:longint;
  begin
  dyl:=round(day1);
  edit1.text:=floattostrf(dyl,ffgeneral,4,0);
  rbs1:=bs1;
  rttl:=bttl;
  if rbs1<rttl then
  begin
  if dyl>=rbs1 then
  begin
  s1:=$0;
  port[$37a]:=$0 or sw2 or sw3 or sw4;
  if biswitch1.pon=true then
  biswitch1.pon:=false;
  end;end;end;
  procedure TForm1.BiSwitch1Off(Sender: TObject);
  begin

```

```

s1:=$0;port[$37a]:=$0 or sw2 or sw3 or sw4;
end;
procedure TForm1.BiSwitch1On(Sender: TObject);
begin
s1:=$1;port[$37a]:=$1 or sw2 or sw3 or sw4;
end;
procedure TForm1.FormShow(Sender: TObject);
begin
s1:=$1;rbs1:=25001;rttl:=25000;
end;end.

```

Max 5.Pas

```

unit Max5;
interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, Buttons, StdCtrls, ExtCtrls;
type
TForm5 = class(TForm)
Panel1: TPanel; Bevel1: TBevel;
GroupBox1: TGroupBox;
Edit1: TEdit; Label1: TLabel; Label2: TLabel; Button1: TButton; BitBtn1: TBitBtn;
RadioGroup2: TRadioGroup; RadioGroup3: TRadioGroup; RadioGroup4: TRadioGroup;
RadioGroup1: TRadioGroup; Edit2: TEdit; Edit3: TEdit; Edit4: TEdit; Edit5: TEdit;
procedure Button1Click(Sender: TObject); procedure BitBtn1Click(Sender: TObject);
procedure RadioGroup1Click(Sender: TObject); procedure RadioGroup2Click(Sender: TObject);
procedure RadioGroup3Click(Sender: TObject); procedure RadioGroup4Click(Sender: TObject);
private
{ Private declarations }
public
{ Public declarations }
end;
function bsttl:integer; function bs1:integer; function bs2:integer;
function bs3:integer; function bs4:integer; function bttl:integer;
var
Form5: TForm5;
implementation
uses main,childwin,max1,max2,max3,max4,max6,max7;
{$R *.DFM}
var ttl,ttl,d1,d2,d3,d4:integer;
function bsttl:integer;
begin bsttl:=ttl; end;
function bs1:integer;
begin bs1:=d1; end;
function bs2:integer;
begin bs2:=d2; end;
function bs3:integer;
begin bs3:=d3; end;
function bs4:integer;
begin bs4:=d4; end;
function bttl:integer;
begin bttl:=ttl; end;
procedure TForm5.Button1Click(Sender: TObject);
begin
mainform.enabled:=true;ttl:=(strtoint(edit1.text)-25000); ttl:=strtoint(edit1.text);

```

```

if radiogroup1.itemindex=0 then
begin
d1:=strtoint(edit2.text);
end ;
if radiogroup1.itemindex=1 then
d1:=(ttl+25000);
if radiogroup2.itemindex=0 then
begin
d2:=strtoint(edit3.text);
end ;
if radiogroup2.itemindex=1 then
d2:=(ttl+25000);
if radiogroup3.itemindex=0 then
begin
d3:=strtoint(edit4.text);
end ;
if radiogroup3.itemindex=1 then
d3:=(ttl+25000);
if radiogroup4.itemindex=0 then
begin
d4:=strtoint(edit5.text);
end ;
if radiogroup4.itemindex=1 then
d4:=(ttl+25000);mainform.enabled:=true;close;
end;
procedure TForm5.BitBtn1Click(Sender: TObject);
begin
mainform.enabled:=true;close;
end;
procedure TForm5.RadioGroup1Click(Sender: TObject);
begin
if radiogroup1.itemindex=0 then
begin
edit2.enabled:=true; edit2.color:=clolive;
end;
if radiogroup1.itemindex=1 then
begin
edit2.enabled:=false; edit2.color:=clbtnface; edit2.text:=' ';
end;end;
procedure TForm5.RadioGroup2Click(Sender: TObject);
begin
if radiogroup2.itemindex=0 then
begin
edit3.enabled:=true; edit3.color:=clolive;
end;
if radiogroup2.itemindex=1 then
begin
edit3.enabled:=false; edit3.color:=clbtnface; edit3.text:=' ';
end;end;
procedure TForm5.RadioGroup3Click(Sender: TObject);
begin
if radiogroup3.itemindex=0 then
begin
edit4.enabled:=true; edit4.color:=clolive;
end;
if radiogroup3.itemindex=1 then

```



```

begin
edit4.enabled:=false; edit4.color:=clbtnface; edit4.text:=' ';
end;
end;

```

```

procedure TForm5.RadioGroup4Click(Sender: TObject);
begin
if radiogroup4.itemindex=0 then
begin
edit5.enabled:=true; edit5.color:=clolive;
end;
if radiogroup4.itemindex=1 then
begin
edit5.enabled:=false; edit5.color:=clbtnface; edit5.text:=' ';
end;end;end.

```

Max6.Pas

