

BAB II

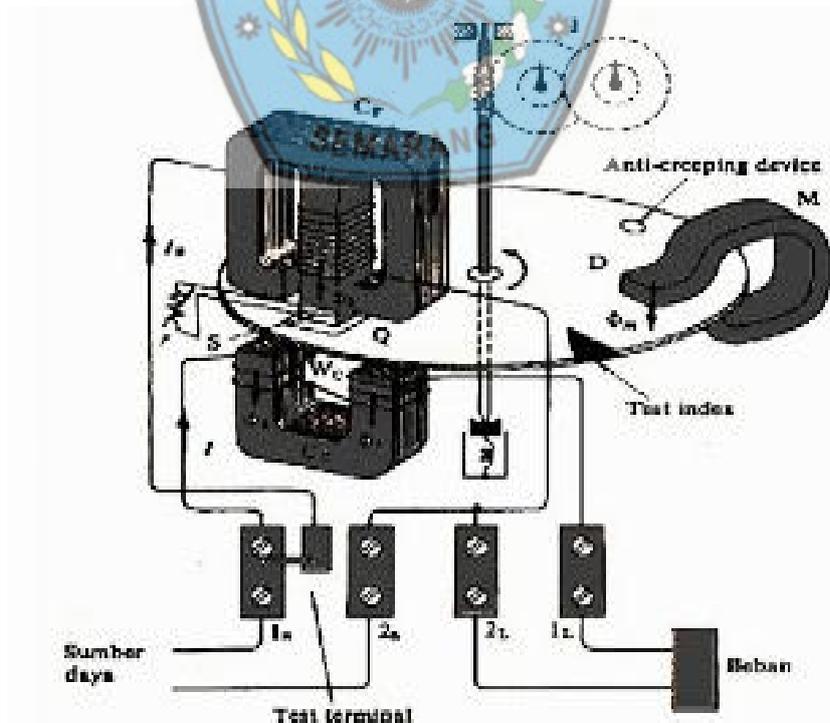
LANDASAN TEORI

2.1. Kwh-Meter Listrik Analog

Kwh-Meter listrik adalah suatu alat pencatat pemakaian daya listrik. Jumlah pemakaian daya listrik dapat dilihat dari angka yang tertera pada Kwh-Meter dengan perantara kepingan aluminium yang berputar.

2.1.1. Prinsip Kerja Kwh-Meter

Dalam alat ukur energi, kumparan-kumparan arus dan tegangan merupakan suatu belitan pada dua buah magnet. Kumparan arus akan membangkitkan *flux* magnet dengan nilai berbanding lurus dengan besar arus. Terjadinya perputaran dari piringan aluminium karena interaksi dari kedua medan magnet ini. Kemudian putaran piringan di transfer pada roda-roda pencatat. Pada transfer nilai putaran keping Aluminium ke roda-roda pencatat dilakukan kalibrasi untuk memperoleh nilai energi terukur dalam besaran Kwh (*Kilowatt hours*)



Gambar 2.1. Prinsip Kerja Kwh-Meter listrik analog dan Pencatatan (Sholeh, Bambang :1997)

Keterangan :

C_p = Inti besi kumparan tegangan

C_c = Inti besi kumparan arus

W_p = Kumparan tegangan

W_c = Kumparan arus

D = Kepingan roda Aluminium

J = Roda-roda pencatat

M = Magnet permanen sebagai pengerem keping aluminium, saat beban kosong

S = Kumparan penyesuai beda fase arus dan tegangan

2.1.2. Kelemahan Kwh-Meter Analog

Dari sisi ketelitian, ada beberapa kelemahan Kwh-Meter analog, antara lain sebagai berikut :

1. Pada saat arus beban mengalir pada kumparan, arus akan menimbulkan *flux* magnet ϕ_1 , sedangkan pada kumparan tegangan terjadi perbedaan fase antara arus dan tegangan sebesar 90° , hal ini karena kumparan tegangan bersifat induktor. Arus yang melalui kumparan tegangan akan menimbulkan *flux* magnet ϕ_2 yang berbeda fase 90° dengan ϕ_1 . Namun *flux* magnetik akan membangkitkan arus *Eddy* pada piringan yang akan menghasilkan gaya yang melawan arah putaran piringan.
2. Pada saat beban berat ϕ_1 akan bertambah besar, pertambahan ini mengakibatkan arus pusar (*eddy current*) pada keping. Aluminium juga bertambah besar, sedang arus *eddy* ini menimbulkan momen lawan pada keping Aluminium, dan akan menghambat putaran keping Aluminium. Untuk mengatasinya pada kumparan arus dipasang shunt magnetis dimana pada saat beban penuh / berat *flux* tidak sepenuhnya dapat menimbulkan momen lawan. Pada kenyataanya beda fase antara ϕ_1 dan ϕ_2 tidak bisa betul-betul 90° , karena adanya kerugian inti dan tekanan pada kumparan tegangan. Untuk mengatasi ini caranya adalah dengan memasang kumparan penyesuai fase pada inti kumparan tegangan.

2.2. Transformator

Transformator adalah suatu alat untuk memindahkan daya listrik arus bolak-balik dari satu rangkaian ke rangkaian lain secara induksi *elektromagnetik*. Bagian *Transformator* yang pertama adalah primer dan bagian yang lain adalah sekunder, maka kawatnya dibagi menjadi dua bagian menurut kegunaan *Transformator* tersebut. Arus listriknya mengalir dari kumparan primer menuju kumparan sekunder.

2.2.1. Prinsip Dasar Transformator

Apabila kumparan primer dihubungkan dengan sumber tegangan, maka akan mengalir arus bolak-balik pada kumparan tersebut. Oleh karena kumparan mempunyai inti, maka timbul *flux* magnet yang juga berubah-ubah pada intinya. Akibat adanya *flux* magnet yang berubah-ubah pada kumparan primer menyebabkan adanya GGL induksi e_p . Besarnya GGL induksi pada kumparan primer dapat dihitung dengan persamaan 2.1 berikut :

$$e_p = -N_p \frac{d\phi}{dt} \text{ Volt} \dots \dots \dots (2.1)$$

(Umanand, Bhat : 1992)

dimana : e_p = GGL Induksi pada kumparan primer.

N_p = Jumlah lilitan kumparan primer

$d\phi$ = Perubahan garis gaya magnet (*weber*)

$$1 \text{ weber} = 10^8 \text{ maxwell}$$

dt = Perubahan waktu dalam satu detik

Flux magnet yang menginduksikan GGL e_p juga dialami oleh kumparan sekunder, karena merupakan *flux* bersama (*mutual flux*). Dengan demikian *flux* tersebut menginduksikan GGL induksi e_s pada kumparan sekunder. Besarnya GGL induksi pada kumparan sekunder adalah :

$$e_s = -N_s \frac{d\phi}{dt} \text{ Volt} \dots \dots \dots (2.2)$$

(Umanand, Bhat : 1992)

dimana : N_s = Jumlah lilitan sekunder

Dari persamaan diatas diperoleh perbandingan lilitan berdasarkan perbandingan GGL induksi yaitu :

$$a = \frac{e_p}{e_s} = \frac{N_p}{N_s} \dots\dots\dots(2.3)$$

(Theraja, Theraja : 1999)

Dengan a adalah nilai perbandingan lilitan *Transformator* (*turn ratio*).

