

## Bab 6.

### FUNGSI-FUNGSI DALAM BASIC

Yang dimaksud dengan statement dan fungsi pustaka adalah semacam subroutine yang sudah disediakan oleh sistem operasi. Pada bahasa Basic terdapat fungsi aritmatika, fungsi string, fungsi suara dan lain-lain.

#### 6.1. Fungsi Aritmatika

- **Fungsi ABS**

Tujuan : mendapatkan harga mutlak dari suatu ungkapan.

Bentuk umum : ABS(X).

##### Program 6.1.

```
cls
A=10
B=3
C=5
? B^2-4*A*C
? abs(B^2-4*A*C)
```

Output :

-191	→	nilai asli
191	→	nilai setelah diabsolutkan

- **Fungsi ATN**

Tujuan : menghitung harga arctangent dari (X) dalam radian. Hasilnya berupa bilangan presisi tunggal dan terletak antara  $-\pi/2$  sampai  $\pi/2$  (1.57143)

Bentuk umum :

? ATN (X)

##### Program 6.2.

```
Print "I      atn(I) "
for I = 0 TO 2 STEP 0.25
  PRINT I, ATN(I)
NEXT
```

Output :

I	ATN(I)
0	0
.25	.2449786631268642
.5	.4636476090008061
.75	.6435011087932844
1	.7853981633974483
1.25	.896055384571344
1.5	.9827937232473291
1.75	1.051650212548374
2	1.107148717794091

- **Fungsi CDBL**

Tujuan : konversi atau mengubah suatu bilangan ke presisi ganda

Bentuk umum :

CDBL (X)

Contoh :

**PRINT CDBL(454.67)**

Output :

**454.670013427734**

- **Fungsi CINT**

Tujuan : konversi suatu nilai X menjadi bilangan integer dengan pembulatan ke atas untuk bagian pecahan/desimal yang lebih besar atau sama dengan 0.5

Bentuk umum :

CINT (X)

### **Program 6.3.**

```
INPUT "Masukkan sembarang angka : ";Angka
? "Pembulatan angka = ";CINT (Angka)
x= 22/7
? x
? cint(x)
```

Output :

**Masukkan sembarang angka : 12.7**  
**Pembulatan angka = 13**  
**3.142857074737549**  
**3**

- **Fungsi COS**

Tujuan : menghitung harga cosinus (X) dalam radian, dengan ketetapan presisi tunggal. Untuk menghitung harga cosinus sudut dalam besaran derajat, (X) harus dikalikan dengan (22/7/180). Jadi bila diinginkan cosinus 60 maka harus ditulis COS (60\*(22/7/180))

Bentuk umum :

COS (X)

Keterangan : X dalam radian

**Program 6.4.**

```
cls
? "satuan radian"
? "i    cos(i)"
for i = 0 to 360 step 60
  print using "#####";i,using "###.##";cos(i)
next
? "satuan derajat"
? "i    cos(i)"
derajat= 3.14/180
for i = 0 to 360 step 60
  print using "#####";i,using "###.##";cos(i*derajat)
Next
```

**Output :**

```
satuan radian
i cos(i)
0 1.00
60 -0.95
120 0.81
180 -0.60
240 0.33
300 -0.02
360 -0.28
```

```
satuan derajat
i cos(i)
```

**0 1.00**  
**60 0.50**  
**120 -0.50**  
**180 -1.00**  
**240 -0.50**  
**300 0.50**  
**360 1.00**

- **Fungsi CSNG**

Tujuan : konversi bilangan (X) menjadi presisi tunggal

Bentuk umum :

CSNG (X)

### **Program 6.5.**

```
cls
'konversi dari single ke double precision
? "Menghitung 10/3"
a!=10/3
? "Tipe single : ";a!
b# = cdbl(a!)
? "diubah ke tipe double : ";b#
'konversi dari double ke single precision
c#=10/3
? "Tipe double : ";c#
d! = csng(c#)
? "diubah ke single : ";d!
```

Output :

```
Menghitung 10/3
Tipe single : 3.333333253860474
diubah ke tipe double : 3.333333253860474
Tipe double : 3.333333333333333
diubah ke single : 3.333333253860474
```

Terlihat pada program di atas, pembagian 10/3 jika disimpan pada variabel bertipe single precision hasilnya kurang valid jika dibandingkan dengan variabel bertipe double precision. Jika hasil pembagian 10/3 disimpan ke variabel bertipe single kemudian diubah ke variabel bertipe double dengan

menggunakan CDBL maka tidak akan membuat hasil perhitungan menjadi lebih teliti.

