

BAB II

MATERI PENUNJANG

2.1. KONSEP DASAR MUSIK

Musik dibangun oleh unsur-unsur dasar yaitu nada, tempo, dinamika, dan irama. Unsur-unsur dasar ini kemudian digabungkan sedemikian sehingga dihasilkan suatu musik yang harmonis.

2.1.1. Nada (*Pitch*)

Suara dihasilkan oleh sumber bunyi yang bergetar secara teratur dalam suatu waktu tertentu. Jumlah getaran per satuan waktu dinamakan dengan frekuensi dan satuannya dinyatakan dengan Hertz (Hz). Suara yang bisa didengar oleh telinga manusia normal adalah suara dengan frekuensi antara 20 hingga 20.000 Hz. Sedangkan nada adalah suara dengan frekuensi tertentu. Sebagai patokan ditetapkan nada A dengan frekuensi 440 Hz. Frekuensi dari nada-nada yang lain mengikuti aturan sebagai berikut:

- C : $C = 1$, disebut juga C adalah *prime* dari C
- D : $C = 9/8$, disebut juga D adalah *seconde* dari C
- E : $C = 5/4$, disebut juga E adalah *terst* dari C
- F : $C = 4/3$, disebut juga F adalah *quart* dari C
- G : $C = 3/2$, disebut juga G adalah *quint* dari C
- A : $C = 5/3$, disebut juga A adalah *sext* dari C
- B : $C = 15/8$, disebut juga B adalah *septime* dari C
- C' : $C = 2$, disebut juga C' adalah *oktaf* dari C

Hal-hal lainnya yang berkaitan dengan nada adalah:

2.1.1.1. Tangga Nada.

Pada dasarnya, nada-nada dalam musik merupakan susunan sejumlah nada yang selalu berulang, karena tersusun oleh nada-nada yang sama meski berbeda frekuensinya. Setiap perulangan yang terjadi kemudian dinamakan *oktaf*. Sedangkan tangga nada adalah kumpulan nada dalam jumlah tertentu dalam setiap oktafnya. Ada 3 macam tangga nada yang dikenal dalam musik, yaitu:

- a. Tangga nada *Diatonis*, yaitu tangga nada yang terdiri atas 8 nada dalam setiap oktafnya.

$$\underbrace{C D E F G A B C'}_{1 \text{ oktaf}} \underbrace{D' E' F' G' A' B' C'}_{1 \text{ oktaf}}$$

- b. Tangga nada *Kromatis*, yaitu tangga nada yang terdiri atas 13 nada dalam setiap oktafnya. Tangga nada kromatis merupakan tangga nada diatonis yang ditambah dengan 5 nada sisipan, yaitu:

C_{\sharp} (cis) atau D_{\flat} (des), D_{\sharp} (dis) atau E_{\flat} (es), F_{\sharp} (fis) atau G_{\flat} (ges), G_{\sharp} (gis) atau A_{\flat} (as), A_{\sharp} (ais) atau B_{\flat} (bes).

Susunan selengkapnya adalah sebagai berikut:

$$\underbrace{C C_{\sharp}(D_{\flat}) D D_{\sharp}(E_{\flat}) E F F_{\sharp}(G_{\flat}) G G_{\sharp}(A_{\flat}) A A_{\sharp}(B_{\flat}) B C'}_{1 \text{ oktaf}}$$

- c. Tangga nada *Pentatonis*, yaitu tangga nada yang terdiri atas 5 nada dalam setiap oktafnya (do, mi, fa, sol, dan si). Tangga nada Pentatonis biasa digunakan dalam musik tradisional, terutama di Jawa.

2.1.1.2. Jarak Nada.

Jarak nada adalah selang yang dimiliki antara 2 buah nada. Sebagai patokan telah ditetapkan jarak tiap pasang nada yang saling berurutan dalam tangga nada kromatis adalah $\frac{1}{2}$.

Dengan demikian jarak nada yang ada adalah:

$$\begin{array}{cccccccc} C & D & E & F & G & A & B & C' \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ 1 & 1 & \frac{1}{2} & 1 & 1 & 1 & 1 & \frac{1}{2} \end{array}$$

Atau jika dinyatakan dalam notasi angka:

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 1 \\ \text{do} & \text{re} & \text{mi} & \text{fa} & \text{sol} & \text{la} & \text{si} & \text{do} \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ 1 & 1 & \frac{1}{2} & 1 & 1 & 1 & 1 & \frac{1}{2} \end{array}$$

2.1.1.3. Nada Dasar.

Nada dasar adalah tetapan untuk nada do natural pada lagu yang bersangkutan. Sebagai contoh, untuk nada dasar do=C, maka tangga nadanya adalah:

$$\begin{array}{cccccccc} C & D & E & F & G & A & B & C' \\ \text{do} & \text{re} & \text{mi} & \text{fa} & \text{sol} & \text{la} & \text{si} & \text{do} \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ 1 & 1 & \frac{1}{2} & 1 & 1 & 1 & 1 & \frac{1}{2} \end{array}$$

Dan untuk nada dasar do=A, maka tangga nadanya adalah:

$$\begin{array}{cccccccc} A & B & C_{\sharp} & D & E & F_{\sharp} & G_{\sharp} & A' \\ \text{do} & \text{re} & \text{mi} & \text{fa} & \text{sol} & \text{la} & \text{si} & \text{do} \\ \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} & \underbrace{\quad} \\ 1 & 1 & \frac{1}{2} & 1 & 1 & 1 & 1 & \frac{1}{2} \end{array}$$

2.1.2. Dinamika (*Dynamics*)

Dinamika adalah kualitas keras atau lembutnya sebuah lagu dinyanyikan. Satuan kuantitas dari dinamika dinyatakan dalam *decibels* (dB). Istilah-istilah umum yang sering digunakan untuk menyatakan dinamika lagu, antara lain seperti yang terlihat pada tabel 2.1.

Tabel 2.1
Macam-macam dinamika

Istilah	Bentuk Singkal	Arti Dinamika
Pianissimo	<i>pp</i>	sangat lembut
Piano	<i>p</i>	lembut
Mezzo piano	<i>mp</i>	agak lembut
Mezzo forte	<i>mf</i>	agak keras
Forte	<i>f</i>	keras
Fortissimo	<i>ff</i>	sangat keras

Dinamika suatu lagu tidak harus sama sepanjang lagu. Perubahan dinamika suatu lagu dinyatakan dengan tanda < atau >.

Tanda < yang disebut *crescendo* (*cresc*) menyatakan bahwa not dibawah tanda ini dinyanyikan dengan dinamika yang makin lama makin keras.

Sedangkan tanda > yang disebut *decrescendo* (*decresc*) menyatakan bahwa not dibawah tanda ini dinyanyikan dengan dinamika yang makin lama makin lembut.

Crescendo dan decrescendo digunakan untuk beberapa not. Sedangkan tanda < atau > yang hanya digunakan untuk sebuah not, dinamakan dengan aksent (*accent*). Tanda aksent < menyatakan bahwa not dibawahnya dinyanyikan lebih keras, sedangkan tanda aksent > menyatakan bahwa not dibawahnya dibawakan dengan lebih lembut dibandingkan dengan not-not lainnya.