Transformator banyak digunakan untuk berbagai keperluan yang disesuaikan dengan kebutuhan. *Transformator* bisa berfungsi apabila ada tegangan, tanpa adanya tegangan jelas tidak akan berhasil melakukan perubahan tegangan yang lebih tinggi atau yang lebih rendah. Dalam *system* kerja, *Transformator* dibagi dua, yaitu :

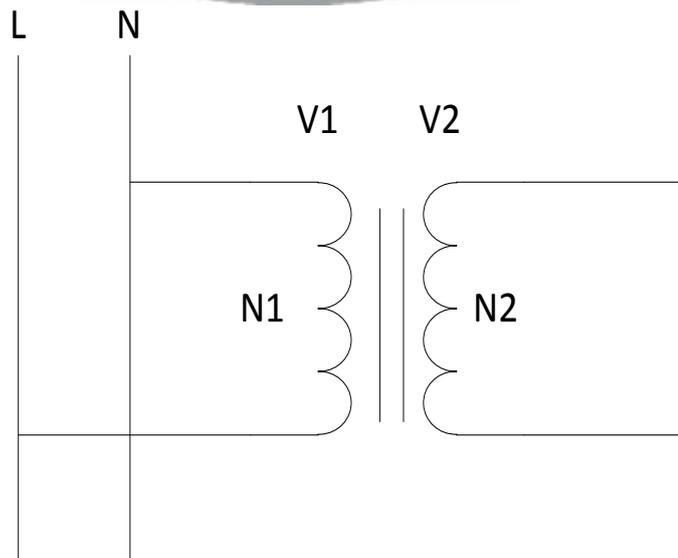
1. *Transformator* penaik tegangan (*step up*).
2. *Transformator* penurun tegangan (*Step Down*)

2.2.2. *Transformator* Tegangan (VT) sebagai Sensor Tegangan

Transformator yang digunakan sebagai sensor tegangan adalah *Transformator* penurun tegangan (*Step Down*). Dengan mengetahui N1 dan N2, membaca tegangan V2 serta menganggap transformator ideal, maka tegangan V1 adalah :

$$V1 = \frac{N1}{N2} \cdot V2 \dots\dots\dots(2.4)$$

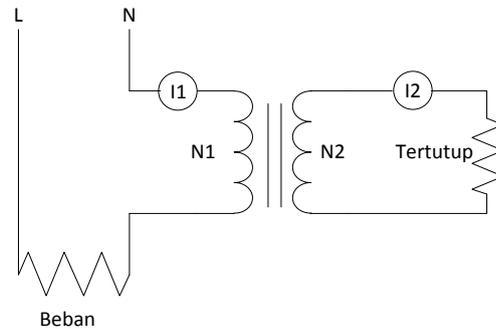
(Umanand, Bhat : 1992)



Gambar 2.2. *Transformator* Tegangan

2.2.3. Transformator Arus (CT) sebagai Sensor Arus

Transformator arus digunakan untuk mengukur arus beban suatu rangkaian. Dengan menggunakan *Transformator* arus maka arus beban yang besar dapat diukur hanya dengan menggunakan alat ukur amper meter yang tidak terlalu besar.



Gambar 2.3. Rangkaian Lengkap Sensor Arus

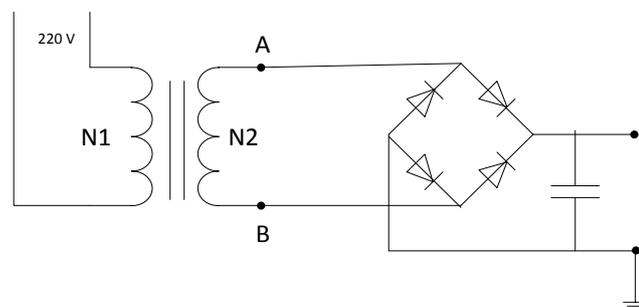
Dengan mengetahui perbandingan *Transformator* $N1/N2$ dan pembacaan ampermeter ($I2$), arus beban $I1$ dapat dihitung. Bila *Transformator* dianggap ideal, maka arus beban :

$$I1 = \frac{N1}{N2} \cdot I2 \quad \dots\dots\dots(2.5)$$

Untuk menjaga agar *flux* tidak berubah, maka perlu diperhatikan agar rangkaian sekunder selalu tertutup. Dalam keadaan rangkaian sekunder terbuka GGM $N2$ $I2$ akan sama dengan nol (karena $I2 = 0$) sedangkan GGM $N1$ $I1$ akan tetap ada, sehingga *flux* normal akan terganggu.

2.3. Rangkaian Penyesuaian Sensor Tegangan

Untuk rangkaian lengkap Penyesuaian Sensor Tegangan dan sensor arus yang digunakan, terlihat seperti gambar 2.4.



Gambar 2.4. Rangkaian Lengkap Penyesuaian Sensor Tegangan

2.4.Rangkaian Pendeteksi Beda Fase (*Zero Crossing Detector*)

Untuk mendapatkan nilai faktor daya yang diinginkan dari beban, maka perlu adanya perancangan modul pendeteksi beda fase. Beda fase diperoleh dengan cara membandingkan antara fase tegangan yang masuk dengan fase arus yang ditarik oleh beban. Tegangan dan arus hasil deteksi sensor masih berupa sinyal sinus dimana fase dari keduanya berbeda. (Hermawan : 2013)

2.4.1. Gerbang logika Ex-Or (*Exclusive Or*)

Gerbang logika Or-eksklusif disebut juga sebagai gerbang “setiap tetapi tidak semua”. Istilah Or-eksklusif sering kali disingkat sebagai Ex-Or. Simbol standard gerbang logika Ex-Or adalah seperti tampak pada gambar 2.8.



Gambar 2.5 Simbol Gerbang Logika

Tabel kebenaran untuk fungsi Ex-Or diberikan pada tabel 2.1, dari tabel tersebut terlihat bahwa tabel tersebut sama seperti tabel kebenaran gerbang Or, kecuali bila semua masukan adalah tinggi (1), gerbang Ex-Or akan membangkitkan keluaran rendah (0).

Tabel 2.1 Tabel Kebenaran Gerbang Ex-Or

Masukan		Keluaran
B	A	Ex-Or
0	0	0
0	1	1
1	0	1
1	1	0

0 = tegangan rendah; 1 = tegangan tinggi

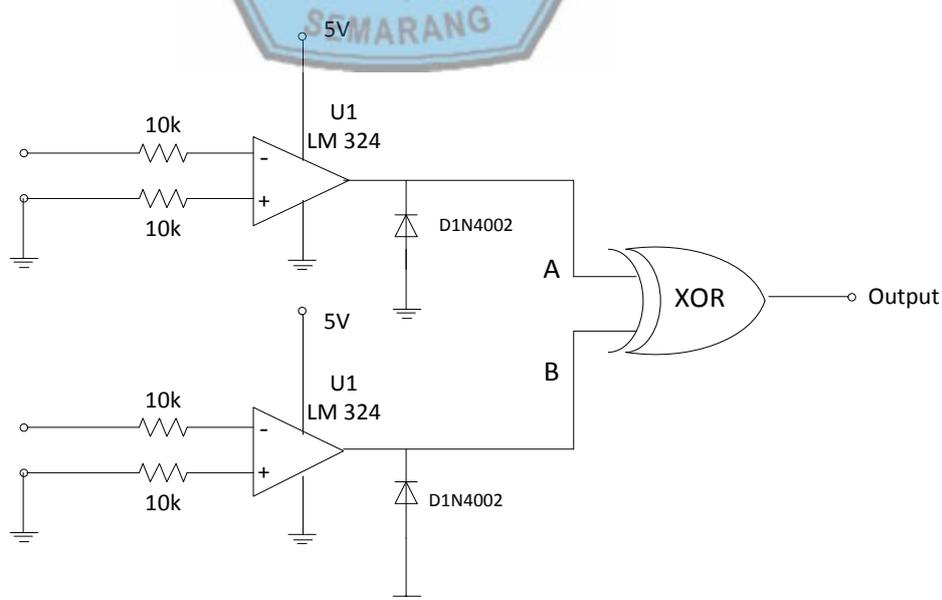
Gerbang Ex-Or hanya akan terbuka bila muncul satuan bilangan ganjil pada masukan. Baris 2 dan 3 dari tabel kebenaran mempunyai satuan bilangan ganjil,

oleh karena itu keluaran akan terbuka dengan level logika tinggi (1). Baris 1 dan 4 dari tabel kebenaran tersebut berisi satuan bilangan genap (0 dan 2), oleh karena itu gerbang Ex-Or tidak terbuka dan akan muncul logika rendah (0) pada keluaran. (<http://www.robotics-university.com/2013/01/gerbang-logika-Ex-Or-exclusive-or.html>)