- **Fungsi EXP**

Tujuan : menghitung harga pemangkatan bilangan e dengan X, dalam besaran presisi tunggal.

Bentuk umum :

EXP (X)

Keterangan :

Bilangan e adalah 2.71828 ..., ialah bilangan yang dipangkatkan dan X adalah bilangan pangkat; X tidak boleh lebih dari 87.3365. Bila X berlebihan akan ada pesan kesalahan "overflow"

**Program 6.6.**

```
INPUT X
PRINT EXP (X-1)
PRINT EXP (X)
```

Output :

```
? 2
2.71828
7.38906
```

- **Fungsi FIX**

Tujuan : untuk mendapatkan bilangan bulat berikut tandanya dengan memotong bagian pecahannya atau desimalnya.

Bentuk umum :

FIX (X)

Hasil dari FIX(X) sama dengan jika diberi perintah  $SGN(X)*INT(ABS(X))$

**Program 6.7.**

```
INPUT X
PRINT SGN(X)*INT(ABS(X))
```

```
PRINT FIX(X)
PRINT FIX(3.1)
PRINT FIX(2.999)
```

Output :

```
? -345.72
-345
-345
3
2
```

- **Fungsi INT**

Tujuan : mendapatkan bilangan integer (bulat) dengan pembulatan ke bawah dari bagian pecahannya.

Bentuk umum :

INT (X)

**Program 6.8.**

```
PRINT INT (15.01)
PRINT INT (15.99999)
```

Output :

```
15
15
```

- **Fungsi LOG**

Tujuan : menghitung logaritma bilangan X dengan bilangan dasar e. Bilangan X harus lebih besar dari 0, sedang besarnya bilangan e adalah 2.7182818284590...

Digunakan untuk menyederhanakan perhitungan ilmiah.

Bentuk umum : LOG (X)

**Program 6.9.**

```
PRINT LOG (10)
PRINT LOG (EXP(10))
```

Output :

**2.30259**  
**10**

- **Fungsi MOD**

Tujuan : mencari sisa dari hasil bagi.

Bentuk umum :

$X \text{ MOD } Y$

Keterangan :

Hasilnya adalah sisa dari X dibagi dengan Y.

**Program 6.10.**

```
INPUT "X = ";X
INPUT "Y = ";Y
PRINT "Sisa dari X dibagi Y = "; X MOD Y
```

Output :

**X = 10**

**Y = 3**

**Sisa dari X dibagi Y = 1**

- **Perintah RANDOMIZE**

Tujuan : menghidupkan pembentuk bilangan acak

Bentuk umum :

$\text{RANDOMIZE } [<\text{ungkapan}>]$

Keterangan :

<ungkapan> menentukan luas penyebaran dari nilai acak yang dihasilkan oleh generator. Bila ungkapan tidak ditulis, maka BASIC akan menanyakan : "Random Number Seed (-32768+32767)?" Dan jika inipun tidak ada maka perhitungan dari instruksi RND akan menghasilkan urutan bilangan acak yang tidak berubah, setiap kali program dijalankan.

Untuk menghasilkan bilangan acak yang betul-betul acak, maka fungsi RANDOMIZE harus ditempatkan pada permulaan program dengan mengganti argumennya setiap kali program dieksekusi.

**Program 6.11.**

```
FOR J=5 TO 10 STEP 5
  RANDOMIZE (J)
  FOR I=1 TO 3
    PRINT RND;
  NEXT I
  PRINT
NEXT J
```

Output :

```
.76220703125 .3424150957725942 .9162242482416332
.77197265625 .3912432207725942 .1603648732416332
```

- **Fungsi RND**

Tujuan : mengambil nilai acak antara 0 dan 1.