2.1.3. Tempo

Tempo adalah ukuran cepat atau lambatnya sebuah lagu dinyanyikan, atau dengan kata lain tempo adalah ukuran dasar tenggat waktu antar ketukan di dalam suatu lagu. Tempo lagu dinyatakan dalam satuan *metronome marks* (*mm*). Sebagai perbandingan untuk pembuatan lagu, lagu *slow* memiliki tempo kurang lebih 42 *mm*, lagu *march* kurang lebih 100 *mm*, dan lagu *metal* kurang lebih 160 *mm*.

Istilah-istilah umum yang sering digunakan untuk menyatakan dinamika suatu lagu, antara lain seperti yang terlihat pada tabel berikut:

Tabel 2.2
Macam-macam tempo

Istilah	Arti
Adagio	slow, lambat
Andante	Lambat tetapi tidak terlalu
Moderato	Sedang
Allegretto	Cepat tetapi tidak terlalu
Allegro	Cepat
presto	Sangat cepat

Perubahan tempo suatu lagu dinyatakan dengan istilah-istilah:

- *accelerando (accel)* yang artinya makin cepat
- *ritardando (rit)* atau *rallentando (rall)* yang artinya makin lambat
- *pia lento* yang artinya lebih lambat
- *pia allegro* yang artinya lebih cepat
- *fermata* (∩) yang artinya dihentikan sementara (ditambah ketukan sesuai selera).

2.1.4. Irama (Rhythm)

Irama mencakup segala hal yang menyangkut waktu dalam musik yaitu ketukan (*beat*), bar (*measure*), dan birama (*meter*).

2.1.4.1. Ketukan (beat)

Ketukan (beat) adalah tenggat waktu permainan antara suatu nada dengan nada berikutnya. Beat dari lagu mars lebih cepat dari lagu pop, misalnya.

2.1.4.2. Bar (measure)

Bar (measure) adalah suatu ukuran jumlah ketukan dalam satu perulangan penekanan nada. Setiap bar diawali dengan suatu penekanan dan diakhiri dengan suatu garis vertikal yang disebut garis bar. Misalnya untuk lagu Burung Tantina, setiap bar terdiri atas 3 ketukan (tiap 3 ketukan diawali dengan satu penekanan), sedangkan lagu Apuse terdiri atas 4 ketukan untuk setiap bar-nya.

2.1.4.3. Birama (meter)

Birama adalah suatu tetapan yang menyatakan jumlah ketukan dalam setiap bar dan harga dari masing-masing ketukan. Macam-macam birama yang dikenal yaitu $\frac{2}{2}$, $\frac{2}{4}$, $\frac{3}{4}$, dan $\frac{6}{8}$.

Dalam suatu birama yang dinyatakan sebagai bilangan rasional $\frac{a}{b}$, a menyatakan jumlah ketukan dalam setiap bar, sedangkan harga dari masing-masing ketukan adalah $\frac{1}{b}$. Contohnya untuk birama $\frac{3}{4}$, maka jumlah ketukan dalam setiap bar adalah 3, dan harga dari masing-masing ketukan adalah $\frac{1}{4}$. Begitu juga untuk jenis-jenis birama yang lain.

2.1.5. Notasi Musik (Music Notation)

Penulisan atau notasi musik dinyatakan dalam 2 bentuk, yaitu notasi angka dan notasi balok.

2.1.5.1 Notasi Angka

Notasi angka dinyatakan dengan angka 1 sampai dengan 7 dan lima nada lainnya (nada-nada sisipan). Dalam tangga nada diatonis, notasi angka dinyatakan sebagai berikut:

Not	:	1	2	3	4	5	6	7	$\dot{1}$
Solmisasi	:	do	re	mi	fa	sol	la	si	do
Jarak	:	1	1	$\frac{1}{2}$	1	1	1	1	$\frac{1}{2}$

Sedangkan dalam tangga nada kromatis dimana jarak antar nada yang berurutan adalah $\frac{1}{2}$, ditulis sebagai:

Not	:	1	\times 2	\times 3	4	\times 5	\times 6	7	$\dot{1}$				
Solmisasi	:	do	di	re	ri	mi	fi	sol	set	la	le	si	do

Perhentian (nada diam) dalam notasi angka dinyatakan dengan angka 0 (nol).

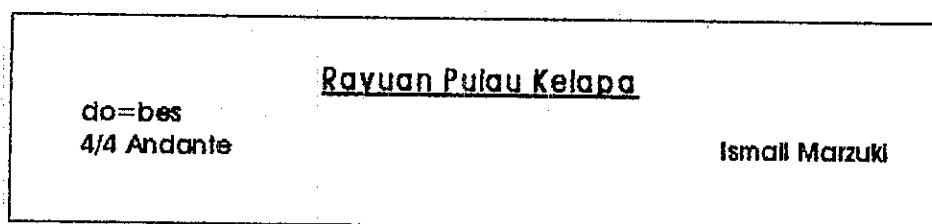
Penghitungan harga ketukan dilakukan dengan kesatuan not, not titik berarti memperpanjang not di depannya. Contoh:

$$\left| 1 \quad \overline{2\ 3} \quad 4 \quad \overline{.4} \right| 3 \dots \left| \right.$$

Nada pertama (do) berharga 1 ketuk karena berdiri sendiri. Nada ke-2 (re) dan nada ke-3 masing-masing berharga $\frac{1}{2}$ ketuk karena menjadi 1 kesatuan. Nada ke-4 (fa) berharga $1\frac{1}{2}$ ketuk (1 karena berdiri sendiri dan $\frac{1}{2}$ karena ditambah not titik yang berharga $\frac{1}{2}$). Nada ke-5 (fa) berharga $\frac{1}{2}$, sedangkan nada ke-6 (mi) berharga 4 ketuk (1 berdiri sendiri ditambah 3 ketuk not titik). Jadi, setiap kesatuan not berharga 1 ketuk.

Dalam penulisan musik dengan notasi angka, nada dasar, tempo, dan birama ditulis di awal lagu. Sedangkan tanda-tanda yang mengatur dinamika (aksen, crescendo, decrescendo) ditulis diatas nada yang dimaksud.

Contoh, dalam lagu Rayuan Pulau Kelapa, tertulis:



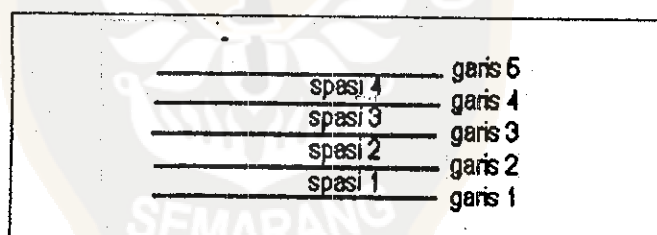
Gambar 2.1
Ilustrasi informasi lagu

Berarti bahwa lagu ini mempunyai nada dasar bes, birama 4/4 dan tempo andante (agak lamban), serta dibuat oleh Ismail Marzuki.

