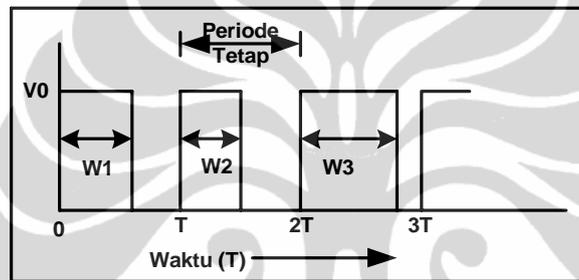


BAB II

LANDASAN TEORI

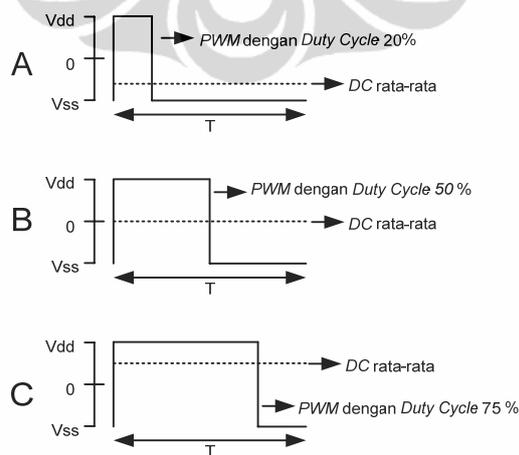
2.1 Definisi PWM

Sinyal *PWM* pada umumnya memiliki amplitudo dan frekuensi dasar yang tetap, namun, lebar pulsanya bervariasi. Lebar pulsa *PWM* berbanding lurus dengan amplitudo sinyal asli yang belum termodulasi (dalam hal ini adalah sinus). Dengan kata lain sinyal *PWM*, frekuensi gelombangnya adalah konstan (tetap) namun *duty cycle* bervariasi (antara 0% hingga 100%), menurut amplitudo sinyal aslinya. Sinyal *PWM* ditunjukkan Gambar 2.1 di bawah ini:



Gambar 2.1 Sebuah gelombang *PWM* [1]

Karena hanya ada 2 kondisi amplitudo sinyal *PWM* (yaitu *Low* dan *High*) maka dapat juga dikatakan bahwa sinyal *PWM* adalah sinyal yang informasinya terletak pada lebar pulsa.



Gambar 2.2 Tegangan rata-rata suatu *PWM* [2]

Gambar 2.2 menunjukkan perubahan *duty cycle* akan merubah tegangan *DC* rata-rata. Jika gelombang *PWM* yang dihasilkan merupakan dari sintesis gelombang sinus maka tegangan *DC* rata-rata akan menunjukkan sinyal sinus. Gelombang *PWM* yang difilter dapat digunakan untuk mengendalikan perangkat analog, dan menjadikan rangkaian pemfilter sebagai suatu *digital-to-analog converter (DAC)*.

Ada beberapa alasan untuk memilih *PWM*:

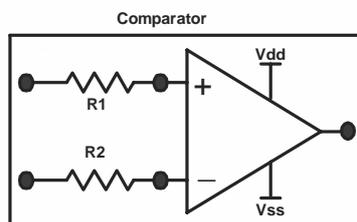
1. Dalam pembangkitan *PWM* hanya butuh 1 bit dari mikrokontroler (bila di bandingkan dengan *DAC* konvensional butuh 8bit, bahkan ada yang 10 bit).
2. Pada hakikatnya sinyal *PWM* merupakan sinyal yang *ON* dan *OFF*, *driver*-nya dapat dibangun dengan rangkaian *BJT* sedangkan *DAC* dibutuhkan rangkaian *driver* lebih kompleks..
3. *PWM* lebih sering dipakai dalam dunia industri.

2.2 Pembangkitan sinyal *PWM*

Sinyal *PWM* dapat dihasilkan dengan berbagai cara, antara lain dengan metode *comparator* dan mikrokontroler.