2.4.2. Rangkaian Komparator

Rangkaian komparator berfungsi untuk mendeteksi perbedaan sudut fase yang mengalir ke beban. Detektor Fase dibuat menggunakan komparator dan gerbang logika Ex-Or. komparator digunakan untuk mendapatkan informasi saat nilai tegangan dan nilai arus tepat melewati titik nol. Gerbang logika Ex-Or digunakan untuk mengetahui nilai beda sudut fase. Nilai perbedaan sudut fase didapat dengan menghitung selang waktu antara tegangan naik dan tegangan turun pada keluaran gerbang logika Ex-Or. (Ardikusuma : 2011)

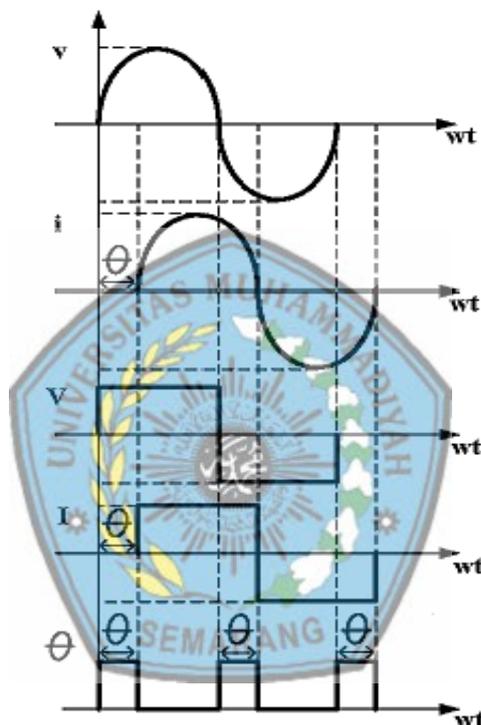
Blok komparator digunakan untuk mengubah sinyal sinus dari tegangan dan arus yang didapat menjadi sinyal kotak agar lebih mudah untuk diolah. Perancangan blok komparator menggunakan salah satu fasilitas yang disediakan oleh *Operational Amplifier (Op Amp)* type LM-324. (Hermawan : 2013)



Gambar 2.6. Gambar Rangkaian komparator dan Ex-Or

Untuk mendapatkan perbedaan waktu dari pergeseran fase antara sensor tegangan dan sensor arus yang sudah melalui komparator, maka *Output* dari IC LM 324 ini akan diumpungkan pada sebuah gerbang logika Ex-Or yang bisa mendeteksi waktu kedua sensor mengeluarkan *Output* yang berbeda. Jika kedua kaki Ex-Or mendapatkan logika yang sama, maka *Output* akan 0, tapi jika salah satu kaki mendapatkan logika rendah atau tinggi, maka *Output* akan 1.

Grafik gelombang input dan *Output* pada rangkaian diatas dapat dilihat pada gambar 2.7. (Ardikusuma : 2011)



Gambar 2.7. Gelombang Sinus, gelombang kotak dan sudut fase dalam (t)

Sudut fase dapat dihitung dengan rumus sebagai berikut :

$$\theta = \frac{\Delta t}{t_{1/2T}} \cdot 180^0 \dots\dots\dots(2.6)$$

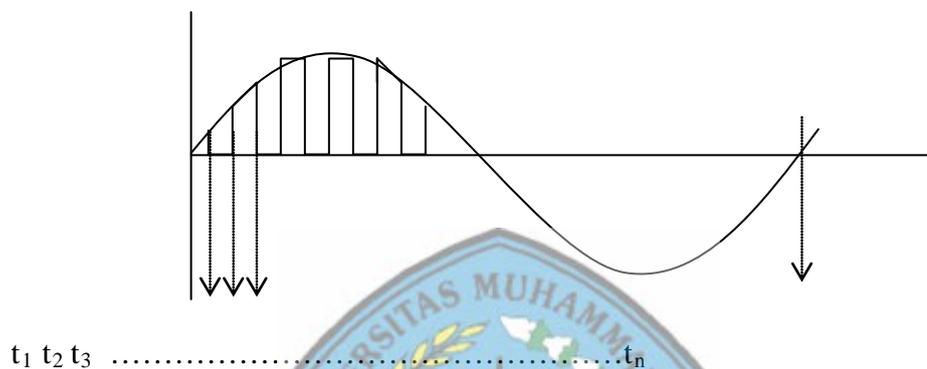
- Dimana : θ = Beda fase dalam sudut (0)
- Δt = beda fase dalam waktu (ms)
- $t_{1/2T}$ = waktu setengah periode sinyal (ms)

Nilai *cosinus* dari beda fase dalam sudut merupakan nilai faktor daya yang dicari. (Hermawan : 2013)

2.5. Proses Pencuplikan/Pencacahan Data

2.5.1. *Sample and Hold*

Sample and Hold adalah proses pengambilan data gelombang arus bolak-balik. Pengambilan data dilakukan dengan mencuplik keadaan tertentu dari kondisi gelombang, kemudian proses pengambilan dengan mencuplik tersebut diterjemahkan kedalam besaran yang dapat dibaca dan dimengerti oleh manusia sehingga dapat diolah. Sebagai ilustrasi dari proses *Sample and Hold* dapat dilihat pada gambar dibawah 2.8.



Gambar 2.8. Proses Sampling Data (Jumrianto : 2003)

2.6. Bahasa Pemrograman

Bahasa pemrograman adalah suatu bahasa yang digunakan untuk dapat berkomunikasi dengan peralatan komputer.

Dalam ilmu komputer dikenal tiga tingkatan bahasa, yaitu :

1. Bahasa tingkat tinggi

Yang termasuk kedalam bahasa tingkat tinggi diantaranya adalah : *Basic, Cobol, Pascal, Visual Basic, Delphi* dan lain-lain.

2. Bahasa tingkat menengah

Diantaranya adalah Bahasa C, *Bascom AVR*

3. Bahasa tingkat rendah

Contoh bahasa tingkat rendah adalah : *Assembly* dan Bahasa Mesin.

Bahasa pemrograman banyak digunakan untuk kebutuhan pemrograman, baik program yang berkaitan dengan bagaimana memanipulasi layar, agar menghasilkan suatu tampilan yang kita kehendaki, pengolahan data-data, sampai pada penggunaan yang lebih spesifik misalnya : mengakses sebuah *port*,

menjalankan robot, mendeteksi suhu, pengontrolan terhadap kerja suatu peralatan dan lain sebagainya.

2.6.1. *Bascom AVR*

2.6.1.1. Type Data

Setiap variable dalam *Bascom AVR* memiliki type data yang menunjukkan muatan atau besarnya memori yang terpakai olehnya. Berikut tipe data pada *Bascom AVR*.

Tabel 2.2. Tipe Data dan Ukurannya

Tipe	Data Ukuran (Byte)	Jangkauan Data
<i>Bit</i>	1/8	0 atau 1
<i>Byte</i>	1	0 s/d 255
<i>Integer</i>	2	-32768 s/d 3.767
<i>Word</i>	2	0 s/d 65535
<i>Long</i>	4	-2147483648 s/d 2147483647
<i>Single</i>	4	$1,5 \times 10^{-45}$ s/d $3,4 \times 10^{38}$
<i>Double</i>	8	5×10^{-328} s/d $1,7 \times 10^{308}$
<i>String</i>	s/d 254	

2.6.1.2. Variabel

Variabel atau peubah digunakan untuk menyimpan data sementara. Variabel diberi nama dan dideklarasikan terlebih dahulu sebelum digunakan. Aturan pemberian nama Variabel sebagai berikut:

- Harus dimulai dengan huruf (bukan angka).
- Tidak ada nama Variabel yang sama dalam sebuah program.
- Maksimum 32 karakter
- Tanpa menggunakan spasi, pemisahan bisa dilakukan dengan garis bawah.
- Tidak menggunakan karakter-karakter khusus yang digunakan sebagai Operator *Bascom AVR*

Variabel dapat dideklarasikan dengan beberapa cara :

1. Dengan pernyataan DIM

Deklarasi ini dibuat dengan perintah **DIM** (singkatan dari *dimension*) dengan aturan sebagai berikut :

Dim <NamaVariabel> As <TipeData>

Contoh :

Dim angka As Integer

Dim bilangan As byte

Jika beberapa Variabel dideklarasikan dalam satu baris, maka harus dipisah dengan tanda koma.