2.1.5.2 Notasi Balok

a. Garis paranada (*staff*)

Garis paranada terdiri dari 5 buah garis dan 4 spasi yang merupakan areal penempatan penulisan not balok. Penomoran garis dan spasi dilakukanurut dari bawah.

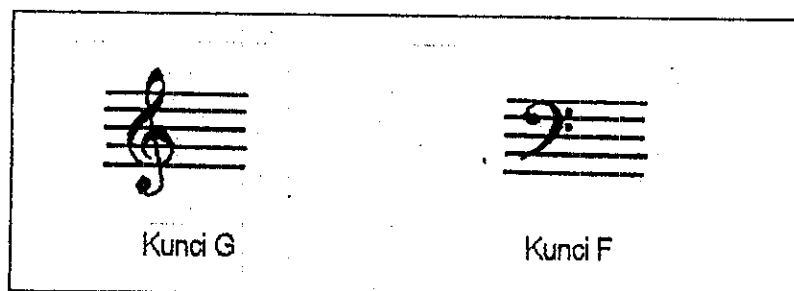


Gambar 2.2
Garis paranada

b. Tanda Kunci (*clefs*)

Ada 2 tanda kunci yaitu kunci G dan kunci F. Kunci G biasanya diperuntukkan bagi suara piano, sedangkan kunci F diperuntukkan bagi suara

bas.



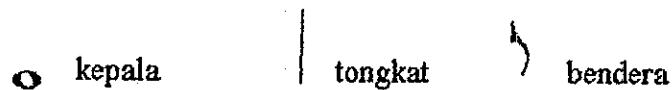
Gambar 2.3
Gambar tanda kunci

c. Not (*note*) dan Perhentian (*rest*)

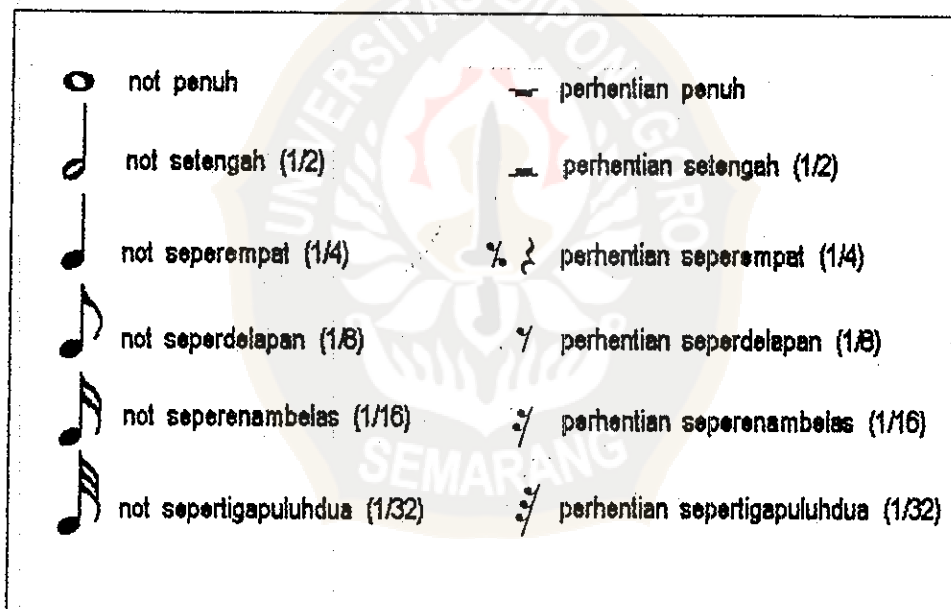
Not adalah tanda yang menyatakan suatu nada tertentu beserta harganya.

Dalam notasi balok, not digambarkan dengan 3 bagian yaitu kepala, tongkat, dan bendera.

Bentuknya adalah sebagai berikut:



Sedangkan perhentian (nada diam) digambarkan dengan bentuk berbeda menurut harganya.



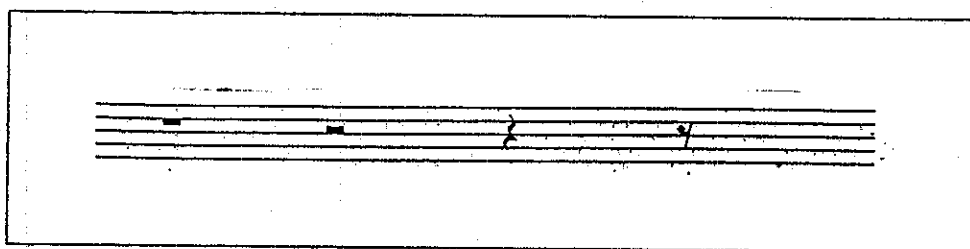
Gambar 2.4
Berbagai macam not dan perhentian

d. Aturan Penulisan

d.1). Penulisan not dan perhentian.

Not ditulis / diletakkan pada areal garis paranada, boleh pada spasi atau tepat pada garisnya. Jika tinggi/rendah not yang bersangkutan mengharuskan not diletakkan diluar garis paranada, maka dibuat garis bantu (khusus untuk not yang dimaksud).

Sedangkan perhentian ditulis dengan bentuk dan letak yang baku (gambar 2.5).



Gambar 2.5
Penulisan perhentian

d.2). Penulisan nada dasar

Dalam not balok, nada dasar dituliskan seperti pada gambar 2.6.



Gambar 2.6
Penulisan nada dasar

d.3). Tanda tempo, dinamika, dan birama serta tanda-tanda lain ditulis seperti pada not angka.

e. Pemberian harga ketukan

e.1). Not Titik (*dotted not*)

Titik yang mengikuti suatu not berarti menambah ketukan sebesar $\frac{1}{2}$ kali not yang diikuti.

Beberapa bentuk *dotted not* yang dikenal adalah:

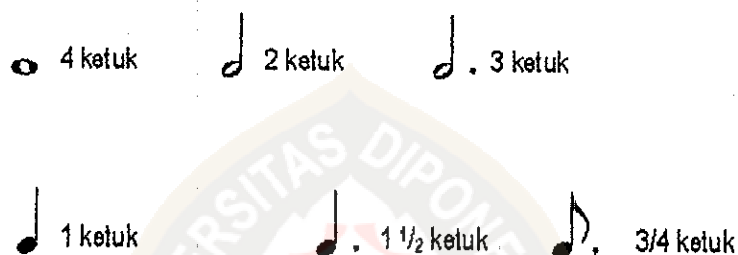
$$\text{♩.} = \text{not } 3/4 \quad \text{♪.} = \text{not } 3/8 \quad \text{♫.} = \text{not } 3/16$$

e.2). Pemberian harga ketukan

Pemberian harga not disesuaikan dengan aturan birama lagu. Pada birama $\frac{4}{4}$ misalnya, dalam setiap bar terdiri atas 4 ketukan dengan masing-masing ketukan menyatakan harga notnya adalah $\frac{1}{4}$.