2.2.1 Pembangkitan sinyal *PWM* dengan *Comparator*

Dasar pembangkitan sinyal *PWM* adalah membandingkan tegangan $V(+)$ *non-inverting* dengan tegangan $V(-)$ *inverting* dengan menggunakan rangkaian *comparator*. Rangkaian *comparator* ditunjukkan seperti Gambar 2.3 di bawah ini:

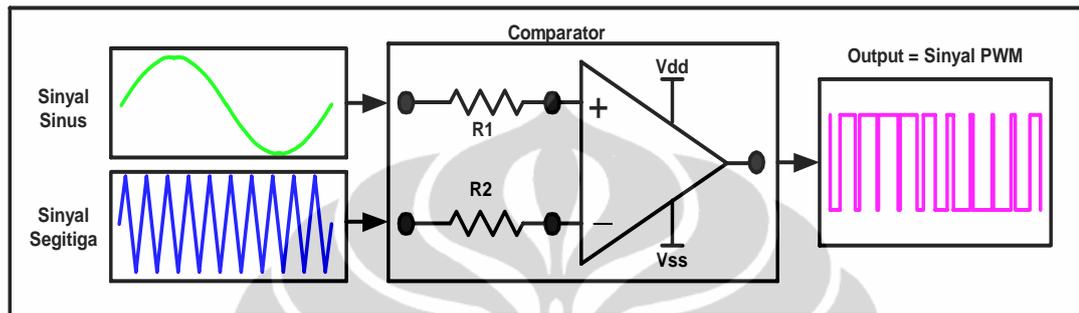


Gambar 2.3. Rangkaian *comparator*

Prinsip kerja pada rangkaian *comparator* diatas adalah bila tegangan non-inverting $V(+)$ lebih besar dari pada tegangan inverting $V(-)$ maka keluaran pembanding akan mendekati Vdd .

Sebaliknya, apabila tegangan *inverting* $V(-)$ lebih besar dari tegangan *non-inverting* $V(+)$ maka tegangan keluaran pembanding akan mendekati V_{ss} .

Bila diberikan input pada pin *inverting* berupa sinus dengan frekuensi tertentu, sedangkan pada input *non-inverting* berupa gelombang segitiga (berfungsi sebagai gelombang *carrier*) dengan frekuensi yang lebih tinggi (misal 10x frekuensi sinus) maka akan dihasilkan sinyal gelombang *PWM*.



Gambar 2.4 Blok diagram pembangkitan *PWM*

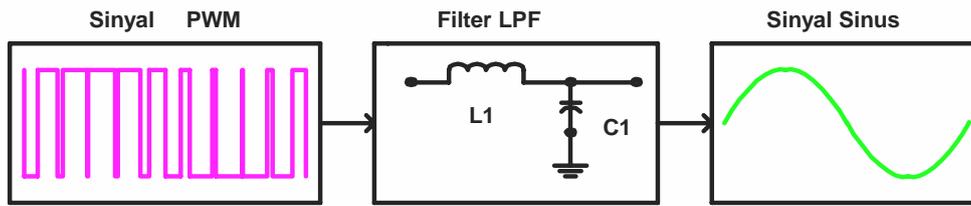
Blok diagram pembangkitan sinyal *PWM* dari rangkaian *comparator* adalah seperti terlihat pada Gambar 2.4.

2.2.2 Pembangkitan sinyal *PWM* dengan mikrokontroler

Mengingat bahwa hanya ada 2 kondisi amplitudo sinyal *PWM* (yaitu *low* dan *high*) sedang informasi *PWM* terletak pada perubahan lebar pulsanya, maka sinyal ini dapat di bangkitkan dengan mikrokontroler, yaitu dengan membuat suatu program yang menghasilkan sinyal seperti sinyal *PWM*. Penjelasan yang detail tentang pembangkitan sinyal *PWM* dengan mikrokontroler di bahas dalam bab 3.

2.3 Pemfilteran/ rekonstruksi sinyal sinus dari sinyal *PWM*

Pengkonversian sinyal *PWM* ke sinyal *analog* dapat dilakukan dengan menggunakan *filter low pass*. Ada dua macam *low pass filter* yang dibahas pada rangkaian ini yang pertama *RC low pass filter* yang kedua *LC low pass filter* yang akan digunakan untuk mengkonversi sinyal *PWM* menjadi gelombang sinus. Blok diagram pembangkitan sinyal sinusoidal dari sinyal *PWM* adalah sebagai berikut ini:



Gambar 2.5. Blok diagram pemfilteran

Pada rangkaian filter ini menggunakan *second order* (menggunakan rangkaian LC). Nilai LC ini harus memenuhi persamaan berikut ini [3]:

$$f_{c1} = \frac{1}{2\pi\sqrt{L_1 C_1}} \quad (P2.1)$$

Sebagai contohnya jika dilakukan perancangan *filter* yang memfilter suatu sinyal dengan frekuensi *carier* 8 kHz, maka penentuan nilai LC-nya adalah sebagai berikut:

$$L_1 C_1 = \frac{1}{(4 \pi^2 f^2)}$$

Anggap nilai kapasitansi yang dipakai adalah $L = 1 \text{ mH}$, sehingga

$$C_1 = \frac{1000}{(4 \pi^2 8000^2)}$$

$$C_1 = 390 \text{ nF}$$

Maka pasangan nilai LC yang dipakai untuk memfilter frekuensi *carier* 8 kHz adalah $L=1 \text{ mH}$ dan $C=390 \text{ nF}$.