Contoh :

Dim angka As Integer, bilangan As byte

2. Dengan pernyataan DEFINT, DEFBIT, DEFBYTE, DEFWORD

Deklarasi dengan pernyataan tersebut secara prinsip tidak berbeda dengan "DIM", perhatikan keterangan dari masing-masing pendeklarasian tersebut :

DEFINT = untuk tipe data integer,

DEFBIT = untuk tipe data bit,

DEFBYTE = untuk tipe data byte,

DEFWORD = untuk tipe data word,

DEFLNG = untuk tipe data long,

DEFSNG = untuk tipe data singel,

DEFDBL = untuk tipe data double.

Cara pendeklarasiannya sebagai berikut :

DEFINT/DEFBIT/DEFBYTE/DEFWORD <Variabel>

Contoh :

DEFINT angka

DEFBYTE bilangan

Untuk Variabel dengan tipe data yang sama dapat dideklarasikan dengan dipisah titik koma, misal :

DEFINT bil_1 ; bil_2 ; bil_3

2.6.1.3. Konstanta

Berbeda dengan Variabel, sebuah konstanta akan bernilai tetap. Sebelum digunakan, konstanta dideklarasikan terlebih dulu dengan cara (ada dua cara) :

Dim nama_konstanta As const nilai_konstanta

Const nama_konstanta = nilai_konstanta

Contoh :

Dim pembagi as const 23

Const pembagi = 23

2.6.1.4. Penulisan Bilangan

Pada *Bascom AVR*, bilangan dapat ditulis dalam 3 bentuk :

1. Desimal ditulis biasa, contoh : 16
2. Biner diawali dengan &B, contoh : &B10001111
3. Heksadesimal diawali dengan &H, contoh : &H8F

2.6.1.5. Alias

Untuk mempermudah pemrograman, biasanya nama *register* dalam *Microcontroller* dibuatkan nama yang identik dengan *Hardware* yang dibuat, contoh :

LED_1 alias PORTC.0 ‘ nama lain dari PORTC.0 adalah LED_1

SW_1 alias PINC.1 ‘ nama lain dari PINC.1 adalah SW_1

2.6.1.6. Array atau Larik

Array atau larik merupakan sekumpulan Variabel dengan nama dan tipe yang sama, yang berbeda indeks keanggotaannya.

Cara mendeklarasikan *array* sebagai berikut :

Dim nama *array* (jumlah anggota) as tipe_data

Contoh:

Dim daku(8) as byte ‘ Variabel daku dengan tipe data byte

‘ dengan 8 anggota

Untuk mengakses *array* dengan cara :

daku(1) = 25 ‘ anggota pertama Variabel daku isinya 25

PORTC=daku(1) ‘ PORTC = nilai anggota pertama Variabel daku

2.6.1.7. Operator

Operator digunakan dalam pengolahan data pemrograman dan biasanya membutuhkan dua Variabel atau dua parameter, sedangkan Operator dituliskan di

antara kedua parameter tersebut. Operator-Operator *Bascom AVR AVR* ditunjukkan pada Tabel 2.3, Tabel 2.4 dan Tabel 2.5.

Tabel 2.3. Operator Aritmatik

Operator	Keterangan	Contoh
+	Operasi penjumlahan	A + B
-	Operasi pengurangan	A – B
*	Operasi perkalian	A * B
/	Operasi pembagian	A / B
%	Operasi sisa pembagian	A % B

Tabel 2.4. Operator Relasional

Operator	Keterangan	Contoh
=	Sama dengan	A = B
<>	Tidak sama dengan	A <> B
>	Lebih besar dari	A > B
<	Lebih kecil dari	A < B
>=	Lebih besar atau sama dengan	A >= B
<=	Lebih kecil atau sama dengan	A <= B

Tabel 2.5. Operator Logika

Operator	Keterangan	Contoh
AND	Operasi AND	&B110 And &B101 = &B100
OR	Operasi OR	&B11001 Or &B10111 = &B11111
NOT	Operasi NOT	NOT &HFF = &HO
EX-OR	Operasi EX-OR	&B1001 Ex-Or &B0111 = &B1110

2.6.1.8. Operasi Bersyarat

A. IF – THEN

Sebuah atau serangkaian instruksi akan dikerjakan jika memenuhi syarat-syarat atau kondisi tertentu. Cara penulisannya sebagai berikut :

If <kondisi> Then <perintah> ‘ 1 baris perintah

If <kondisi> Then ‘ lebih dari 1 perintah

<perintah 1>

<perintah 2>

...

End If

B. IF - THEN – ELSE

Versi lengkap dari sebuah atau serangkaian instruksi akan dikerjakan jika memenuhi syarat-syarat atau kondisi tertentu, jika tidak dipenuhi maka instruksi atau serangkaian instruksi lainnya-lah yang akan dikerjakan. Cara penulisannya sebagai berikut :

If <kondisi> Then

<perintah 1>

...

Else

<perintah 2>

...

End If

C. IF - THEN - ELSEIF

Sama seperti IF-THEN-ELSE, hanya jika kondisi tidak dipenuhi masih dilakukan pengujian apakah suatu kondisi memenuhi syarat lainnya. Cara atau sintaks (*syntax*) penulisannya sebagai berikut :

If <kondisi 1> Then

<perintah 1>

...

Elseif <kondisi 2> Then

<perintah 2>

...

End If

D. SELECT - CASE

Cocok digunakan untuk menangani pengujian kondisi yang jumlahnya cukup banyak. Cara penulisannya :

Select case <Variabel>

Case 1: <perintah 1>



Case 2: <perintah 2>

...

End Select

2.6.1.9. Pengulangan Operasi

A. FOR - NEXT

Perintah ini digunakan untuk melaksanakan perintah secara berulang sesuai dengan jumlah yang ditentukan. Sintaks penulisannya :

For <var> = <nil_awal> To <nil_akhir><step angka>

<perintah>

Next [<var>]

B. DO - LOOP

Pernyataan ini untuk melakukan pengulangan terus menerus tanpa henti (pengulangan tak berhingga) selama *Microcontroller*-nya masih mendapatkan detak dan/atau catu daya. Cara penulisannya :

Do

<pernyataan>

...

Loop

Jika pengulangan dibatasi oleh suatu kondisi maka caranya ditunjukkan berikut ini, artinya pengulangan terus dilakukan sehingga suatu kondisi terpenuhi atau melakukan pengulangan selama kondisinya salah :

Do

<pernyataan>

...

Loop Until <kondisi>

C. WHILE - WEND

Berbeda dengan DO-LOOP, instruksi ini digunakan untuk melakukan pengulangan selama kondisinya benar, cara penulisannya :

While <kondisi>

<perintah>

...

Wend

2.6.1.10. Lompatan Proses

A. GOSUB <nama_subrutin>

Perintah ini akan melakukan lompatan sebuah subrutin, kemudian kembali lagi setelah subrutin perintah tersebut selesai dikerjakan. Rutin yang dibuat harus diakhiri dengan instruksi **RETURN**. Contoh :

Print "We will start execution here"

Gosub Routine

Print "Back from Routine"

End

Routine:

Print "This will be executed"

Return

B. GOTO <label>

Perintah ini untuk melakukan lompatan ke label kemudian melakukan serangkaian instruksi tanpa harus kembali lagi, sehingga tidak perlu **RETURN**.

Contoh :

Dim A As Byte

Start: 'sebuah label diakhiri dengan :

A = A + 1 'naikkan Variabel A

If A < 10 Then 'apakah lebih kecil 10?

Goto Start 'ya, lakukan lagi

End If 'akhir IF

Print "Ready" 'ok

C. EXIT

Untuk keluar secara langsung dari perulangan DO-LOOP, FOR-NEXT, WHILE-WEND. Cara penulisannya sebagai berikut :

EXIT FOR (keluar dari For-Next)

EXIT DO (keluar dari Do-Loop)

EXIT WHILE (keluar dari While-Wend)

EXIT SUB (keluar dari Sub-Endsub)

EXIT FUNCTION (keluar dari suatu fungsi)

(Agfianto : 2010)

2.7. Microcontroller AVR ATmega

Adalah *Microcontroller* 16 dan 32 bit *Microcontroller* daya rendah, integrasi tinggi dan mudah digunakan Buatan *Atmel Corporation*. *Microcontroller* keluarga AVR ATmega ini menjadi *Microcontroller* yang paling populer dikalangan pengembang *systemMicrocontroller* dan penghobi *Microcontroller*. Hal ini dikarenakan kelebihan-kelebihan yang dimiliki *Microcontroller* ini misalnya fitur yang lengkap, konsumsi daya yang rendah, penggunaannya mudah, banyak tersedia di pasaran dan yang paling penting adalah harga yang sangat terjangkau.