Contoh pemberian ketukan:

Pada birama $\frac{4}{4}$:



2.1.6. Harmoni

Harmoni atau keserasian dalam suatu lagu dapat dicapai apabila tempo, dinamika, nada, serta lirik (syair) lagu sesuai dengan jiwa yang terkandung di dalam lagu tersebut.

Selain itu, di dalam suatu lagu juga biasa ditambahkan akord (*accord*) yaitu memainkan beberapa nada sekaligus. Akord dimainkan bersamaan dengan dimainkannya not pertama dari setiap bar (*measure*). Akord-akord pokok yang sering digunakan adalah akord I terdiri atas nada-nada 1 (do), 3 (mi), 5(sol), disebut *tonika*, akord V terdiri atas nada-nada 5 (sol), 7 (si), 2(re), disebut *dominan*, akord IV terdiri atas nada-nada 4 (fa), 6 (la), 1(do), disebut *subdominan*, akord vi terdiri atas nada-nada 6 (la), 1 (do), 3(mi), disebut *tonika pembantu*, akord iii terdiri atas nada-nada 3 (mi), 5 (sol), 7(si), disebut *dominan pembantu*, akord ii terdiri atas nada-nada 2 (re), 4 (fa), 6(la), disebut *subdominan pembantu*.

2.2. KONSEP PEMROGRAMAN

2.2.1. Program

Program adalah seperangkat ekspresi, pernyataan atau kombinasi antara keduanya yang disusun berdasarkan urutan langkah untuk menyelesaikan suatu masalah, dan diimplementasikan melalui suatu bahasa pemrograman agar dapat dieksekusi oleh komputer.

Program yang baik adalah program yang memenuhi kriteria-kriteria berikut:

- a. Tahan uji, dalam pengertian dapat menyelesaikan masalah yang dijanjikan serta mendeteksi kemungkinan kesalahan pemakaian program.
- b. Perawatannya mudah, dalam arti mudah diubah dan dimodifikasi.
- c. Luwes, dalam arti dapat ditransfer ke dalam komputer yang berbeda dengan modifikasi minimum.
- d. *User friendly*, artinya mudah dimengerti dan digunakan oleh pemakai atau pemrogram yang lain.
- e. Penampilannya menarik, kerjanya cepat dan efisien.

2.2.2. Bahasa Pemrograman

Bahasa pemrograman adalah suatu aturan / tata cara penulisan program yang mengandung 2 unsur penting, yaitu:

- a. *Syntax*, berisi aturan-aturan gramatikal yang mengatur tata cara penulisan kata, ekspresi, dan pernyataan.
- b. *Semantik*, berisi aturan-aturan yang menyatakan suatu arti.

2.2.3. Pemrograman

Pemrograman adalah proses pengimplementasian urutan langkah untuk menyelesaikan suatu masalah dengan menggunakan suatu bahasa pemrograman tertentu.

Langkah-langkah pemrograman:

1. Pemahaman masalah yang akan diselesaikan dengan menspesifikasikan program, dimana terdapat hubungan eksak antara input dan output.
2. Merencanakan suatu metode penyelesaian masalah. Untuk program yang besar dapat dipecah-pecah menjadi beberapa program yang lebih kecil (modul).
3. Mengembangkan metode yang sudah ada dengan menyusun *algoritma* dan *flowchart* untuk mempermudah penulisan program.
4. Menulis program dalam bahasa pemrograman tertentu.
5. Menguji program yang telah dibuat. Kebenaran *syntax* (penulisan) diuji oleh komputer, sedangkan kebenaran logika penulisan dapat dilihat dari *out-put* program.
6. Membuat dokumentasi yang antara lain berisi rincian teknis pemakaian program.

2.2.4. Algoritma

Algoritma adalah suatu metode khusus yang terdiri atas serangkaian langkah terstruktur dan ditulis secara matematis yang akan dikerjakan untuk menyelesaikan suatu masalah dengan bantuan komputer.

Ciri-ciri algoritma yang baik:

- a. Merupakan metode yang tepat dalam memecahkan masalah secara logika.
- b. Menghasilkan output yang tepat dan benar dalam waktu yang singkat.
- c. Ditulis dalam bahasa standart secara sistematis sehingga tidak menimbulkan arti ganda.
- d. Mudah dipahami, dan proses berakhir setelah semua langkah yang tertulis selesai dikerjakan.

2.2.5. Bagan Alir (*Flowchart*)

Bagan alir (*Flowchart*) adalah sekumpulan simbol yang menggambarkan urutan operasi, aliran data, aliran kontrol dan operasi logika dalam proses penyelesaian masalah.

Berikut ini adalah bentuk dari simbol-simbol yang dikenal dalam flowchart, nama, dan arti dari simbol yang dimaksud:



Proses, berisi instruksi yang akan dijalankan, misalnya menghitung, memindah data, dan lain-lain.



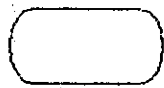
Input-output, berisi instruksi untuk membaca data atau menuliskan hasil proses.



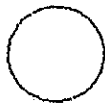
Decision, digunakan untuk menentukan pilihan yang akan diambil untuk langkah berikutnya berdasarkan kondisi yang ada.



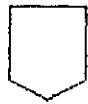
Preparation, digunakan untuk memberi harga awal, dan mengubah harga indeks.



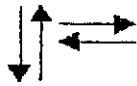
Terminal, merupakan tanda dimulai dan diakhirinya keseluruhan proses.



Connector, sebagai tanda memutuskan suatu aliran proses dan kemudian menyambungkannya pada bagian lain di halaman yang sama.



Off Page Connector, berfungsi seperti connector, digunakan jika sambungannya berada pada halaman yang berlainan



Arrows, menunjukkan arah aliran proses berjalan

2.3. BAHASA PEMROGRAMAN C

Bahasa pemrograman C merupakan bahasa pemrograman beraras menengah (*semi high level language*), diciptakan oleh Denis M. Ritchie pada tahun 1972. Keunggulan bahasa C diantaranya adalah kecepatan eksekusinya yang hampir sama dengan bahasa rakitan (bahasa mesin); bisa melakukan akses secara langsung ke perangkat komputer (prosesor, video, keyboard, dll), merupakan bahasa pemrograman yang terstruktur dan modular sehingga mudah ditulis, dipahami, dan dikembangkan; serta dapat digunakan untuk berbagai bidang aplikasi seperti sistem keuangan, sains, basis data, seni, dan lain-lain.

2.3.1. Komponen Bahasa C

2.3.1.1. Character Set

Character Set merupakan kumpulan karakter yang digunakan untuk penulisan bahasa C, terdiri atas:

- Huruf-huruf dalam abjad latin: (A-Z, a-z)
- Angka : 0-9
- Karakter khusus: !#%^&(){}";:<>? dan lain-lain atau kombinasinya seperti \n, \r, dan sebagainya.