Untuk frekuensi *carier* 80 Hz, maka penentuan nilai LC nya adalah sebagai berikut:

$$f_{c2} = \frac{1}{2\pi\sqrt{L_2 C_2}} \quad (P2.2)$$

Di karenakan $f_{c1} = 100 f_{c2}$ maka

$$f_{c2} = \frac{1}{100 \cdot 2\pi\sqrt{L_1 C_1}}$$

sehingga,

$$f_{c2} = \frac{1}{2\pi\sqrt{(100 L_1 \cdot 100 C_1)}}$$

dengan demikian diperoleh bahwa $L_2 = 100 L_1 = 100 \text{ mH}$ dan $C_2 = 100 C_1 = 39 \text{ uF}$, maka pasangan nilai LC yang dipakai untuk memfilter frekuensi *carier* 80 Hz adalah $L=100 \text{ mH}$ dan $C=39 \text{ uF}$.

2.4 Perancangan modulator 8-PSK

Modulasi dibagi menjadi 2 macam: modulasi analog dan modulasi digital. Ada 3 tipe dasar teknik modulasi digital, yaitu: *Amplitude-Shift Keying (ASK)*, *Frequency - Shift Keying (FSK)*, *Phase-Shift Keying (PSK)*. Pada *ASK*, amplitudo sinyal *carrier* di ubah berdasarkan respon informasi dan yang lainnya dijaga tetap. Bit 1 di transmisikan dengan sebuah sinyal carier dengan suatu amplitude tertentu. Pada *FSK*, kita mengubah respon frekuensi ke informasi, suatu frekuensi tertentu untuk nilai 1 dan frekuensi yang lain untuk nilai 0. Dalam *PSK*, kita mengubah fasa sinus untuk mengindikasikan informasi. Ada bermacam modulasi *PSK* diantaranya adalah *BPSK*, *QPSK* dan *8-PSK*.

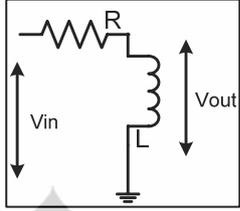
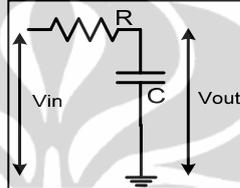
Pada modulasi *8-PSK* sinyal informasi terletak pada fasa sinus. Tiap fasa mengindikasikan suatu data biner tertentu. Fasa dalam konteks ini adalah sudut *start* dimana sinyal sinus mulai, sehingga dalam perancangan modulator *8-PSK* dibutuhkan rangkaian penggeser fasa, selain rangkaian penggeser fasa, juga dibutuhkan pendeteksi *zero cross* (*hardware* dan *software*), generator 3-bit (*software*), juga perancangan pensaklaran (*hardware* dan *software*).

2.4.1 Phase shifter / penggeseran fasa

Metode yang digunakan dalam penggeseran fasa adalah dengan membuat simulasi penggeseran fasa melalui program bantu simulator Multisim versi 8. Seperti diketahui bahwa rangkaian L dan C dapat mengubah fasa sinyal sinus, namun untuk merancang suatu penggeser fasa dengan menjadikan L atau C sebagai variabel penggesernya dirasa kurang praktis di karenakan tidak setiap nilai L maupun C yang dibutuhkan tersedia di pasar, oleh sebab itu digunakan variabel R yang dikombinasikan dengan L dan C, untuk menggeser fasa sesuai yang dibutuhkan.

Tabel 2.1 menjelaskan bahwa jika R dikombinasikan dengan L maupun C maka dapat berpengaruh pada pergeseran fasa. Pada kolom *Transfer Function* terlihat bahwa nilai R merupakan variabel yang mempengaruhi *Vout*.