Dari segi pemrograman, *Microcontroller* AVR ATmega Instruksi-intruksi pemrograman sudah mendukung *system RISC (Reduction Instruction Set Computing)* sehingga kebanyakan instruksi-intruksi pemrograman *Microcontroller* AVR ATmega dapat dieksekusi atau dijalankan oleh satu siklus *pulsaclock*. Dengan kemampuan mengeksekusi instruksi setiap siklus *clock* maka untuk *Microcontroller* AVR ATmega yang menggunakan *clock* 16 MHz kecepatan eksekusi dapat mencapai 16 MIPS (*Milion Instruction Per Second*) atau dapat menjalankan 16 Juta Instruksi dalam setiap detik.

Catu daya tegangan yang dibutuhkan oleh *Microcontroller* AVR ATmega adalah berkisar antara 4,5 Volt sampai dengan 5,5 volt dengan arus yang diserap hanya berkisar antara 100mA sampai dengan 500mA maka konsumsi daya *Microcontroller* ini terbilang sangat rendah karena hanya dibawah 1 Watt.

Fitur-Fitur(Kelengkapan) *Microcontroller* AVR ATmega :

1. *Microcontroller* 8 Bit dengan konsumsi daya rendah tapi memiliki unjuk kerja yang tinggi.
2. Arsitektur RISC (*Reduced Instruction Set Computing*/Komputasi dengan instruksi yang dikurangi/direduksi) yang diperluas.
3. Memiliki 131 Instruksi berdaya guna tinggi dan banyak dapat dieksekusi dalam setiap 1 *pulsaclock*
4. Memiliki 32 x 8 *Register* penggunaan umum
5. Beroperasi sepenuhnya dalam kondisi *static*

6. Kecepatan operasi 16 MIPS pada *Clock* 16 MHz
7. Pengali 2 terpadu dalam setiap kemasan IC
8. Memori program dan data yang tidak mudah hilang
9. Memiliki 16 KB *In System Self Programmable Flash* memori dengan daya tahan 1000 kali Baca atau tulis
10. Pilihan Seksi *Boot Code* dengan *Lock Bit* Mandiri
11. Memiliki 512 Byte EEPROM (*Electrical Erasable And Programmable Read Only Memori*) dengan daya tahan 100.000 kali Baca/Tulis
12. 1KB SRAM Internal
13. Pengunci pemrograman untuk melindungi program aplikasi
14. JTAG *interface* (penghubung)
15. Memiliki 2 buah 8 bit *timer/counter* dengan pengali, pembagi dan mode perbandingan yang terpisah
16. Memiliki 1 buah 16 Bit *timer/counter* dengan pengali, pembagi dan mode perbandingan yang terpisah.
17. RTC (*Real Time Counter*) dengan *Osilator* terpisah
18. Empat buah saluran PWM (*Pulse Width Modulation*)
19. 8 Saluran ADC (*Analog to Digital Converter*)
20. *Byte Oriented Two Wire Serial Interface*
21. Saluran komunikasi *Serial USART* yang dapat diprogram
22. *Master/Slave SPI Serial interface*
23. *Programmable Watchdog Timer with Separate On-chip Oscillator*
24. *On-chip Analog Comparator*
25. 32 (4 x 8 *Port*) Saluran Input Output
26. Bentuk kemasan 40 pin PDIP, 44 pin TQFP atau 44 pad MLF
27. Tegangan kerja 4,5 Volt – 5.5 Volt
28. Kecepatan *clock* maksimum 16 MHz

(Sumber: <http://www.teknikMicrocontroller.com/Microcontroller-avr-ATMega/>)

2.7.1. AVR ATmega 32

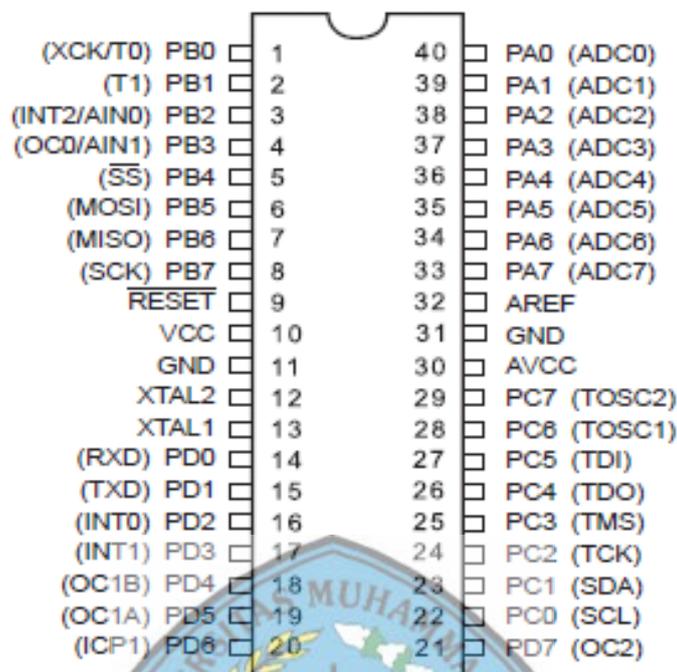
AVR (*Alf and Vegard's Risc Processor*) merupakan salah satu jenis *Microcontroller* yang di dalamnya terdapat berbagai macam fungsi. AVR memiliki keunggulan dibandingkan dengan *Microcontroller* jenis lain, keunggulannya yaitu AVR memiliki kecepatan eksekusi program yang lebih cepat karena sebagian besar instruksi dieksekusi dalam 1 siklus *clock*, lebih cepat bila dibandingkan dengan *Microcontroller* jenis MCS51 yang memiliki arsitektur CISC (*Complex Instruction Set Compute*) dimana mikrokontoller MCS51 membutuhkan 12 siklus *clock* untuk mengeksekusi 1 instruksi. Selain itu kelebihan *Microcontroller* AVR memiliki POS (*Power On Reset*), yaitu tidak perlu adanya tombol reset dari luar karena cukup hanya dengan mematikan supply, maka secara otomatis AVR akan melakukan reset. Untuk beberapa jenis AVR terdapat beberapa fungsi khusus seperti ADC, EEPROM sekitar 128 bytes sampai dengan 512 bytes.

Dalam hal ini yang digunakan adalah *Microcontroller* AVR tipe *ATMega 32* standar. Perbedaannya dengan AVR tipe *ATMega 32L* terletak pada besarnya tegangan kerja yang dibutuhkan. Untuk *ATMega 32L* tegangan kerjanya antara 2,7V - 5,5V sedangkan untuk *ATMega 32* hanya dapat bekerja pada tegangan 4,5V - 5,5V.