2.3.1.2. Identifier

Identifier adalah nama yang diberikan untuk unsur-unsur dalam program C seperti variabel, fungsi, dan larik (*array*). Aturan penulisan identifier, terdiri atas:

- Terdiri atas huruf saja atau huruf dan angka
- Karakter pertama harus berupa huruf atau garis bawah
- Panjang maksimum 31 karakter
- Tidak boleh menggunakan *key-word*.

2.3.1.3. Kata kunci (*key-word*)

Kata kunci (*key-word*) adalah kata-kata yang sudah memiliki arti khusus dan telah disediakan oleh bahasa C untuk maksud tertentu. Kata-kata yang termasuk *key-word* adalah sebagai berikut:

asm	do	if	static
auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	return	union
const	float	short	unsigned
continue	for	signed	void
default	goto	sizeof	while

Kata-kata kunci (*keyword*) dalam C

2.3.1.4. Tipe Data

Ada 2 jenis tipe data dalam bahasa C yaitu:

1. Tipe data dasar, yang terdiri atas karakter (*char*), integer (*int*), floating-point (*float*), double-floating-point (*double*), kosong (*Void*).

Jangkauan dari masing-masing tipe data dasar diberikan dalam tabel berikut

Tabel 2.3
Jangkauan tipe data

Tipe Data	Jumlah Bit/Byte	Jangkauan
Char	8/1	-128 s/d 127
Int	16/2	-32768 s/d 32767
Float	32/4	3,4xE-38 s/d 3,4xE38
Double	64/8	1,7xE-308 s/d 1,7xE308
Void	-	3,4xE-4932 s/d 3,4xE4932

2. Tipe data tambahan/turunan (*derived data type*), terdiri atas:

- Penunjuk (pointer)

Pointer adalah variabel yang menyimpan alamat dari obyek data yang lain, deklarasinya sebagai berikut:

```
Jenis_data *nama_pointer
```

- Larik (array)

Larik (array) adalah sekelompok data yang bertipe sama. Masing-masing data dinamakan elemen array, yang masing-masing dibedakan dengan menggunakan indeks.

Deklarasinya :

```
Jenis_data nama_larik [n1][n2]...[ni]
```

Dengan [n1][n2]...[ni] adalah jumlah elemen pada dimensi ke-i dan indeks dimulai dari 0 sampai n-1.

- Struktur (struct)

Struktur adalah sekumpulan data yang berbeda jenis tetapi saling berhubungan satu sama lain.

- Serikat (union)

Serikat adalah variabel yang dapat menyimpan data dari jenis yang berbeda-beda. Serikat hampir sama dengan struktur, bedanya anggota-anggota suatu serikat tidak dapat aktif secara bersama-sama.

2.3.1.5. Konstanta

- a). Konstanta Integer dan Floating-Point, merupakan konstanta numerik, dapat diawali dengan tanda minus (-), tidak boleh ada tanda koma atau spasi, tidak boleh melebihi jangkauan data. Konstanta integer merupakan bilangan yang berharga bulat dan dapat ditulis dalam sistem desimal, sistem oktal, serta sistem hexadesimal. Konstanta floating-point merupakan bilangan dengan sistem desimal yang mengandung titik desimal atau eksponensial.
- b). Konstanta karakter, merupakan karakter tunggal yang diapit oleh tanda petik tunggal.
- c). Konstanta string, terdiri atas sederetan karakter yang ditulis diantara tanda petik ganda.

2.3.1.6. Variabel

Variabel digunakan untuk mewakili suatu jenis informasi di dalam suatu bagian program.

2.3.1.7. Operator dan Operand

Operator adalah tanda yang digunakan untuk melakukan operasi tertentu, sedangkan operand adalah obyek data yang dikenakan operasi oleh operator.

a). Operator Aritmatika (*Arithmetic Operator*)

Terdiri atas tanda-tanda dalam operasi aritmatika yaitu tanda : + (penjumlahan), - (pengurangan), *(perkalian), / (pembagian), dan %(sisa pembagian bilangan bulat).

b). Operator Relasi

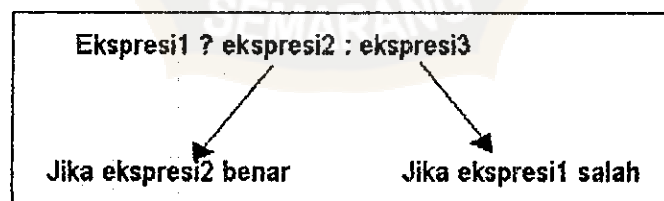
Digunakan untuk membandingkan ekspresi, terdiri atas tanda-tanda < (kurang dari), > (lebih dari), <= (kurang dari atau sama dengan), >= (lebih dari atau sama dengan), == (sama dengan), dan != (tidak sama dengan).

c). Operator Logika

Digunakan untuk mengkombinasikan dua atau lebih operator relasi, terdiri atas: && (dan), || (atau), dan ! (tidak/bukan).

d). Operator Kondisional

Terdiri atas tanda ? dan : yang digunakan sebagai statemen if-else. Bentuk umum :



Gambar 2.7.
Ilustrasi operator kondisional

e). Operator Penugasan (*assignment operator*)

Terdiri atas =, +=, -=, *=, /=, dan %=.

Contoh: $x += 10$ berarti $x = x + 10$

Tingkatan pengerjaan operator dapat dilihat dari tabel 2.4.

Tabel 2.4
Tingkatan pengerjaan operator

Tingkat	Operator
1	++ --
2	* / %
3	+ -
4	< <= > >=
5	== !=
6	&&
7	
8	?:
9	= += -= *= /= %=

2.3.1.8. Ekspresi

Ekspresi adalah kombinasi antara operand dan operator.

Contoh: $3*x+15$

Luas=panjang*lebar

2.3.1.9. Statement

Statement/ Pernyataan adalah unsur dasar pembentukan suatu program.

a. Expression statement

Penulisannya diikuti dengan tanda titik koma (;)

Contoh: `i++;`

`Luas_segi4=panjang*lebar;`

b. Compound atau Block statement

Terdiri atas dua atau lebih statement yang dikelompokkan menjadi satu.

c. Control statement (control flow)

Adalah statement yang digunakan untuk mengendalikan langkah-langkah dari program.

c.1. If-else

Digunakan untuk melakukan test logika dan melaksanakan pekerjaan berikutnya berdasarkan hasil test.

```

If (ekspresi)
    Statement1;  → dikerjakan jika ekspresi benar
Else
    Statement2;  → dikerjakan jika ekspresi salah

```

Gambar 2.8.
Bentuk umum statement if-else

c.2. Switch

Memungkinkan pilihan lebih dari satu berdasarkan suatu ekspresi.

```

switch (ekspresi)
{
    case label-1
        statement-1;
    case label-2
        statement-2;
    .....
    case label-n
        statement-n;
    Default:
        statement-z;
}

```

Gambar 2.9.
Bentuk umum statement switch

c.3. For-loop

Digunakan untuk pengulangan sekelompok instruksi dalam suatu program sampai suatu kondisi terpenuhi.

```

For (exp1; exp2; exp3)
    Statement;

```

dimana: exp1 → memberi harga awal
exp2 → kondisi yang harus dipenuhi
..exp3 → mengubah harga variabel

Gambar 2.10
Bentuk umum statement for

c.4. While-loop

Digunakan untuk melaksanakan satu atau sekelompok statement selama suatu kondisi tertentu dipenuhi.