Table 2.1 Rangkaian dasar penggeser fasa dengan R sebagai variabel

No	Komponen Pembentuk	Skema Rangkaian	Transfer Function
1	R L seri		$1 - \frac{L}{R} e^{-\frac{t}{R}}$
2	R C seri		$\frac{1}{RC} e^{-\frac{t}{RC}}$

Fasa-fasa yang digeser adalah fasa 0° , 30° , 45° , 60° , 90° , 120° , 135° , 150° , 180° , 210° , 225° , 240° , 270° , 300° , 315° dan 330° . Mengingat bahwa penggeseran fasa-fasa yang besar dapat direkayasa dari fasa yang lebih kecil, maka tidak semua fasa dibuat. Penjelasananya adalah sebagai berikut: fasa 330° dapat dibentuk dari 300° digeser 30° . Tabel 2.2 menjelaskan bahwa hanyalah fasa 0° , 30° , 45° , 60° -lah yang bisa digunakan untuk semua penggeseran fasa.

Table 2.2 Pembentukan fasa

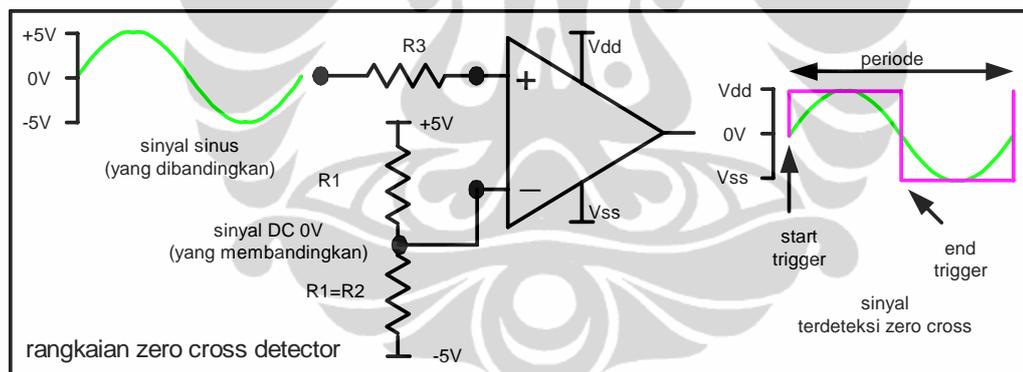
Fasa	Fasa Pembentuk
0	0
30	30
45	45
60	60
90	60 + 30
120	90 + 30
135	90 + 45
150	90 + 60
180	150 + 30
210	180 + 30
225	180 + 45
240	180 + 60
270	240 + 30
300	270 + 30
315	270 + 45
330	270 + 60

Sehingga cukup fasa fasa kecil yang di simulasikan dan dirancang yaitu fasa 30° , 45° , 60° , selebihnya dibangun dari fasa – fasa ini mengikuti alur pikiran yang telah di jelaskan pada Table 2.2.

2.4.2 Pendeteksian fasa /pendeteksian *zero cross*.

Pendeteksian *zero cross* merupakan salah satu point penting dalam perancangan hardware. Hal ini karena *zero cross* merupakan informasi yang mewakili titik dimana sinyal sinus melalui sumbu x sehingga jika ini diketahui maka dimulailah pensaklaran (*switching*).

Cara kerja rangkaian *zero cross* adalah dengan konsep *comparator* yaitu membandingkan input pada pin *non-inverting* dengan input pada pin *inverting*. Pada pin *Non-inv* diberi tegangan sinus, sedangkan pada pin *inv* di beri tegangan nol. Karena tegangan sinus bolak –balik antara $+5V$ ke $-5V$ maka setelah melalui rangkaian *comparator* akan dihasilkan sinyal kotak seperti Gambar 2.6.



Gambar 2.6 Rangkaian *Zero cross*

Hal terpenting dengan diketahuinya titik *zero cross* ini adalah bahwa pada saat fasa 0° melalui sumbu X, maka mikrokontroler harus mengirimkan sinyal *8-PSK* (mikrokontroler melakukan pensaklaran untuk menghasilkan sinyal *8-PSK*).