ATMega 32 merupakan *Microcontroller* AVR 8 bit berkemampuan tinggi dengan daya yang rendah, dan memiliki 32 x 8 *general purpose working register*. Kecepatan eksekusi program yang dimiliki *ATMega 32* lebih cepat karena sebagian besar instruksi dieksekusi dalam 1 *clock* dengan arsitektur RISC hampir mencapai 16 MIPS pada frekuensi 16 MHz. Memori data dan program yang tidak mudah hilang (*Nonvolatile Program and Data Memories*) dengan Pemrograman *Flash* memiliki kapasitas 8K Bytes, dan memiliki daya tahan 1000 siklus tulis/hapus program. Fasilitas *timer/counter* yang ada pada *Microcontroller* ini terdiri dari dua buah *Timer/counter* 8 bit dan satu buah *Timer/counter* 16 bit. Fitur-fitur tersebut adalah fitur-fitur yang digunakan dalam proyek akhir ini. Fitur-fitur lainnya yang disediakan *ATMega 32* adalah adanya 4 kanal PWM, 6 kanal ADC 10 bit, pemrograman serial USART, *On-chip Analog Comparator*, dan *interrupt*. (Fauzi, Madhawirawan : 2013)

Arsitektur AVR ini menggabungkan perintah secara efektif dengan 32

Konfigurasi dari kaki-kaki IC AVR ATMega 32 dapat dilihat pada gambar 2.10 dibawah :



Gambar 2.10 Konfigurasi Pin ATmega 32 (Datasheet AVR ATmega32)

Secara fungsional konfigurasi pin ATmega 32 adalah sebagai berikut:

- a. VCC
 - Tegangan sumber
- b. GND (*Ground*)
 - *Ground*
- c. Port A (PA7 – PA0)

Port A adalah 8-bit port I/O yang bersifat *bi-directional* dan setiap pin memiliki internal *pull-up* resistor. Output buffer port A dapat mengalirkan arus sebesar 20 mA. Ketika port A digunakan sebagai input dan di *pull-up* secara langsung, maka port A akan mengeluarkan arus jika internal *pull-up* resistor diaktifkan. Pin-pin dari port A memiliki fungsi khusus yaitu dapat berfungsi sebagai channel ADC (*Analog to Digital Converter*) sebesar 10 bit. Fungsi-fungsi khusus pin-pin port A dapat ditabelkan seperti yang tertera pada table 2.6.

Tabel 2.6 Fungsi khusus port A

<i>Port</i>	<i>Alternate Function</i>
PA7	ADC7 (ADCinput channel 7)
PA6	ADC6 (ADCinput channel 6)
PA5	ADC5 (ADCinput channel 5)
PA4	ADC4 (ADCinput channel 4)
PA3	ADC3 (ADCinput channel 3)
PA2	ADC2 (ADCinput channel 2)
PA1	ADC1 (ADCinput channel 1)
PA0	ADC0 (ADCinput channel 0)

d. *Port B (PB7 – PB0)*

Port B adalah 8-bit *port I/O* yang bersifat *bi-directional* dan setiap pin mengandung internal *pull-up* resistor. *Output* buffer *port B* dapat mengalirkan arus sebesar 20 mA. Ketika *port B* digunakan sebagai input dan di *pull-down* secara *external*, *port B* akan mengalirkan arus jika internal *pull-up* resistor diaktifkan.

Pin-pin *port B* memiliki fungsi-fungsi khusus, diantaranya :

SCK *port B*, bit 7

Input pin *clock* untuk *up/downloading* memory.

MISO *port B*, bit 6

Pin *Output* data untuk *uploading* memory.

MOSI *port B*, bit 5

Pin input data untuk *downloading* memory.

Fungsi-fungsi khusus pin-pin *port B* dapat ditabelkan seperti pada tabel 2.7

Tabel 2.7 Fungsi khususport B

<i>Port</i>	<i>Alternate Function</i>
PB7	SCK (<i>SPI</i> Bus Serial Clock)
PB6	MISO (<i>SPI</i> Bus Master Input/Slave Output)
PB6	MOSI (<i>SPI</i> Bus Master Output/Slave Input)
PB5	SS (<i>SPI</i> Slave Select Input)
PB3	AIN1 (Analog Comparator Negative Input) OCO (Timer/counter0 Output Compare Match Output)
PB2	AIN0 (Analog Comparator Positive Input) INT2 (External Interrupt 2 Input)
PB1	T1 (Timer/counter1 External Counter Input)
PB0	T0 (Timer/counter External Counter Input) XCK (USART)

	<i>External Clock Input/Output</i>
--	------------------------------------

e. *Port C (PC7 – PC0)*

Port C adalah 8-bit *port I/O* yang berfungsi *bi-directional* dan setiap pin memiliki internal *pull-up* resistor. *Output buffer port C* dapat mengalirkan arus sebesar 20 mA. Ketika *port C* digunakan sebagai input dan di *pull-down* secara langsung, maka *port C* akan mengeluarkan arus jika internal *pull-up* resistor diaktifkan. Fungsi-fungsi khusus pin-pin *port C* dapat ditabelkan seperti yang tertera pada tabel dibawah ini.

Tabel 2.8. Fungsi khususport C

<i>Port</i>	<i>Alternate Function</i>
PC7	TOSC2 (<i>Timer Oscillator Pin 2</i>)
PC6	TOSC1 (<i>Timer Oscillator Pin 1</i>)
PC6	TD1 (<i>JTAG Test Data In</i>)
PC5	TD0 (<i>JTAG Test Data Out</i>)
PC3	TMS (<i>JTAG Test Mode Select</i>)
PC2	TCK (<i>JTAG Test Clock</i>)
PC1	SDA (<i>Two-wire Serial Bus Data Input/Output Line</i>)
PC0	SCL (<i>Two-wire Serial Bus Clock Line</i>)

f. *Port D (PD7 – PD0)*

Port D adalah 8-bit *port I/O* yang berfungsi *bi-directional* dan setiap pin memiliki internal *pull-up* resistor. *Output buffer port D* dapat mengalirkan arus sebesar 20 mA. Ketika *port D* digunakan sebagai input dan di *pull-down* secara langsung, maka *port D* akan mengeluarkan arus jika internal *pull-up* resistor diaktifkan. Fungsi-fungsi khusus pin-pin *port D* dapat ditabelkan seperti yang tertera pada tabel dibawah ini.

Tabel 2.9 Fungsi khususport D

<i>Port</i>	<i>Alternate Function</i>
PD7	OC2 (<i>Timer / Counter2 Output Compare Match Output</i>)
PD6	ICP1 (<i>Timer/counter1 Input Capture Pin</i>)
PD6	OCIB (<i>Timer/counter1 Output Compare B Match Output</i>)
PD5	TD0 (<i>JTAG Test Data Out</i>)
PD3	INT1 (<i>External Interrupt 1 Input</i>)
PD2	INT0 (<i>External Interrupt 0 Input</i>)
PD1	TXD (<i>USART Output Pin</i>)
PD0	RXD (<i>USART Input Pin</i>)

Sumber : <http://fmpunya.blogspot.com/2012/06/dasar-teori-Microcontroller-ATMega-32.html>

2.7.1.3. Prinsip Kerja *ATMega 32*

Prinsip kerja sebuah *Microcontroller* dapat dijelaskan sebagai berikut:

1. Berdasarkan data yang ada pada *register* program *counter*. *Microcontroller* mengambil data dari ROM dengan alamat sebagaimana ditunjukkan dalam program *counter*. Selanjutnya program *counter* ditambah nilainya dengan 1 secara otomatis. Data yang diambil tersebut merupakan urutan instruksi program pengendali *Microcontroller* yang sebelumnya telah dituliskan oleh pembuatnya.
2. Instruksi tersebut diolah dan dijalankan. Proses pengerjaan bergantung pada jenis instruksi, bisa membaca, mengubah nilai-nilai dalam *register*, RAM, isi *port* atau melakukan pembacaan dan dilanjutkan dengan pengubahan data.
3. Program *counter* telah berubah nilainya (baik karena penambahan secara otomatis sebagaimana dijelaskan pada langkah 1 di atas atau karena pengubahan data pada langkah 2). Selanjutnya yang dilakukan *Microcontroller* adalah mengulang kembali siklus ini pada langkah 1. Demikian seterusnya hingga catu daya dimatikan.

Berdasarkan uraian di atas dapat disimpulkan bahwa pada dasarnya unjuk kerja *Microcontroller* sangatlah tergantung pada urutan instruksi yang dijalankannya, yaitu program yang ditulis dalam ROM.