Bentuk umum : RND

**Program 6.12.**

```
FOR I=1 TO 5
  PRINT INT(RND*100);
NEXT
```

Output :

```
96 20 51 13 87
```

- **Fungsi SGN**

Tujuan : mendapatkan nilai 1, 0 atau -1 untuk  $X > 0$ ,  $X = 0$  atau  $X < 0$

Bentuk umum :

SGN (X)

Keterangan :

Bila  $X > 0$  maka  $SGN(X) = 1$

$X = 0$  maka  $SGN(X) = 0$

$X < 0$  maka  $SGN(X) = -1$

**Program 6.13.**

```
INPUT "Masukkan bilangan : ";X
```



```
IF SGN(X)=0 THEN
  PRINT "Bilangan = 0"
ELSEIF SGN (X)<0 THEN
  PRINT "Bilangan < 0"
ELSEIF SGN (X)>0 THEN
  PRINT "Bilangan > 0"
END IF
```

Output :

Masukkan bilangan : -23  
Bilangan < 0

**Program 6.14.**

```
REM Contoh program menggunakan fungsi SGN
Input a,b,c
D=b^2-4*a*c
SELECT CASE sgn(D)
  case 1
    ? "Ada 2 akar real"
  case 0
    ? "Akarnya kembar"
  case -1
    ? "Akarnya imajiner"
END SELECT
```

Output :

? 2,3,4  
Akarnya imajiner

- **Fungsi SIN**

Tujuan : menghitung harga sinus (X) dalam radian, dengan ketetapan presisi tunggal. Untuk menghitung harga sinus sudut dalam besaran derajat, (X) harus dikalikan dengan (22/7/180). Jadi bila diinginkan sinus 60 maka harus ditulis SIN (60\*(22/7/180))

Bentuk umum :

SIN (X)

Keterangan : X dalam radian

**Program 6.15.**

```
REM Contoh penggunaan fungsi SIN
sudut = 180/3.14
? "I    sin(I)"
for i = 0 to 360 step 30
  ? using " ###";i, using " ###.##";sin(i/sudut)
next I
```

**Output :**

```
 I  sin(I)
 0  0.00
30  0.50
60  0.87
90  1.00
120 0.87
150 0.50
180 0.00
210 -0.50
240 -0.86
270 -1.00
300 -0.87
330 -0.50
360 -0.00
```

• **Fungsi SQR**

Tujuan : untuk mencari akar pangkat dua dari suatu bilangan

Bentuk umum :

SQR (X)

Keterangan :

Mencari akar pangkat dua dari bilangan X. Fungsinya sama dengan  $X^{.5}$

**Program 6.16.**

```
INPUT "Bilangan = ";X
PRINT "Akar pangkat dua dari bilangan = ";SQR (X)
```

Output :

**Bilangan = 4**

## Akar pangkat dua dari bilangan = 2

- **Fungsi TAN**

Tujuan : menghitung harga tangent X dalam radian dengan ketetapan presisi tunggal.

### Program 6.17.

```
derajat=45
radian=180/3.14
? tan(derajat)
? tan(derajat/radian)
```

Output :

```
1.619775190543862 hasilnya salah
.9992039960893519 hasil yang benar (mendekati 1)
```

Keterangan :

Sama dengan fungsi sinus dan cosinus, untuk menghitung harga tangent dalam besaran derajat, nilai X harus dikalikan dulu dengan (22/7/180)

### Program 6.18.

```
INPUT "Besarnya sudut : ";X
PRINT "Nilai tangent = ";TAN (X*(22/7/180))
```

Output :

```
Besarnya sudut : 45
Nilai tangent = 1.00063244458459
```

## 6.2. Fungsi STRING

- **Fungsi ASC**

Tujuan : untuk mendapatkan kode ASCII dari suatu data bertipe karakter.

Bentuk umum :

ASC (X\$)

**Program 6.19.**

```
input "Masukkan karakter yang akan dicari kode ASCII-nya : ";A$
PRINT ASC (A$)
```

Output :

```
Masukkan karakter yang akan dicari kode ASCII-nya : B
66
Masukkan karakter yang akan dicari kode ASCII-nya : b
98
```

- **Fungsi CHR\$**

Tujuan : untuk mendapatkan karakter dari suatu kode ASCII (mempunyai fungsi yang berkebalikan dengan ASC).

Bentuk umum :

CHR\$ (I)

Keterangan : nilai I dari 0 sampai 255.

