Dasar C Arduino

Pendahuluan

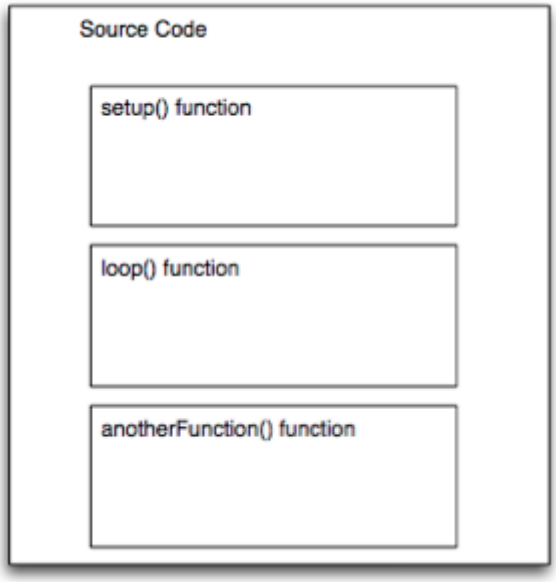
Semua bahasa perograman terdiri dari

1. ekspresi,
2. statemen,
3. blok statemen
4. blok fungsi

Ekspresi adalah kombinasi operand dan operator contoh 2+3 , X > Y dst… 2,3,X dan Y adalah operand sedangkan + dan > adalah operator .

Statemen adalah instruksi lengkap dalam bahasa C diakhiri dgn tanda **;** (titik koma) contoh : A= 2+3;

Statemen dan ekspresi C arduino indentik dengan ANSI-C , Struktur Program C Arduino minimal terdiri dari dua fungsi yaitu setup() dan loop().



Fungsi setup() dijalankan pertama kali setiap board arduino dihidupkan sedangkan fungsi loop() dijalankan terus menerus selama board arduino hidup. Pada program standar C ANSI fungsi yg pertama dijalankan adalah fungsi main() pada C arduino adalah fungsi setup().

berikut ini struktur minimal C Arduino :

```
//setup digunakan untuk inisialisasi variable, mode pin dll
void setup()
{
statement
}

void loop()
 {
statement
}
```

contoh program

```
// Pin 13  dihubungkan ke  LED pada kebanyakan  Arduino board.
// pin 13 diberi nama led:
int led = 13;

//  setup routine dijalankan sekali saat  arduino direset :
void setup() {
// inisialisasi  pin  digital  13  sbg  output.
pinMode(led, OUTPUT);
}

// loop  melakukan elsekusi berulang:
void loop() {
digitalWrite(led, HIGH); //LED on
delay(1000);
digitalWrite(led, LOW); // LED off
delay(1000);
}
```

VARIABLE DAN KONSTANTA
Variabel adalah memori penyimpanan data yang nilainya dapat diubah-ubah. variable disimpan di RAM
Konstanta adalah memori penyimpanan data yang nilainya tidak dapat diubah setelah program di compile. Konstanta disimpan di memory program.

VARIABLE SCOPE
Variable dan konstanta global dapat diakses di seluruh bagian program.
Variable dan konstanta lokal hanya dapat diakses didalam fungsi tempat dideklarasikannya.

Type Data

Arduino C mendukung sebagian besar tipe data ANSI C dengan beberapa pengecualian. Variabel adalah sebuah atau beberapa lokasi memori yang diberi nama. Ketika anda mendefinisikan variabel Anda juga harus memberitahu compiler jenis data variable tsb.
Tipe data dari variabel penting karena untuk menentukan berapa banyak byte
memori yg disediakan untuk variabel itu, dan jenis data yang dapat disimpan dalam

variabel.

Type	Size in Bits	Range
char	8	–127 to 127
unsigned char	8	0 to 255
signed char	8	–127 to 127
int	16	–32,767 to 32,767
unsigned int	16	0 to 65,535
signed int	16	–32,767 to 32,767
short int	16	–32,767 to 32,767
unsigned short int	16	0 to 65,535
signed short int	16	–32,767 to 32,767
long int	32	–2,147,483,647 to 2,147,483,647
long long int	64	$-(2^{63}-1)$ to $2^{63}-1$ (Added by C99)
signed long int	32	–2,147,483,647 to 2,147,483,647
unsigned long int	32	0 to 4,294,967,295
unsigned long long int	64	$2^{64}-1$ (Added by C99)
float	32	1E–37 to 1E+37 with six digits of precision
double	64	1E–37 to 1E+37 with ten digits of precision
long double	80	1E–37 to 1E+37 with ten digits of precision

CONTROL STATEMENT

- While Loop

while (expression) //selama expresi bernilai true laksanakan perulangan blok statemen dibawahnya.

```
{
statement1;
statement2;
…..
}
```

atau jika satement hanya 1 baris statement tdk perlu di berikurung { }

```
while(expression)
statement;
```

- Do/While Loop

do //jalankan statement dibawah (minimal 1 kali perulangan dilaksanakan) ,

```
{
statement1;
statement2;
…
}
while (expression); // jika ekspresi true , jalankan
kembali perulangan.
```

atau jika satement hanya 1 baris

do
statement;
while (expression);