Bentuk umum:

```
While (kondisi)
    Statement;
```

2.3.2. Input Output Dasar

Ada beberapa fungsi yang digunakan untuk operasi input output dasar, yaitu:

a. Fungsi *getchar()*;

Digunakan untuk membaca satu karakter dari peralatan input standart (keyboard).

b. Fungsi *putchar()*;

Digunakan untuk menampilkan satu karakter ke peralatan output standart (monitor).

c. Fungsi *printf()*;

Digunakan untuk mencetak data ke monitor atau printer.

d. Fungsi *scanf()*;

Digunakan untuk membaca data/informasi dari peralatan input standart (keyboard, disket).

e. Fungsi *gets()*;

Digunakan untuk membaca suatu string.

f. Fungsi *puts()*;

Digunakan untuk mencetak string.

g. Continuation Character (\);

Digunakan untuk menghubungkan beberapa baris string konstan.

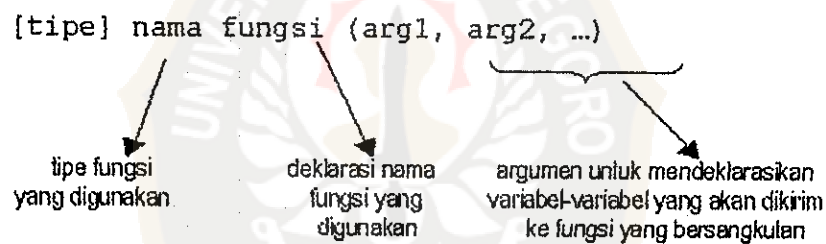
2.3.3. Struktur Program

2.3.3.1. Fungsi

Fungsi merupakan blok kode yang dirancang untuk melaksanakan tugas khusus, dengan tujuan:

- program menjadi terstruktur, sehingga mudah dipahami dan dikembangkan.
- Dapat mengurangi pengulangan (duplikasi) kode.

Bentuk deklarasi fungsi:



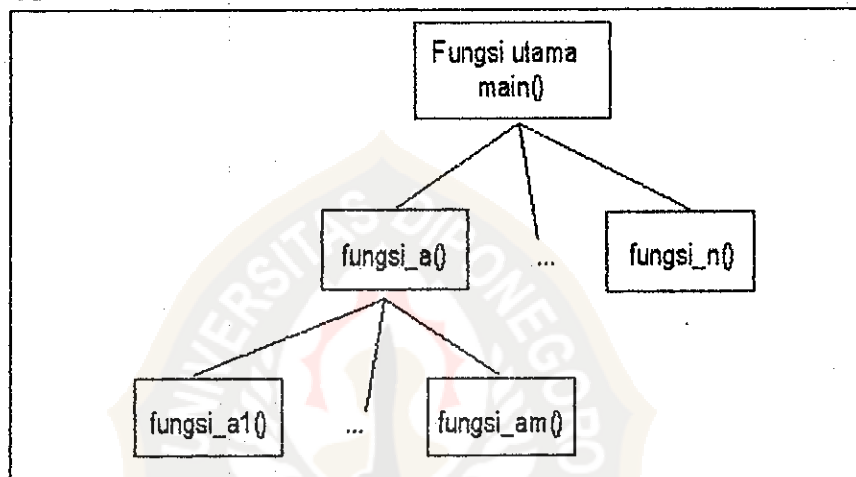
Jika diinginkan suatu fungsi tidak mengirimkan hasil (hanya melakukan suatu proses) maka tipe fungsi diisi dengan *void*.

2.3.3.2. Pengarah Praprosesor #include

#include merupakan salah satu jenis pengarah praprosesor, yang digunakan untuk membaca file judul (*header file*), yaitu file yang diantaranya berisi deklarasi fungsi dan definisi konstanta. File-file judul mempunyai ciri-ciri yaitu namanya diakhiri dengan extension .h, misalnya `#include <stdio.h>`, `#include <math.h>`, dan lain-lain.

2.3.3.3. Pemrograman Terstruktur

Suatu program yang terstruktur dikembangkan dengan menggunakan konsep *top-down design* (rancang atas-bawah). Pada C, suatu program disusun dari sejumlah fungsi dengan tugas tertentu. Selanjutnya masing-masing fungsi bisa dipecah-pecah lagi menjadi fungsi yang lebih kecil, sehingga akan memudahkan pencarian kesalahan atau pengembangannya.



Gambar 2.11
Ilustrasi program terstruktur

Contoh struktur program:

```

#include <stdio.h>
void ubah(int);
main()
{
    int x=27;
    .....
    ubah(x);
    printf("x=%d\n", x);
}
void ubah(int y)
{
    ==85;
}
  
```

→ pengarah ke header file
 → deklarasi fungsi *ubah()*
 } fungsi utama *main()*
 } fungsi *ubah()*

2.3.4.. PEMROGRAMAN GRAFIS

Untuk memperindah tampilan pada layar, pemrograman grafis ditambahkan ke dalam program musikal dengan menggunakan fasilitas-fasilitas yang sudah disediakan dalam bahasa C.

2.3.4.1. Pendukung Grafik

Agar program grafis dapat dijalankan, harus tersedia file-file pendukung grafik, yaitu:

1. Graphics.h : berisi deklarasi variabel eksternal, definisi tipe, dan deklarasi fungsi grafik. File ini digunakan oleh setiap program grafik.
2. *.BGI : file dengan ekstensi BGI berisi data pengendali grafik. Misal untuk adapter VGA nama file-nya adalah VGA.BGI.
- 3.*.CHR : berisi data huruf (*fonts*). Misalnya untuk model huruf gothic nama file-nya adalah GOTHIC-.CHR.

2.3.4.2. Perintah Dasar Pemrograman Grafik

Ada 4 buah fungsi yang merupakan perintah dasar dalam pemrograman grafis dengan bahasa C, yaitu:

1. *Initgraph()*, digunakan untuk membuka sistem grafik.

Deklarasinya :

```
void far initgraph (int far kendaligraph,
                   int far modegraph,
                   int far path);
```

- *kendaligraph* adalah pointer yang menunjuk ke data yang bertipe integer untuk menunjukkan pengendali (adapter) grafik yang digunakan.. Untuk adapter CGA diisi dengan 1, adapter EGA diisi dengan 3, sedangkan adapter VGA diisi dengan 9.