**Program 6.20.**

```
INPUT (X)
PRINT CHR$ (X)
```

Output :

```
? 66
B
? 98
b
```

- **Fungsi HEX\$**

Tujuan : konversi kode ASCII desimal menjadi kode hexadesimal

Bentuk umum :

HEX\$ (X)

Keterangan :

X dalam desimal.

**Program 6.21.**

```
INPUT "Desimal = ";X
```

```
PRINT "Hexadesimal = ";HEX$ (X)
```

Output :

```
Desimal = 15  
Hexadesimal = F
```

- **Fungsi INKEY\$**

Tujuan : mendapatkan satu karakter dari keyboard tanpa ada echo yang tampak di layar monitor.

Bentuk umum :

```
INKEY$
```

- **Fungsi INSTR**

Tujuan : mencari karakter pertama dari karakter kedua dan memberitahu letak karakter pertama tersebut dari tepi kiri terhadap susunan karakter kedua.

Bentuk umum :

```
INSTR ([I],X$,Y$)
```

Keterangan :

X\$ = karakter pertama

Y\$ = karakter kedua

Harga [I] berkisar dari 1-255

### **Program 6.22.**

```
X$="ABCDEFGBCGH"  
Y$="B"  
PRINT "Letak karakter B yang pertama dijumpai pada susunan  
ABCDEFGBCGH = ";_  
INSTR (X$,Y$)  
PRINT "Letak karakter B yang ditemukan setelah 4 karakter  
pertama = ";_  
INSTR (4,X$,Y$)
```

Output :

```
Letak karakter B yang pertama dijumpai pada susunan  
ABCDEFGBCGH = 2
```

**Letak karakter B yang ditemukan setelah 4 karakter pertama = 7**

- **Fungsi LEFT\$**

Tujuan : mengambil sejumlah karakter dari kiri dari suatu susunan karakter.

Bentuk umum :

LEFT (X\$,I)

Keterangan :

I = jumlah karakter yang diambil

X\$=susunan karakter yang diambil

**Program 6.23.**

```
CLS
X$="DAISY"
FOR J=1 TO LEN(X$)
PRINT LEFT$(X$,J)
NEXT J
FOR I=LEN(X$)-1 TO 1 STEP -1
PRINT LEFT$(X$,I)
NEXT I
```

Output :

```
D
DA
DAI
DAIS
DAISY
DAIS
DAI
DA
D
```

- **Fungsi LEN**

Tujuan : menghitung panjang dari karakter string termasuk spasi

Bentuk umum :

LEN (X\$)

Keterangan :

Mencari panjang dari susunan karakter X\$

**Program 6.24.**

```
INPUT "Nama (maks. 10 digit) : ";Nama$
PRINT "Panjang nama = ";LEN(Nama$);" digit"
```

Output :

**Nama (maks. 10 digit) : DAISY**  
**Panjang nama = 5 digit**

• **Fungsi MID\$**

Tujuan :

- Mengambil karakter string sepanjang J karakter dari susunan X\$ dimulai dari karakter ke-I.
- Mengganti sebagian dari suatu string dengan string yang lain.

Bentuk umum :

```
MID$ (X$,I,[,J])
MID$(<String1>,n,[,m])=<String2>
```

Keterangan :

- Bila J tidak disebutkan, maka akan diberi nilai 1.
- Karakter dari string1 dimulai dari karakter yang ke-n, akan diganti dengan karakter dari string2 sepanjang m. Bila m tidak disebutkan, seluruh karakter string2 akan digunakan. Tetapi bila string2 lebih panjang maka ia akan dipotong sepanjang string2 saja.

**Program 6.25.**

```
A$="SELAMAT "
B$="PAGI SIANG MALAM"
PRINT A$;MID$(B$,1,4)
PRINT A$;MID$(B$,6,5)
PRINT A$;MID$(B$,12,5)
```

Output :

```
SELAMAT PAGI
SELAMAT SIANG
SELAMAT MALAM
```

**Program 6.26.**

```
AS="JAKARTA, INDONESIA"
MID$(AS,10)="JAWA"
PRINT AS
```

Output :

```
JAKARTA, JAWANESIA
```

**Program 6.27.**

```
cls
input "Masukkan sembarang kata : ",kata$
for i = len(kata$) to 1 step -1
  ? mid$(kata$,i,1);
next i
```

**Output :**

```
Masukkan sembarang kata : Surabaya kota Pahlawan
nawalhaP atok ayabaruS
```

- **Fungsi OCT\$**

Tujuan : konversi kode ASCII desimal menjadi kode oktal (basis delapan)

Bentuk umum :

OCT\$(X)

Keterangan : X dalam desimal.