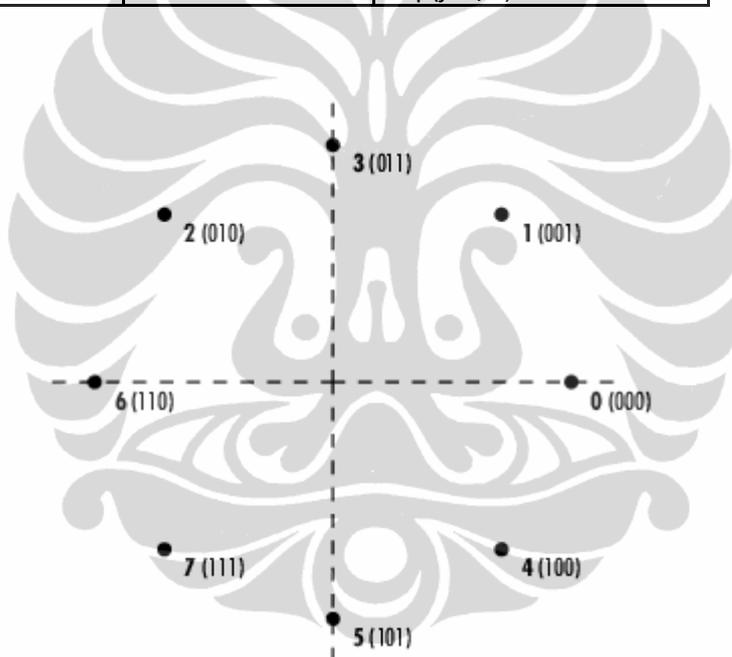
2.4.3 Pen-saklar-an (*switching*).

Yang di maksud dengan pen-saklar-an (*switching*) adalah pemodulasian *8-PSK*. Modulasi *8-PSK* yang di buat dalam penelitian ini adalah *8 Gray-PSK*,

artinya urutan *biner gray code* dimodulasikan dengan fasa yang bersesuaian, sesuai dengan Tabel 2.3.

Tabel 2.3 Input dan out modulasi 8 *Gray-PSK*

Urutan Biner	Input Modulator <i>Gray Code</i>	Output Modulator fasa sinus
0	000	$\exp(0)$
1	001	$\exp(j\pi/4)$
2	011	$\exp(j\pi/2) = \exp(j2\pi/4)$
3	010	$\exp(j3\pi/4)$
4	110	$\exp(j\pi) = \exp(j4\pi/4)$
5	111	$\exp(j5\pi/4)$
6	101	$\exp(j3\pi/2) = \exp(j6\pi/4)$
7	100	$\exp(j7\pi/4)$



Gambar 2.7 Fasor 8 *Gray-PSK*

Peng-kode-an *Gray* merupakan teknik yang sering digunakan dalam skema modulasi multilevel guna meminimalisir *bit error rate* dengan mengatur simbol modulasi sedemikian rupa, sehingga simbol representasi *biner* yang berdekatan hanya berbeda 1 bit.

Tabel 2.3, memperjelas bahwa keseluruhan efek subsistem ini merupakan pemetaan kode Gray yang ditunjukkan Gambar 2.7. Perhatikan bahwa angka pada kolom kedua dari Tabel 2.3, terlihat dengan urutan berlawanan jarum jam.

Dengan demikian Tabel 2.3 merupakan dasar yang dipakai dalam pensaklaran 8 *Gray-PSK*.

2.5 Mikrokontroler Atmega16

Mikrokontroler Atmega16 merupakan mikrokontroler produksi Atmel, spesifikasi umumnya adalah :

- Memori:
 - 16K Bytes of *In-System Self-Programmable Flash*
 - 512 Bytes *EEPROM*
 - 1K Byte *Internal SRAM*
- Port Input output
 - 32 *Programmable port input output*
 - 40-pin *PDIP*, 44-lead *TQFP*, and 44-pad *MLF*

Deskripsi pin mikrokontroler tersebut adalah:

VCC
merupakan pin tegangan suply.

GND
merupakan pin ground.

Port A (PA7..PA0)

Port A menyediakan input analog bagi *A/D Converter*. Port A juga menyediakan *port input-output 8-bit bi-directional*, jika *A/D Converter* tidak dipakai. Port A berada dalam kondisi *tri-state* saat reset terjadi, meskipun *clock* tidak berjalan.

Port B (PB7..PB0)

Port B merupakan *port input-output 8-bit bi-directional* dengan resistor *pull-up* internal. Port B berada dalam kondisi *tri-state* saat reset terjadi, meskipun *clock* tidak berjalan.

Port C (PC7..PC0)

Port C merupakan *port input-output 8-bit bi-directional* dengan resistor *pull-up* internal.

Port D (PD7..PD0)

Port D merupakan *input-output 8-bit bi-directional* dengan resistor *pull-up* internal.

RESET

Merupakan input reset. Tegangan *low* pada pin ini akan membangkitkan *reset* meskipun *clock* tidak berjalan.