Dengan membuat program yang bermacam-macam, tentunya *Microcontroller* dapat mengerjakan tugas yang bermacam-macam pula. Fasilitas-fasilitas yang ada misalnya *timer/counter*, *port I/O*, *serial port*, *Analog to Digital Converter (ADC)* dapat dimanfaatkan oleh *Programmer* untuk menghasilkan kinerja yang dikehendaki. Sebagai contoh *ADC* digunakan oleh *Microcontroller* sebagai alat ukur digital untuk mengukur tegangan sinyal masukan, selanjutnya hasil pembacaan *ADC* diolah untuk kemudian dikirimkan ke sebuah *display* yang terhubung pada *port I/O* guna menampilkan hasil pembacaan yang telah diolah. Proses pengendalian *ADC*, pemberian sinyal-sinyal yang tepat pada *display*, kesemuanya dikerjakan secara berurutan pada program yang ditulis dalam ROM. Penulisan program *Microcontroller* pada umumnya menggunakan bahasa

assembly untuk *Microcontroller* yang bersangkutan (setiap jenis *Microcontroller* memiliki instruksi bahasa *assembly* yang berbeda-beda). Dengan bantuan sebuah perangkat komputer (PC), bahasa *assembly* tersebut diubah menjadi bahasa mesin *Microcontroller* dan selanjutnya disalin ke dalam ROM dari *Microcontroller*.

Berikut ini adalah fitur khusus dari *ATMega 32* :

1. 32 K byte *ISP flash program memory*
2. 2 K byte SRAM
3. 1 K byte EEPROM
4. Frekuensi osilator max. 16 MHz
5. 32 pin *Input/ Output*
6. 8 channel 10 bit *ADC, analog comparator*
7. Satu 16 bit *timer/counter* dan dua 8 bit *timer/counter*
8. *Watchdog timer, RTC, 4 channel PWM, master/ slave SPI, TWI*
9. *Programmable USART*
10. *Package 40 PDIP*

(<http://samirnganjuk.wordpress.com/2011/05/11/ATMega-32/>)

2.8. LCD (*Liquid Crystal Display*)

LCD (Liquid Crystal Display) adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama. *LCD (Liquid Crystal Display)* sudah digunakan di berbagai bidang, misalnya dalam alat-alat elektronik, seperti kalkulator ataupun layar komputer. Pada *LCD (Liquid Crystal Display)* berwarna semacam monitor, terdapat banyak sekali titik cahaya (*pixel*) yang terdiri dari satu buah kristal cair sebagai suatu titik cahaya. Walaupun disebut sebagai titik cahaya, namun kristal cair ini tidak memancarkan cahaya sendiri. (Fauzi, Madhawirawan : 2013)

LCD (Liquid Crystal Display) yang penulis gunakan dalam Perangkat ini dapat dilihat seperti gambar 2.11.



**Gambar 2.11.LCD (Liquid Crystal Display)20 Karakter 4 Baris ERM2004SYG-2
(Datasheet LCD (Liquid Crystal Display) ERM2004-2)**

Tabel 2.10 Spesifikasi Teknis : ERM2004-2

<i>Gross Weight (kg)</i>	0.1000
<i>Manufacturer</i>	EastRising
<i>Series Number</i>	ERM2004-2
<i>Part Number(Order Number)</i>	ERM2004SYG-2
<i>Connection</i>	Pin Header
<i>Display Format</i>	20x4 Character
<i>Sunlight Readable</i>	Yes
<i>Touch Panel Optional</i>	No
<i>Outline Dimension</i>	98.00(W)x60.0(H)x14.0(T)mm
<i>Visual Area</i>	76.00x25.20mm
<i>Active Area</i>	70.40(W)x20.80(H)mm
<i>Character Size</i>	2.95x4.75mm
<i>Diagonal Size</i>	No
<i>Dot (Pixel) Size</i>	0.55x0.55mm
<i>Dot (Pixel) Pitch</i>	0.60x0.60mm
<i>IC Package</i>	COB
<i>IC or Equivalent</i>	AIP31066 , HD44780, KS0066 , SPLC780 , ST7066
<i>Interface</i>	6800 4-bit Parallel , 6800 8-bit Parallel
<i>Display Type</i>	STN-LCD (Liquid Crystal Display)Yellow Green
<i>Response Time(Typ)</i>	No
<i>Contrast Ratio(Typ)</i>	No
<i>Colors</i>	No
<i>Viewing Direction</i>	6:00
<i>Viewing Angle Range</i>	No
<i>Appearance</i>	Black on Yellow Green

<i>Brightness(Typ)</i>	<i>No</i>
<i>Backlight Color</i>	<i>White Color</i>
<i>Backlight Current (Typ)</i>	<i>75mA</i>
<i>Power Supply(Typ)</i>	<i>3.3V, 5V</i>
<i>Supply Current for LCM(Max)</i>	<i>2000uA</i>
<i>Operating Temp</i>	<i>-20 °C~70 °C</i>
<i>Storage Temperature</i>	<i>-30 °C~80 °C</i>

http://www.buydisplay.com/download/manual/ERM2004-2_Series_Datasheet.pdf

2.9. Daya Listrik

Daya dalam rangkaian DC sama dengan perkalian antara arus dan tegangan. Daya dalam rangkaian AC pada setiap saat sama dengan perkalian dari harga daya rata-rata dalam satu periode sama dengan perkalian antara arus dan tegangan efektif. Tetapi jika ada reaktansi dalam rangkaian, arus dan tegangan tidak sefase selama siklusnya, seperti halnya arus bernilai negatif sementara tegangan bernilai positif. Hal ini menghasilkan besarnya daya kurang dari perkalian I dan V.

Perkalian arus dan tegangan efektif dalam rangkaian AC dinyatakan dalam *voltampere* (VA) atau *kilovoltampere* (KVA). Satu KVA sama dengan 1.000 VA. Daya yang berguna atau daya nyata diukur dalam Watt dan diperoleh jika *voltampere* dari rangkaian dikalikan dengan faktor yang disebut dengan faktor daya. Maka dalam rangkaian AC satu Fase adalah :

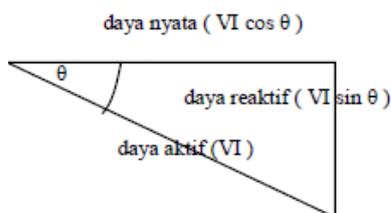
$$P \text{ (dalam watt)} = V \cdot I \cdot \text{Faktor Daya} \dots \dots \dots (2.7)$$

$$\text{Faktor Daya (Cos phi)} = \frac{P}{V \cdot I} \dots \dots \dots (2.8)$$

Oleh karena daya adalah V.I dikalikan dengan faktor daya, maka faktor daya suatu rangkaian AC sama dengan *cosinus* dari sudut Fase. Hubungan antara daya dalam watt (P), *voltampere* (VA) dan *voltampere reaktif* (VAR) dapat dinyatakan dengan segitiga seperti yang ditunjukkan dalam Gambar 2.12. sudut θ adalah sudut Fase rangkaian. Alas segitiga menyatakan daya nyata (VA), tingginya menyatakan daya reaktif (VAR), dan *hipotunosa* menyatakan daya aktif (W).

Harga faktor daya tergantung dari beda Fase antara arus dan tegangan. *Capasitor* daya AC sebagai kompensator yang dihubungkan jaringan maka akan mengakibatkan arus beban mendahului 90^0 , $I_c = I_m \sin (\omega t+90^0)$. Sehingga akan mengakibatkan arus beban menjadi sefase dengan tegangan.

Dimana arus beban yang tertinggal 90^0 akan terkompensasi arus *capasitor* mendahului sebesar 90^0 , $I_b = I_b \sin (\omega t-90^0+90^0) = I_b \sin t$. Hal tersebut ditunjukkan pada Gambar 2.12.



Gambar 2.12. Hubungan Antara Daya, Voltampere dan Voltampere Reaktif

Oleh karena *Voltampere* sama dengan $V.I$ daya nyatanya adalah $V.I \cos \phi$, dan *voltampere* reaktifnya $V.I \sin \phi$. Juga terjadi hubungan sebagai berikut

$$\text{Daya aktif} = \sqrt{(\text{daya nyata})^2 + \text{daya reaktif}^2} \dots \dots \dots (2.9)$$

(Ardikusuma : 2011)

Watt adalah satuan dari daya (*power*). Wattmeter adalah alat ukur untuk mengukur daya yang terdapat dalam suatu komponen elektronik. Salah satu fungsi mengetahui daya pada suatu rangkaian elektrik adalah hubungannya dengan efisiensi dan hemat energi. Misalnya dalam membuat sebuah alat elektronik harus benar-benar memperhitungkan efisiensi daya dari *input* ke *Output*. Hal ini bila tidak diperhatikan maka akan terjadi kerugian daya yang besar. Maka itulah pentingnya perlu mengetahui daya yang terjadi pada rangkaian elektrik. Daya listrik dalam pengertiannya dapat dikelompokkan dalam dua kelompok sesuai dengan catu tenaga listriknya, yaitu daya listrik DC dan daya listrik AC.