```

unit Max6;
interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, Spin, ExtCtrls, Buttons;
type
TForm6 = class(TForm) Panel1: TPanel;
Bevel1: TBevel; GroupBox1: TGroupBox;
SpinEdit1: TSpinEdit; Panel2: TPanel;
Bevel2: TBevel; Label1: TLabel; Panel3: TPanel; Label2: TLabel;
SpinEdit2: TSpinEdit; Panel4: TPanel; Panel5: TPanel;
Label3: TLabel; Label4: TLabel; SpinEdit3: TSpinEdit;
SpinEdit4: TSpinEdit; BitBtn1: TBitBtn; BitBtn2: TBitBtn;
procedure BitBtn1Click(Sender: TObject);
procedure BitBtn2Click(Sender: TObject);
private
public
end;
function pr1:integer; function pr2:integer; function pr3:integer; function pr4:integer;
var
Form6: TForm6;
implementation
uses main,max1,max2,max3,max4,max5,max7;
{$R *.DFM}
var p1,p2,p3,p4:integer;
function pr1:integer;
begin
pr1:=p1-1;
end;
function pr2:integer;
begin
pr2:=p2-2;
end;
function pr3:integer;
begin
pr3:=p3-3;
end;
function pr4:integer;
begin

```

```

pr4:=p4-4;
end;
procedure TForm6.BitBtn1Click(Sender: TObject);
begin
mainform.enabled:=true;
p1:=spinedit1.value;p2:=spinedit2.value;p3:=spinedit3.value;p4:=spinedit4.value;close;end;
procedure TForm6.BitBtn2Click(Sender: TObject);
begin
mainform.enabled:=true;close;
end;end.

```

Max 7.Pas

```

unit Max7;
interface
uses
SysUtils, WinTypes, WinProcs, Messages, Classes, Graphics, Controls,
Forms, Dialogs, StdCtrls, VBXCtrl, Gauge, ExtCtrls, Buttons;
type
TForm7 = class(TForm)
BiGauge1: TBiGauge; BiGauge2: TBiGauge; BiGauge3: TBiGauge; BiGauge4: TBiGauge;
BitBtn1: TBitBtn; Timer1: TTimer; Edit1: TEdit;
Label1: TLabel; Label2: TLabel; Label3: TLabel; Label4: TLabel;
Label5: TLabel; Label6: TLabel; Label7: TLabel; Label8: TLabel;
Label9: TLabel; Label10: TLabel; Label11: TLabel; Label12: TLabel;
Label13: TLabel; Label14: TLabel;
procedure BitBtn1Click(Sender: TObject);
procedure Timer1Timer(Sender: TObject);
private
public
end;
var
Form7: TForm7;
implementation
uses main,childwin,max1,max2,max3,max4,max5;
{$R *.DFM}
procedure TForm7.BitBtn1Click(Sender: TObject);
begin
mainform.enabled:=true;close;
end;
procedure TForm7.Timer1Timer(Sender: TObject);
begin
bigauge1.value:=(100*psn1 div j);bigauge2.value:=(100*psn2 div j);bigauge3.value:=(100*psn3 div
j);bigauge4.value:=(100*psn4 div j);
if j<2 then edit1.text:='0'
else
edit1.text:=inttostr(j);label8.caption:=inttostr(100*psn1 div j);label10.caption:=inttostr(100*psn2 div
j);label12.caption:=inttostr(100*psn3 div j);label14.caption:=inttostr(100*psn4 div j);
end;end.

```

BIO DATA



Agus Sunyoto dilahirkan di Ngawi pada tanggal 25 September 1973 dari pasangan Kusnan dan Karniasih yang kini berdomisili di Pojok, Kwadungan - Ngawi dan merupakan anak pertama dari empat bersaudara.

Riwayat pendidikan penulis adalah sebagai berikut:

Tahun 1985 : Lulus SD Negeri Pojok I

Tahun 1988 : Lulus SMP Negeri 2 Ngawi

Tahun 1991 : Lulus SMA Negeri 2 Ngawi

Tahun 1991 : Diterima di Jurusan Teknik Elektro FTI-ITS dengan
NRP.2912201640 melalui program UMPTN

Pada bulan Maret 1997 mengikuti seminar dan ujian Tugas Akhir pada Bidang Studi Elektronika dan diharapkan dapat mengikuti wisuda pada bulan April 1997.

Selama berkuliah di Jurusan Teknik Elektro FTI-ITS penulis aktif sebagai:

- Panitia dan pelaksana berbagai kegiatan kemahasiswaan
- Koordinator Laboratorium Rangkaian Listrik, Bidang Studi Elektronika FTI-ITS
- Asisten praktikum Rangkaian Listrik, Elektronika dan Elektronika Lanjutan II di Laboratorium Bidang Studi Elektronika