

MODUL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN 1

Pengenalan : Variabel, Operasi, dan Mekanisme I/O



TIM ASISTEN PEMROGRAMAN
ANGKATAN 10

ILMU KOMPUTER FPMIPA UPI

9/9/2019

PENDAHULUAN

Memulai perjalanan sebagai *Computer Scientist* tentunya harus diawali dengan dasar – dasar pengetahuan tentang komputer yaitu *Program*. Dimulai dari bagaimana sebuah Algoritma dan Pemrograman.

Penulis mengucapkan terima kasih atas bantuan dari berbagai pihak baik secara langsung maupun tidak langsung dalam proses pembuatan modul ini.

PENDAHULUAN

Apa itu Algoritma ? Algoritma adalah cara atau sebuah jalan yang tersusun untuk menyelesaikan sebuah masalah. Dalam pemrograman, Algoritma digunakan untuk membuat program yang akan menyelesaikan masalah pada komputer.

Apa yang akan dipelajari ?

- Sintaksis (*Syntax*) dalam Bahasa C
- Menulis program dalam Bahasa C
- Kompilasi
- Variabel dan tipe data
- Operator dan Operasi aritmatika dasar
- Mekanisme Input dan Output (I/O)

ISI

Pada bagian ini akan dijelaskan apa yang akan dipelajari dari modul ini.

Compiler

Compiler adalah sebuah program yang akan mengubah sebuah instruksi ke dalam Bahasa mesin. (*Cambridge University Press, 2013*).

Salah satu compiler yang digunakan untuk mengkompilasi kode program berbahasa C adalah GNU *Compiler Collective (GCC)*.

Windows

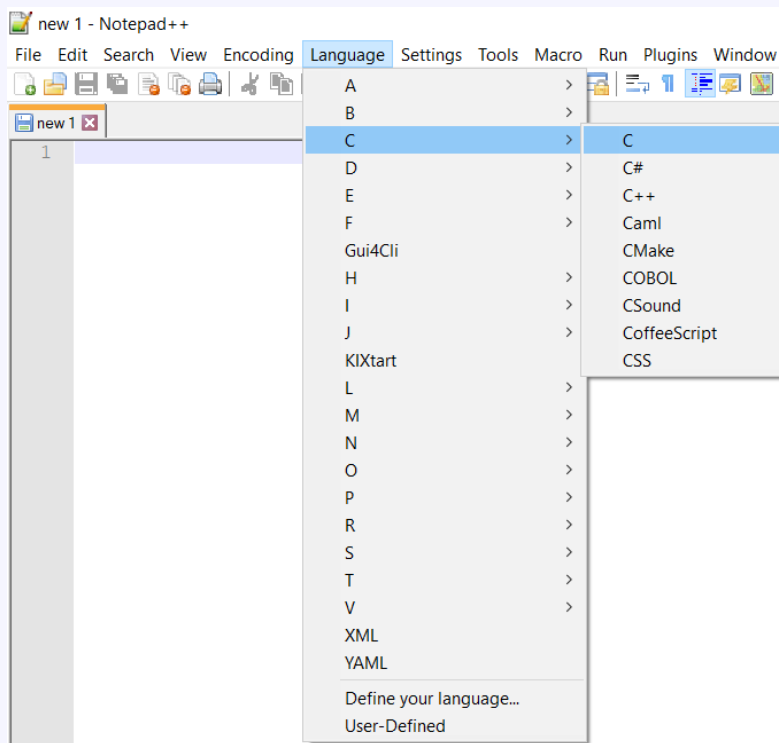
Di modul ini kita akan membahas cara menginstall salah satu compiler yaitu MinGW.

- Instalasi MinGW
Instalasi disarankan menggunakan installer yang akan mendownload kembali paket MinGW yang diperlukan dan otomatis mengatur sistem agar bisa langsung digunakan.
- Manual Instalasi MinGW
 1. Simpan folder MinGW yang sudah dibagikan di C drive,
 2. Buka Control Panel, lalu buka bagian System,
 3. Pilih menu "Advancement system settings"
 4. Pilih menu "Environment Variables"
 5. Pilih "Path" di "System Variables"
 6. Pilih "Edit" ...
 7. Pilih "New", lalu masukan alamat "C:\MinGW\bin".
 8. Lalu pilih "Ok".