Daya listrik DC dirumuskan sebagai berikut :

$$P = V . I \dots \dots \dots (2.10)$$

- Dimana : P = Daya / Power (Watt)
- V = Tegangan (Volt)
- I = Arus (Ampere)

Daya listrik AC ada dua macam yaitu daya untuk satu Fase dan daya untuk tiga Fase. Pada *system* satu Fase dirumuskan sebagai berikut :

$$P = V \cdot I \cdot \cos \phi \dots \dots \dots (2.11)$$

Dimana:

V = Tegangan kerja (Volt)

I = Arus yang mengalir ke beban (Ampere)

$\cos \phi$ = faktor daya

Pada *system* tiga Fase dirumuskan sebagai berikut :

$$P = 3 V \cdot I \cdot \cos \phi \dots \dots \dots (2.12)$$

Dimana:

V = Tegangan antar Fase (Volt)

I = Arus yang mengalir ke beban (Ampere)

$\cos \phi$ = faktor daya

(Kuswanto : 2010)

2.9.1. Wattmeter

Wattmeter adalah alat ukur untuk mengukur daya yang terdapat dalam suatu komponen elektronik. Wattmeter ini mengukur daya listrik pada beban yang sedang beroperasi dalam suatu *system* kelistrikan dengan beberapa kondisi beban seperti beban DC, beban AC satu Faseserta beban AC tiga Fase. Wattmeter merupakan *instrument* pengukur daya listrik yang pembacaannya dalam satuan watt dimana merupakan kombinasi dari voltmeter dan amperemeter.

Dalam pengoperasiannya harus memperhatikan petunjuk yang ada pada *manual book* atau tabel yang tertera. Demikian juga dalam hal pembacaan data harus mengacu pada *manual book* yang ada. Pengukuran daya listrik secara langsung adalah dengan menggunakan wattmeter, ada beberapa jenis wattmeter antara lain wattmeter *elektrodinamik*, wattmeter induksi, wattmeter *elektrostatik* dan sebagainya.

Pada wattmeter *elektrodinamik* cukup *familiar* dalam desain dan konstruksi *elektrodinamometer* tipe ammeter dan voltmeter analog. Kedua koilnya dihubungkan dengan sirkuit yang berbeda dalam pengukuran *power*. Untuk prinsip kerja wattmeter induksi sama dengan prinsip kerja amperemeter dan

voltmeter induksi. Perbedaan dengan wattmeter jenis *dinamometer* adalah wattmeter induksi hanya dapat dipakai dengan suplai listrik bolak-balik sedangkan wattmeter jenis *dinamometer* dapat dipakai baik dengan suplai listrik bolak-balik atau searah.

Pengukuran daya arus searah dapat dilakukan dengan alat ukur wattmeter. Didalam *instrument* ini terdapat dua macam sensor yaitu sensor arus dan sensor Tegangan. (Kuswanto : 2010)

2.9.2. Daya Kompleks

Daya rata-rata sebenarnya adalah daya yang dipakai oleh komponen pasif resistor yang merupakan daya yang terpakai atau terserap. Jika dalam penerapan nyata, daya nyata inilah yang ditagihkan kepada pelanggan dari PLN. Daya rata-rata memiliki simbol P dan satuan Watt. Secara matematis daya rata-rata atau daya aktif adalah hasil perkalian antara tegangan efektif dan arus efektif dan koefisien faktor dayanya.

$$P = V_{eff} \cdot I_{eff} \cdot \cos \phi \dots\dots\dots(2.13)$$

Daya reaktif adalah daya yang muncul karena adanya komponen pasif selain resistor yang terpasang, bisa kapasitor maupun induktor. Daya ini seminimal mungkin dihindari atau setidaknya diperkecil karena daya reaktif merupakan daya rugi-rugi. Daya reaktif disimbolkan Q dan memiliki satuan Volt Amper Reaktif (VAR) secara matematis daya reaktif adalah hasil perkalian antara tegangan efektif, arus efektif dan sin phi.

$$Q = V_{eff} \cdot I_{eff} \cdot \sin \phi \dots\dots\dots(2.14)$$

Daya tampak atau nyata merupakan daya yang sebenarnya disuplai oleh PLN kepada pelanggan. Daya ini merupakan *resultan* daya antara daya rata-rata dan daya reaktif. Simbol dari daya tampak adalah S dan memiliki satuan Volt Amper (VA). Secara matematis, daya tampak atau daya nyata adalah hasil perkalian antara tegangan efektif dan arus efektifnya.

$$S = V_{eff} \cdot I_{eff} \dots\dots\dots(2.15)$$

Daya tampak atau daya nyata biasa juga disebut dengan daya kompleks, yaitu gabungan antara daya rata-rata dan daya reaktif. Daya kompleks memiliki dua buah bagian, yaitu bagian nyata dan nilai imajiner sebagai berikut :

$$S = P + jQ = V_{eff} \cdot I_{eff} \cdot \cos \phi + j V_{eff} \cdot I_{eff} \cdot \sin \phi = V_{eff} \cdot I_{eff} \dots\dots\dots(2.16)$$

(Hermawan : 2013)

2.10. Faktor Daya

Faktor daya atau *power faktor(pf)* adalah perbandingan antara daya rata-rata atau daya aktif terhadap daya tampak. Secara matematis faktor daya dapat ditulis dalam persamaan sebagai berikut :

$$Pf = \frac{P}{S} = \frac{V_{eff} \cdot I_{eff} \cdot \cos \phi}{V_{eff} \cdot I_{eff}} \dots\dots\dots(2.17)$$

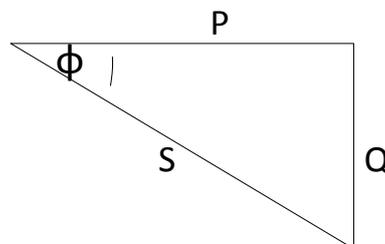
$$Pf = \cos \phi \dots\dots\dots(2.18)$$

Dengan menggunakan segitiga daya, maka dapat dilihat hubungan antara faktor daya dengan daya pada rangkaian RLC. Untuk beban yang bersifat induktif faktor daya akan bernilai positif, karena arus yang mengalir mengalami pergeseran fase (*Phase Shift*) dan tertinggal atau *lagging* terhadap tegangannya.



Gambar 2.13.Segitiga Daya pada Beban Induktif

Sedangkan pada beban yang bersifat kapasitif, faktor daya akan bernilai negatif karena arus yang mengalir mengalami pergeseran fase (*Phase Shift*) dan mendahului atau *leading* terhadap tegangannya.



Gambar 2.14.Segitiga Daya pada Beban Kapasitif

(Hermawan : 2013)

2.11. Energi Listrik

Kilowattjam, juga ditulis Kilowatt-jam. (simbol kW·h, kW h atau Kwh) adalah sebuah satuan energi.Energi yang dikirim oleh peralatan listrik biasanya diukur dan diberi biaya menggunakan satuan Kwh.Perlu diketahui bahwa Kwh adalah produk tenaga dalam Kilowatt dikali waktu dalam jam, bukan kW per h.

Contoh : Bila sebuah pemanas menggunakan tenaga 1000 watt (1 Kilowatt) dan pemanas tersebut dinyalakan selama satu jam maka energi yang digunakan adalah satu Kilowatt jam atau satu Kilowatt hour (sama dengan 3600 Kilojoule).

Menggunakan bola lampu 60 watt selama satu jam memakan 0.06 Kilowatt jam listrik. Bila digunakan selama seribu jam memakan 60 Kilowatt jam listrik.Bila bola lampu 100 watt dinyalakan selama satu jam per hari selama 30 hari, energi yang digunakan adalah :

$$100 \text{ W} \cdot (1\text{Kw}/1000) \cdot 30 \text{ h} = 3 \text{ Kwh} \dots\dots\dots(2.19)$$

(http://id.wikipedia.org/wiki/Kilowatt_jam)