- *modegraph* dapat diisi dengan nilai-nilai yang tertera dalam tabel berikut:

Tabel 2.5
Konstanta simbolis modegrafik

Pengendali Grafik	Mode Grafik	Nilai	Kolom: x Baris	Jumlah Warna
CGA	CGAC0	0	320 X 200	4 warna
	CGAC1	1	320 X 200	4 warna
	CGAC2	2	320 X 200	4 warna
	CGAC3	3	320 X 200	4 warna
	CGAHI	4	640 X 200	2 warna
EGA	EGALO	0	640 X 200	16 warna
	EGAHI	1	640 X 350	16 warna
VGA	VGALO	0	640 X 200	16 warna
	VGAMED	1	640 X 350	16 warna
	VGAHI	2	640 X 480	16 warna

- *path* menyatakan nama direktori tempat file *.BGI berada. Jika file *.BGI berada pada direktori yang sedang aktif (*Current Directori*), string yang ditunjuk path cukup diisi dengan string kosong ("").

2. *graphresult()* dipakai untuk mengetahui kesalahan yang mungkin timbul setelah suatu operasi grafik dilakukan.

Deklarasinya :

```
Int var graphresult (void);
```

3. *grapherrormsg()* dipakai untuk mendapatkan pesan kesalahan grafik.

Deklarasinya :

```
Int var graphresult (int kode_salah)
```

4. *closegraph* digunakan untuk menutup kembali sistem grafik dan mengembalikannya ke dalam mode layar yang aktif sebelum pemanggilan *initgraph()*. Dengan demikian maka memori yang tadi digunakan untuk menyimpan data pengendali grafik akan dibebaskan kembali.

2.3.4.3. Fungsi-Fungsi dalam Pemrograman Grafik

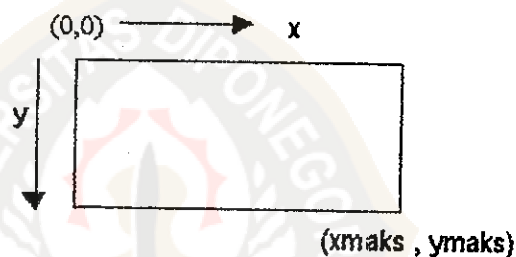
2.3.4.3.1. Mengakses Piksel

Ada 2 macam fungsi untuk mengakses piksel, yaitu:

- *putpixel()*, digunakan untuk menggambar piksel (titik) pada koordinat (x,y) dalam sistem koordinat layar, disertai dengan kode warna yang dikehendaki.

Deklarasinya:

```
Void far putpixel(int x, int y, int wrn);
```



Gambar 2.12
Sistem Koordinat Layar

- *getpixel()*, digunakan untuk memperoleh kode warna dari suatu piksel (titik) pada koordinat (x,y).

Deklarasinya:

```
Unsigned far getpixel(int x, int y);
```

2.3.4.3.2. Memindah Penunjuk Gambar

Fungsi untuk memindah penunjuk gambar adalah *moveto()*.

Fungsi *moveto()* digunakan untuk memindah penunjuk gambar ke posisi (x,y).

Deklarasinya:

```
Void far moveto(int x, int y);
```

2.3.4.3.3. Membuat dan Mengatur Bentuk Garis

Fungsi-fungsi yang bisa digunakan untuk membuat dan mengatur bentuk garis adalah:

- `line()`, digunakan untuk menggambar garis dari posisi (x_1, y_1) ke posisi (x_2, y_2) . Sedangkan posisi penunjuk gambar tidak berubah.

Deklarasinya:

```
void far line(int x1, int y1, int x2, int y2);
```

- `setlinestyle()`, digunakan untuk mengatur bentuk dan ketebalan garis

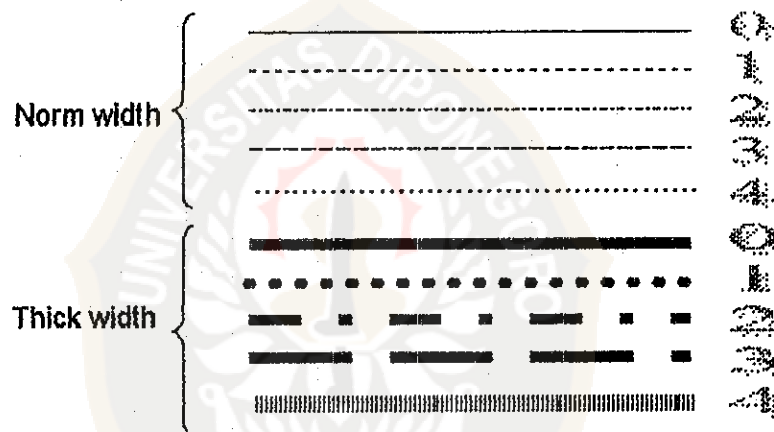
Deklarasinya:

```
void far setlinestyle(int bentuk, unsigned pola,
                     int tebal);
```

- bentuk bisa diisi dengan nilai 0, 1, 2, 3, atau 4, nilai yang diberikan menyatakan bentuk garis yang diinginkan, seperti terlihat dalam gambar berikut:
- pola diisi dengan nilai yang menyatakan pola yang didefinisikan oleh pemrogram. Nilai yang ada pada parameter ini hanya berpengaruh jika bentuk diisi dengan nol (0). Dan jika bentuk diisi dengan nilai 4 pemakai bisa mendefinisikan sendiri bentuk garis yang diinginkan dengan membuat pola yang terdiri atas 16 bit yang berisi 0 atau 1. Angka 1 berarti menggambar piksel, sedangkan angka 0 berarti tidak menggambar piksel.
 Contohnya, untuk pola 01010101010101 (0XAAAA) menyatakan bentuk garis putus-putus (spasi titik spasi).
- tebal bisa diisi dengan nilai 1 untuk menyatakan garis tipis, atau 3 untuk menyatakan garis tebal.

Tabel 2.6
Konstanta pengatur bentuk dan tebal garis

Konstanta	Nilai	Keterangan
Memilih bentuk garis		
Solid_line	0	didefinisikan oleh user
Dotted_line	1	
Center_line	2	
Dashed_line	3	
User_bit_line	4	
Memilih tebal garis		
Norm_width	1	
Thick_width	3	



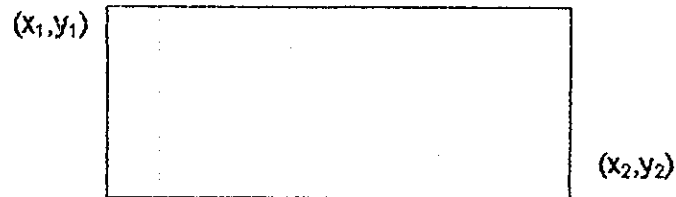
Gambar: 2.13
Macam-macam bentuk garis yang bisa dibuat.

2.3.4.3.4. Membuat Kotak

Fungsi yang digunakan untuk membuat kotak dari posisi (x_1, y_1) yang merupakan pojok kiri-atas kotak ke posisi (x_2, y_2) yang merupakan pojok kanan-bawah kotak, adalah *rectangle()*.