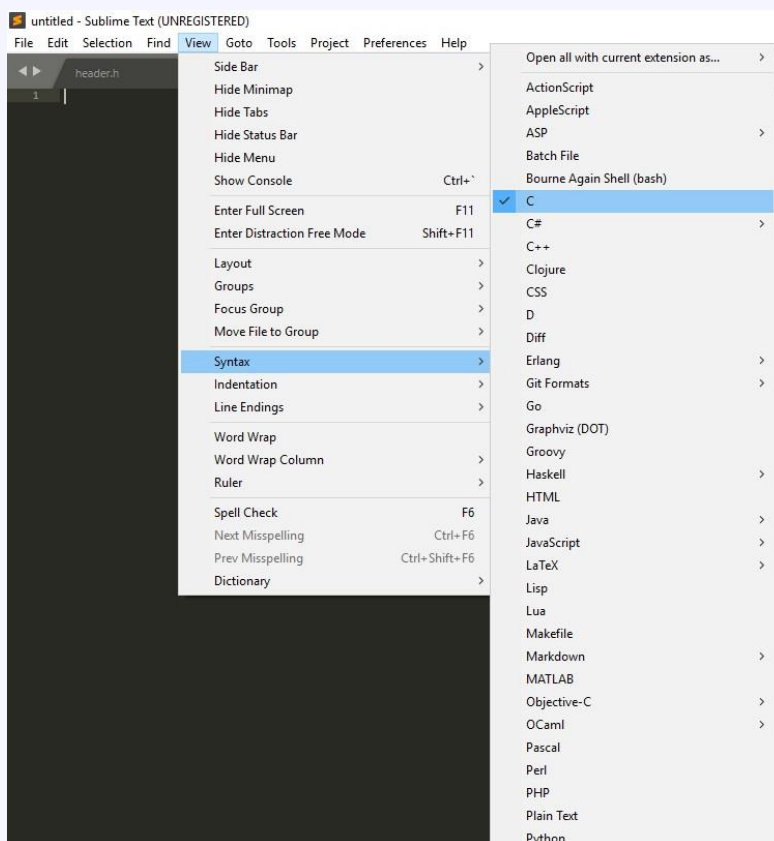
Text Editor

Disarankan menggunakan text editor yang mendukung Bahasa C seperti Notepad++ atau Sublime Text. Atur text untuk meng-highlight sintaksis bahasa C.

Disarankan jangan memakai Integrated Development Environment (IDE) ketika mempelajari sebuah Bahasa.



Setting di notepad++



Setting di Sublime Text

Mulai Ngoding Kuy

Untuk menulis program C dapat diawali dengan template berikut.

Template Program Bahasa C

Berikut merupakan template untuk program Bahasa C.

```
C hello.c  ×
C hello.c > ...
1  #include <stdio.h> // Ini adalah header
2
3  int main(){
4
5      /* Isi kodemu disini */
6
7  return 0;
8  }
```

Hello World !

Selanjutnya kita kembangkan kodingan diatas menjadi program yang akan menampilkan kata "Hello World !".

Caranya dengan menambahkan kode berikut pada baris ke 4 yaitu dibawah sintaksis " int main(){ " seperti gambar berikut.

```
C hello.c  ×
g: > Aspro > C hello.c > ...
1  #include <stdio.h> // Ini adalah header
2
3  int main(){
4
5      /* Isi kodemu disini */
6
7      // Menampilkan " Hello World ! " ke layar
8      printf("Hello World !\n");
9
10 return 0;
11 }
```

Penjelasan

Masih bingung dengan maksud kode diatas ? berikut adalah gambaran singkat tentang penjelasan kodingan singkat tersebut.

Hello.c:1 `#include`, cara untuk mengimpor konten dari sebuah file ke source code. Pada Bahasa C ekstensi `.h` digunakan untuk file header yang berisi fungsi untuk digunakan pada program anda.

Hello.c:3 Awal dari fungsi utama (`Main`). Untuk memulai bagian isi dari fungsi, digunakan karakter Opening Brace "`{`" yang mengindikasikan awal dari sebuah blok. Kode program yang akan dijalankan harus ditulis setelah karakter tersebut.

Hello.c:5 Ini merupakan komentar. Komentar biasanya digunakan untuk menjelaskan source code. Komentar tidak akan mengganggu jalan kerjanya sebuah program.

Hello.c:8 Memanggil fungsi `printf` untuk menampilkan tulisan ke layar, dengan isi teks "`Hello World !\n`" dimana kata yang akan ditampilkan berada di antara tanda dua kutip. Sedangkan `\n` berfungsi untuk memberi baris baru (newline/enter). Jangan lupa karakter semicolon atau titik koma untuk mengakhiri pemanggilan fungsi.