**Program 29.**

```
cls
? "Desimal","Octal","Hexa"
for i = 10 to 25 step 5
  ? i,oct$(i),hex$(i)
next
```

**Output :**



<b>Desimal</b>	<b>Octal</b>	<b>Hexa</b>
10	12	A
15	17	F
20	24	14
25	31	19

- **Fungsi RIGHTS**

Tujuan : mengambil sejumlah karakter dari kanan dari suatu susunan karakter.

Bentuk umum :

RIGHT (X\$,I)

Keterangan :

I = jumlah karakter yang diambil

X\$=susunan karakter yang diambil

**Program 6.30.**

```
A$="SELAMAT"  
B$="PAGI SIANG"  
C$="SORE MALAM"  
PRINT A$;RIGHT$(B$,6)  
PRINT A$;RIGHT$(C$,6)
```

Output :

```
SELAMAT SIANG  
SELAMAT MALAM
```

- **Fungsi STR\$**

Tujuan : mengubah besaran numerik menjadi string

Bentuk umum :

STR\$ (X)

**Program 6.31.**

```
A=12345  
B=54321  
A$=STR$(A)  
B$=STR$(B)  
PRINT A+B  
PRINT A$+B$
```

Output :

66666  
12345 54321

- **Fungsi STRING\$**

Tujuan : mendapatkan sejumlah string dari suatu susunan karakter string atau untuk mendapatkan string yang sama dengan kode ASCII tertentu.

Bentuk umum :

STRING\$ (I,J)

STRING\$ (I,X\$)

Keterangan :

I = jumlah string yang akan diperoleh

J = kode ASCII

X\$=susunan karakter yang diambil

**Program 6.32.**

```
A$="G"+STRING$(5,"O")+"AL"  
PRINT A$  
PRINT STRING$(1,A$)+"OAL"
```

Output :

```
GOOOOOAL  
GOAL
```

- **Fungsi VAL**

Tujuan : mengubah besaran string menjadi numerik

Bentuk umum :

VAL (X\$)

**Program 6.33.**

```
A$="12345 "  
B$="54321"  
A=VAL(A$)  
B=VAL(B$)  
PRINT A$+B$  
PRINT A+B
```

Output :

12345 54321  
66666

### 6.3. Fungsi Layar

- **Perintah Locate**

Tujuan : menempatkan pada posisi koordinat (x,y) di layar.

Bentuk umum :

Locate Y,X

Keterangan :

X = koordinat sumbu x

Y = koordinat sumbu y

#### Program 6.34.

```
INPUT "Masukkan nama : ";Nama$  
PRINT "Ditampilkan di tengah layar : "  
LOCATE 12,(80-LEN(Nama$))/2  
PRINT Nama$
```

Output :

Masukkan nama : **Turbo Basic**  
Ditampilkan di tengah layar :

**Turbo Basic**

- **Fungsi POS**

Tujuan : untuk mengetahui posisi kursor.

Bentuk umum :

POS (X)

#### Program 6.35.

```
FOR I=1 TO 5  
FOR J=65 TO 255  
IF POS (X) >10 THEN  
LOCATE I+1,1  
J=255
```

```
ELSE
  PRINT CHR$(J);
END IF
NEXT J
NEXT I
```

Output :

```
ABCDEFGHIJ
ABCDEFGHIJ
ABCDEFGHIJ
ABCDEFGHIJ
ABCDEFGHIJ
```

- **Fungsi Color**

Tujuan : untuk menentukan warna tertentu dalam mode grafik

Bentuk umum :

COLOR = <nomor warna>

0	Hitam	8	Abu-abu
1	Biru	9	Biru terang
2	Hijau	10	Hijau terang
3	Cyan	11	Cyan terang
4	Merah	12	Merah terang
5	Magenta	13	Magenta terang
6	Coklat	14	Kuning
7	Putih	15	Putih terang

Keterangan :

COLOR digunakan untuk resolusi rendah, HCOLOR digunakan untuk resolusi tinggi. Nomor warna dari resolusi rendah dan tinggi berbeda.