- For Loop

```
for (expr1; expr2; expr3) //laksanakan perulangan sebanyak
expr2 dimulai dari expr1.
{
statement1;
statement2;
…
}
```

atau jika satement hanya 1 baris

```
for(expr1; expr2; expr3)
statement;
```

- If/Else

```
if (expression) //jika ekspresi bernilai true laksanakan blok
statemen dibawahnya
{
statement1;
statement2;
…
}
```

atau jika satement hanya 1 baris

```
if(expression)
statement;
```

- SWITCH/CASE

```
switch (expression)
{
case const1:     // jika expression = const1 eksekusi
staement2 dibawahnya
statement1;
statement2;

case const2: // jika expression = const2 eksekusi
statemenibawahnya
statement3;
statement4;
case constX: // jika espression=constX eksekusi statemen
dibawahnya
statement5;
statement6;

default:
statement7;     // jika ekspression tdk  sama dgn
const1,cons2, dan constX
statement8;
}
```

- BREAK, CONTINUE,AND GOTO

break, continue, & goto statements digunakan untuk merubah alur eksekusi untuk statement pengulangan for, while, do/while, and switch .

Break
break statement digunakan untuk keluar dari pengulangan for, while, do/while, atau switch . break keluar dari blok pengulangan dimana dia berada saja.

Continue
continue akan menyebabkan program memulai iterasi berikutnya dari statemen pengulangan while,
do/while, atau for loop. continue sama seperti break pengulangan akan berhenti pada point yg dituju (di skip), bedanya continue akan memulai loop lagi, dari atas, sedangkan break memaksa keluar dari loop / pengulangan .

```
setup()
{

}

loop()
{
int i;
for(i; i<7; i++)
{
if(i==3)
continue;
```

```
printf(%d, i )
}
}
```

OPERATOR DAN EXPRESSI

Operator Penugasan ("=")
Setelah variabel telah dideklarasikan, operasi dapat dilakukan pada variable menggunakan operator penugasan (=). Sebuah nilai yang diberikan untuk variabel dapat berupa sebuah konstanta, variabel, atau ekspresi. Sebuah ekspresi dalam bahasa C adalah kombinasi dari operan (pengidentifikasi) dan operator. operaor penugasan antara lain sebagai berikut:

```
Variable = nilai ,
contoh    nilai_max = 10;
```

Operator Arithmetic :
Perkalian *
Pembagian /
Modulo %
Penjumlahan +
Pengurangan atau Negasi –

Operator Logika
AND &&
OR ||

Operator relasional
sama dengan ==
tdk sama dengan !=
lebih kecil <
lebih kecil atau sama dengan <=
lebih besar >
lebih besar atau sama dengan >=

increment
++variable atau variable++ artinya variable=variable+1 ,

misal ++X atau X++ artinya X = X+1

perbedaaanya adalah untuk ++X , tambahkan dulu nilai X dgn 1 baru di proses baris kode tsb, sedangkan X++ , proses dulu baris code baru tambahkan nilai X dgn 1

deccrement
–variable atau variable– artinya variable=variable-1 ,

misal –Y atau Y– artinya Y = X-1

perbedaaanya adalah untuk –Y , kurangi dulu nilai Y dgn 1 baru di proses baris kode tsb, sedangkan Y– , proses dulu baris code tsb baru kurangi nilai Y dgn 1

Compound
compound digunakan hanya untuk menyingkat baris expresi saja contoh nya sbb

a += 3; artinya a = a + 3
b -= 3; artinya b = b – 3
c *= 3; artinya c = c * 3
d /= a; artinya d = d / a
a |= 3; artinya a = a OR 3
b &= 3; artinya b = b AND 3
c ^= 3; artinya c = c ^ 3

PORTD &= 3; artinya PORTD = PORTD & 3

kondisonal Expressi

```
if(expression_A)
expression_B;
else
expression_C;
```

baris kode diatas bisa digantikan oleh kondisional expressi:

```
expression_A ? expression_B : expression_C;
```

contoh : (A > 10) ? x = 5 : x =22;
kondisional expresi diatas artinya sama dengan :

```
IF(A>10)
{
x=5;
}
Else
{
x= 22
}
```

FUNGSI

yang dimaksud fungsi adalah sebuah blok statement yg melaksanakan tugas tertentu dan bisa dipakai lebih dari 1 kali di dlm program. program bahasa C terdiri dari fungsi fungsi dan sebuah program c minimal terdiri dari 1 fungsi . fungsi setup() dijalankan pertama kali saat program dijalankan. fungsi terdiri dari dua bagian utama yaitu kepala fungsi dan blok fungsi berikut ini bentuk umum sebuah fungsi adalah :

```
type name_fungsi (type paramameter)
{

statement

}
```

kepala fungsi terdiri dari type return value, nama fungsi , types dan nama parameter(jika ada) .
Statement di block fungsi menyatakan apa yg fungsi harus lakukan.
jika return value atau parameter tdk ada maka typennya ditulis void seperti dibawah ini

```
void namefungsi (void)
{

statement

}
```

Structures, Union, dan Data Storage

Arduino Library

Digital Input

void pinMode (int pin, int mode)
memerintahkan Arduino apakah pin akan diset sbg digital input atau output. Mode dapat berupa Input atau OUTPUT.

int digitalRead (int pin)
Membaca nilai dari pin digital. Nilai yg didapat dapat berupa 1 atau 0.

void digitalWrite (int pin, int value)
Menulis nilai ke pin digital. Nilai yg ditulis dapat berupa 1 atau 0.

Analog Input

int analogRead (int pin)
Membaca nilai dari pin analog. Nilai yg didapat adalah antara 0 dan 1023.

void analogWrite (int pin, int value)
Menulis nilai analog ke pin. nilai yg ditulis antara 0 dan 255.