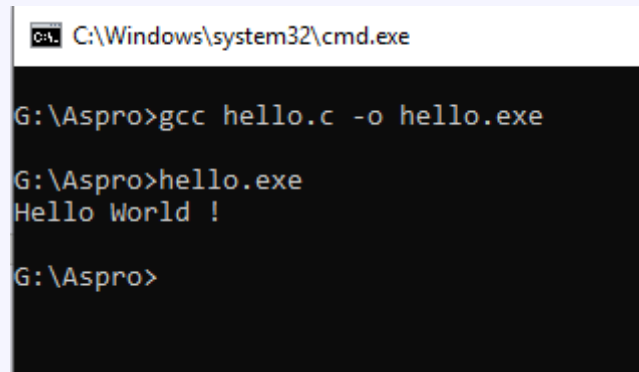
Hello.c:10 Return dari fungsi main yang memberikan sistem operasi tanda untuk keluar dari program setelah kode dijalankan.

Hello.c:11 Pada akhir kode, Main ditutup dengan karakter closing brace "`}`" dan menjadi akhir dari program.

Kompilasi

Berikut adalah cara mengompilasi kode yang telah dibuat agar menjadi suatu program.

Dengan catatan kode di bawah disimpan pada Drive G pada folder Aspro.



```
C:\Windows\system32\cmd.exe

G:\Aspro>gcc hello.c -o hello.exe

G:\Aspro>hello.exe
Hello World !

G:\Aspro>
```

Untuk mengubah direktori/folder pada cmd, gunakan perintah `cd`. Nama folder harus sama persis.

Misal :

```
G:\>cd Aspro
```

```
G:\Aspro>_
```

Untuk mengompilasi gunakan sintaksis berikut :

```
gcc [nama_source_kode] -o [nama_program_yang_dihasilkan]
```

```
gcc hello.c -o hello.exe
```

Jalankan program dengan mengetik nama programnya.

```
hello.exe
```

Komentar

Komentar dalam source code digunakan untuk menjelaskan kode untuk mengingat atau memahami dari fungsi kode yang telah dibuat. Oleh kompiler, komentar akan diabaikan dengan kata lain komentar tidak akan mempengaruhi jalannya sebuah program.

Penulisan komentar untuk satu baris menggunakan double forward slash (`//`). Untuk penulisan komentar dengan lebih dari satu baris menggunakan format berikut (`/* ... */`).

Programmer yang baik selalu memberi komentar pada programnya.

```
1 // ini cara menulis komentar satu baris
2
3 /*
4 ini
5 cara
6 menulis
7 banyak
8 baris
9 */
```


Variabel dan Tipe Data

Pada bagian ini kita akan membahas Variabel dan Tipe Data.

Variabel

Variabel merupakan tempat untuk menyimpan data dengan tipe tertentu yang isinya bias diubah – ubah sesuai dengan tipenya.

Aturan penamaan variabel :

- Case-sensitive
- Nama variabel tidak boleh berupa angka ataupun diawali dengan angka
- Nama variabel tidak boleh mengandung spasi. Gantinya dapat dengan menggunakan karakter underscore _ atau dengan menggabungkan kata ke bentuk **camelCase**.
- Nama variabel tidak boleh mengandung karakter-karakter simbol (@, #, \$, %, dll.) walaupun terletak di tengah ataupun akhir nama variabel.
- Nama variabel tidak boleh menggunakan kata kunci maupun makro yang telah didefinisikan dalam Bahasa C, contoh: **void**, **printf**, **scanf**, **return**, dll.
- Penamaan variabel diharapkan merupakan deskripsi dari penggunaannya. Sebagai contoh, variabel menampung nilai luas diberi nama luas.

Tipe Data (Data Type)

Tipe data adalah jenis data berdasarkan isi dan sifatnya. Dalam logika kita tipe data adalah jenis dari sesuatu yang dapat dimasukkan ke dalam kotak kosong yang hanya khusus dibuat untuk jenis benda dengan jenis tertentu.

a. Bilangan Bulat (*Integer*)

Tipe Data	Jangkauan	Format sintaksis
int	-32768, ... , +32768	%d
long int	-2147483648, ... , +2147483647	%l
short int	-128, ... , +127	%d

b. Bilangan riil (*Floating-Point*)

Tipe Data	Jangkauan	Format sintaksis
float	3.4e-38 sampai 3.4e+38	%f
double	1.7e-308 sampai 1.7e+308	%f
long double	1.7e-308 sampai 1.7e+308	%lf

c. Karakter (*Character*)

Tipe Data	Jangkauan	Format sintaksis
char	-128 sampai +127	%c
signed char	-128 sampai +127	%c
long double	0 sampai 1.7e+308	%c

Deklarasi Variabel

Untuk mendeklarasikan variabel dalam bahasa C, digunakan sintaksis berikut :

[tipe_data][nama_variable];

Contoh deklarasi dengan beragam bentuk penamaan variabel (underscore, dash, camelCase) :

int angka;

float angka_berkoma;

char variabelKarakter;