- **Fungsi SPACES**

Tujuan : mendapatkan sejumlah karakter string kosong dihitung dari tepi kiri.

Bentuk umum :

SPACES (X)

Keterangan :

Nilai X dibulatkan menjadi integer dari 1 sampai 255.

#### **Program 6.36.**

```
FOR I=1 TO 5
  PRINT SPACES(I);
  PRINT I;" BASIC"
NEXT I
```

Output :

```
1 BASIC
2 BASIC
3 BASIC
4 BASIC
5 BASIC
```

#### **6.4. Fungsi untuk Pemrograman Suara**

Ada 3 (tiga) perintah khusus dalam bahasa Basic untuk menyusun pemrograman musik (suara) yaitu : BEEP, SOUND dan PLAY.

- **Fungsi BEEP**

Tujuan : digunakan untuk membunyikan speaker dengan nada dan waktu tertentu.

Efek yang sama untuk perintah BEEP, dapat dicapai dengan perintah CHR\$(7). Nada dan suara yang ditimbulkan sama; sedangkan perbedaannya adalah dengan menggunakan PRINT CHR\$(7) akan ditampilkan satu baris kosong di monitor.

#### **Program 6.37.**

```
INPUT A
BEEP
INPUT B
PRINT CHR$(7)
```

- **Fungsi SOUND**

Tujuan : untuk menimbulkan suara dengan frekuensi dan durasi tertentu.

Bentuk umum :

SOUND frekuensi, durasi.

Keterangan :

frekuensi yang diijinkan berkisar antara 37 sampai 32767, sedangkan durasi yang diperkenankan bernilai antara 0 sampai 65535.

Pada perintah BEEP dan CH\$(7), frekuensi suara yang dibunyikan sebesar 800 hertz dengan durasi seperempat detik. Jika perintah tersebut dilakukan oleh fungsi SOUND, maka hasil yang diperoleh dengan menghitung durasi sebesar 18.2 per satu detik. Jadi untuk membunyikan frekuensi 800 Hertz selama  $\frac{1}{4}$  detik diperlukan durasi :

$$\frac{1}{4} * 18.2 = 4.55$$

Penyusunan program suara dengan SOUND agak rumit karena dituntut pengetahuan nada dan irama secara baik. Walaupun demikian, tabel berikut ini dapat dipakai sebagai pedoman dalam membunyikan nada dengan frekuensi yang sesuai.

Nada	Frekuensi
C	130.810
D	146.830
E	164.810
F	174.610
G	196.000
A	220.000
B	246.960
C	261.630
D	293.660
E	329.630
F	349.230
G	392.000
A	440.000
B	493.880
C	523.250

**Program 6.38.**

```

FOR I=440 TO 1000 STEP 5
SOUND K,0.5
NEXT K
FOR K=1000 TO 440 STEP -5

```

<b>SOUND K,0.5</b> <b>NEXT K</b>
-------------------------------------

- **Fungsi PLAY**

Tujuan : sebagai perintah pemrograman musik dengan parameter string yang berisi kombinasi karakter dan variabel untuk nada yang akan dibunyikan.

Bentuk umum :

PLAY string

Keterangan :

- string dapat berisikan karakter-karakter sebagai berikut :

A	B	C	D	E	F
G	#	+	-	On	Nn
Ln	Nn	.	Pn	Tn	MF
MB	MN	ML	MS	X	

Setiap karakter mempunyai efek tertentu dengan n yang berisikan nilai parameter tertentu.

- **Karakter A sampai dengan G**

Setiap karakter menyatakan satu not. Contohnya dapat dilihat pada tabel berikut. Dimana tabel ini berlaku untuk kunci nada C.

Karakter	Suaranya
C	Do
D	Re
E	Mi
F	Fa
G	Sol
A	La
B	Si
C	Do

Jika kunci nadanya diubah menjadi, misal F#, maka isi seluruh tabel tersebut ikut berubah. Tabel untuk kunci nada F# sebagai berikut :

Karakter	Suaranya
F#	Do
Ab	Re

Bb	Mi
B	Fa
Db	Sol
Eb	La
F	Si
F#	Do

Penulisan not setengah nada dinyatakan dalam simbol #.