Deklarasinya:

```
Void far rectangle(int x1, int y1, int x2, int y2);
```

Gambar 2.14
Menentukan koordinat kotak

2.3.4.3.5. Mengatur Warna Gambar dan Latar Belakang

Untuk mengatur warna gambar yang berbentuk garis, kotak, lingkaran, atau poligon, fungsi yang digunakan adalah `setcolor()`. Fungsi `setcolor()` dituliskan sebelum menggambar obyek. Deklarasinya:

```
void far setcolor(int warna);
```

Sedangkan untuk mengatur warna latar belakang, digunakan fungsi `setbcolor()`. Seperti halnya fungsi `setcolor`, fungsi `setbcolor` ditulis sebelum penulisan perintah menggambar obyek.

Deklarasinya:

```
void far setbcolor(int warna);
```

Baik untuk fungsi `setcolor()` atau `setbcolor()`, warna diisi dengan nilai yang mewakili suatu warna seperti ditunjukkan dalam tabel berikut:

Tabel 2.7
Nilai konstanta warna

Konstanta	Nilai	Warna	Konstanta	Nilai	Warna
Black	0	Hitam	Darkgray	8	Abu-abu tua
Blue	1	Biru	Lightblue	9	Biru cerah
Green	2	Hijau	Lightgreen	10	Hijau cerah
Cyan	3	Sian	Lightcyan	11	Sian cerah
Red	4	Merah	Lightred	12	Merah cerah
Magenta	5	Magenta	Lightmagenta	13	Magenta crh.
Brown	6	Coklat	Yellow	14	Kuning
Lightgray	7	Abu-abu	White	15	Putih

2.3.4.3.6. Membuat Arsiran atau Blok pada Gambar

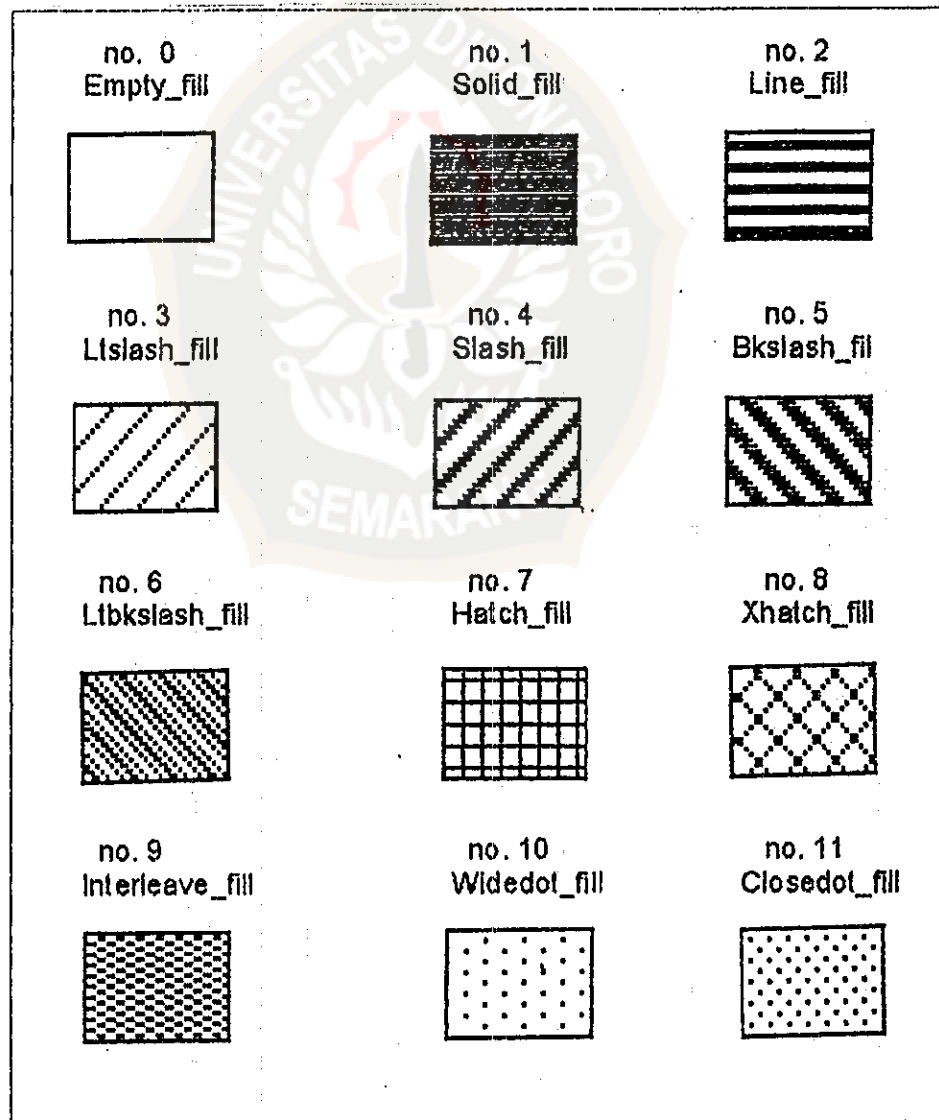
Untuk mengarsir atau memblok gambar, ada beberapa fungsi yang bisa digunakan, diantaranya:

- `setfillstyle()`, digunakan untuk membuat arsiran pada suatu obyek tertentu.

Deklarasinya:

```
void far setfillstyle(int arsir, int warna);
```

arsir diisi dengan suatu nilai yang menunjukkan pola arsiran yang diinginkan, seperti yang ada pada gambar berikut:



Gambar. 2.15.
Pola arsiran dalam Turbo C

- *bar()*, digunakan untuk membuat bentuk suatu kotak dengan pola arsiran yang sudah dideklarasikan terlebih dahulu melalui fungsi *setfillstyle()*.

Deklarasinya:

```
void far bar(int x1, int y1, int x2, int y2);
```

2.3.4.3.7. Menampilkan Teks pada Mode Grafik

Untuk menampilkan teks pada grafik, ada beberapa fungsi yang bisa digunakan, diantaranya:

- *outtextxy()*, digunakan untuk menampilkan string ke posisi (x,y).

Deklarasinya:

```
void far outtextxy(int x, int y, char string);
```

- *settextstyle()*, digunakan untuk memilih jenis huruf (font), arah tampilan, serta ukuran dari string yang akan ditampilkan melalui *outtextxy()*.

Deklarasinya:

```
void far settextstyle(int font, int arah,  
                    int ukuran);
```

- Font dapat diisi berdasarkan nilai konstanta seperti yang

terlihat pada tabel berikut:

Tabel 2.8.
Nilai konstanta font

Konstanta	Nilai	File yang diperlukan
Default_font	0	-
Triplex_font	1	Trip.chr
Small_font	2	Litt.chr
Sansserif_font	3	Sans.chr
Gothic_font	4	Goth.chr

- Arah menyatakan arah tulisan yang ingin ditampilkan. Jika diisi nilai 0 maka arah tulisannya mendatar, jika diisi nilai 1 maka arah tulisannya vertikal.
- Ukuran menyatakan ukuran tulisan yang ingin ditampilkan. Jika diisi nilai 0 (kecuali untuk `default_font`) ukuran teks dapat diatur melalui fungsi `setusercharsize()`, dan jika tidak terdapat pada program, maka ukuran bawaan (normal) dari turbo C yang akan ditampilkan.