□ **Karakter On**

Dipakai untuk menentukan oktaf yang akan dipakai. Ada 7 (tujuh) oktaf yang dapat dipilih dengan nomor dari 0 sampai dengan 6. Oktaf standar yang dipakai adalah 4.

□ **Karakter Nn**

Dipakai untuk memainkan nada n. Dengan n bernilai antara 0 sampai 84. Nn dipakai untuk mempersingkat penulisan tangga nada (A sampai dengan G) serta penentuan oktafnya melalui On.

□ **Karakter Ln**

Dipakai untuk menentukan panjang dari nada berikutnya. Panjang nada ditentukan oleh rumus :  $1/n$  dengan n bernilai antara 1 sampai 64.

Contoh :

Karakter	Panjangnya
L1	satu nada
L2	setengah nada
L4	seperempat nada
L64	seperenampuluhempat nada

□ **Karakter Pn**

Dipakai untuk menentukan panjang waktu istirahat. Nilainya berkisar antara 1 sampai dengan 64. Cara penentuan nilai Pn adalah sama dengan cara penentuan waktu Ln.

□ **Karakter . (titik)**



Dipakai untuk memperpanjang nada. Dimana satu titik di depan suatu nada akan memperpanjang nada tersebut dengan  $\frac{1}{2}$  kali panjang nada tersebut.

Contoh :

A.. akan dimainkan selama  $9/4$

A... akan dimainkan selama  $27/8$

Dimana panjang permainan nada dapat ditentukan dengan karakter Ln.

□ **Karakter Tn**

Dipakai untuk menentukan tempo dimana n menyatakan jumlah not  $\frac{1}{4}$  dalam 1 menit. n bernilai antara 32 sampai 255. Nilai n yang standar adalah 120.

Jika panjang suatu nada tidak dapat diwakili oleh  $(A-G)n$  atau  $Ln(A-G)$ , maka alternatif lain harus dipakai, yaitu : merubah nilai n pada karakter T. Jika panjang nada akan diubah X kali, maka nilai n harus diubah dengan rumus  $n/X$ . Hasilnya dibulatkan dan ditulis di belakang karakter T.

□ **Karakter MF**

Dipakai untuk memainkan musik di latar depan. Maksudnya : nyanyian akan dilangsungkan satu per satu sampai selesai. Selama hal ini berlangsung, komputer tidak dapat melakukan proses lain.

□ **Karakter MB**

Dipakai untuk memainkan musik di latar belakang. Maksudnya : sambil dijalankan, komputer dapat melakukan proses lain.

□ **Karakter MN**

Dipakai untuk mengaktifkan kondisi baku (standar) yang berlaku untuk karakter MN, ML dan MS. Nilainya sama dengan 7.8 dalam karakter L.

□ **Karakter ML**

Dipakai untuk mengaktifkan musik legato. Setiap nada dimainkan secara penuh.

□ **Karakter MS**

Dipakai untuk mengaktifkan musik staccato. Setiap nada dimainkan dengan L sama dengan  $\frac{3}{4}$ .

□ **Karakter Xstring**

Dipakai untuk melaksanakan string yang tertulis di belakang X. Jadi karakter ini dipakai untuk meringkas penulisan program. Sekaligus menyiapkan pemakaian ulang di bagian lain dalam program tersebut.

**Program 6.39.**

```
FOR K=1 TO 2
  PLAY "CDEFGABC"
  FOR TUNGGU=1 TO 555
  NEXT TUNGGU
NEXT K
```

Output :

Akan memainkan nada dari nada do sampai do sebanyak 2 kali dengan selingan beberapa detik.

**Program 6.40.**

```
PLAY "MN T120 L4 O2"
FOR A=1 TO 4
  READ A$
  PLAY A$
NEXT A
DATA "F#8G8A.A8F#GA2."
DATA "F#G.G8AF#G2."
DATA "A>C#.C#8DDC#."
DATA "C#8<BBG.G8F#F#>D2."
```

Output :

Akan memainkan lagu daerah Jawa Tengah : SUWE ORA JAMU.