Contoh deklarasi banyak variabel dalam satu baris :

int nomor1, nomor2, nomor3;

```
C variabel.c
g: > Aspro > C variabel.c
1  #include <stdio.h> // Ini adalah header
2
3  int main(){
4
5      // Contoh deklarasi variabel
6      int angka;
7      float bilangan_koma;
8      char karakter;
9
10     // Contoh deklarasi pada satu baris
11     char a, b, c;
12     /* Hasilnya akan sama dengan
13         char a;
14         char b;
15         char c;
16     */
17
18     return 0;
19 }
```

Operator

Operator adalah simbol atau tanda yang diletakkan pada dua buah operan dapat menghasilkan sebuah hasil. Operator dibagi menjadi berbagai jenis, diantaranya :

Operator Assignment

Berikut merupakan template yang dapat digunakan untuk membuat program dengan Bahasa C. Operator Penugasan atau Operator Assignment (=) yang digunakan untuk memasukkan nilai ke dalam suatu variabel.

Contoh untuk menugaskan (assign) n agar memegang nilai 5 :

```

C assignment.c ×
Aspro > C assignment.c > ...
1  #include <stdio.h>
2
3  int main(){
4
5      // Deklarasi variabel
6      int angka;
7      // Pemasukan nilai ke variabel "angka"
8      angka = 5;
9
10     // Deklarasi dan pemasukan nilai dalam satu baris
11     char huruf = 'a'; // Assignment tipe data char menggunakan tanda petik ( ' ' )
12
13     // Deklarasi beberapa variabel dan pemasukan nilai sekaligus
14     int angka1, angka2, angka3;
15     angka1 = angka2 = angka3 = 23; // ketiga variabel akan berisi nilai 23
16
17     return 0;
18
19 }

```

Operator Unary

Operator yang digunakan untuk melakukan operasi matematik yang hanya menggunakan satu operand.

Operator	Jenis Operasi	Contoh penggunaan
+	Membuat nilai positif	+10
-	Membuat nilai negatif	-10
++	Increment (menambahkan nilai 1)	a++
--	Decrement (mengurangi nilai 1)	a--

Operator Aritmatika

Operator yang digunakan untuk melakukan operasi matematik dengan dua operand.

Operator	Jenis Operasi	Contoh penggunaan
+	Penjumlahan	1 + 1
-	Pengurangan	3 - 1
*	Perkalian	5 * 5
/	Pembagian	20 / 4
%	Modulus (sisa bagi)	20 % 4

Mekanisme Input dan Output

Salah satu cara untuk melakukan input dan output dalam Bahasa pemrograman C ialah dengan menggunakan **printf** dan **scanf**. Masih banyak cara lainnya tidak dibahas di modul ini.

Printf dan scanf merupakan fungsi dari library `stdio.h`. Pastikan anda melakukan include di awal source code.

Formatted Print

Formatted print atau pencetakan yang diformat, merupakan fungsi untuk menampilkan data ke output standar8 (layar). Untuk informasi lebih lanjut, silahkan baca manual untuk printf.

Penulisannya ialah seperti berikut :

```
printf("konten yang akan di print");
```

Untuk menampilkan data dari variabel direpresentasikan dengan simbol formatnya kemudian nama variabel disebutkan setelahnya. Sehingga ditulis seperti berikut:

```
float angka = 2.3;
```

```
printf("%f", angka);
```

```
//      ^      ^      menyebutkan nama variabel yang akan diprint
```

```
//      ^ perwakilan variabel dengan simbol format
```

Menampilkan teks dan variabel sekaligus :

```
int nomor = 5;
```

```
char huruf = 'A';
```

```
printf("Nomor antrian: %c%d.", huruf, nomor); // Nomor antrian: A5
```

```
C output.c x
Aspro > C output.c > ...
1  #include <stdio.h>
2
3  int main(){
4
5      // Menampilkan teks
6      printf("konten yang akan di print\n");
7
8      float angka = 2.3;
9      printf("%f\n", angka);
10     //      ^      ^ menyebutkan nama variabel yang akan diprint
11     //      ^ perwakilan variabel dengan simbol format.
12
13     // Menampilkan teks dan variabel sekaligus
14     int nomor = 5;
15     char huruf = 'A';
16     printf("Nomor antrian: %c %d.", huruf, nomor); // Nomor antrian: A5.
17
18     return 0;
19 }
```

Formatted Scan

Formatted scan atau pemindaian terformat, merupakan fungsi untuk meminta masukan (input) dari user melalui input standar (console).

Untuk informasi lebih lanjut, silahkan baca manual untuk scanf. Untuk meminta masukan dari user, variabel diwakilkan dengan simbol formatnya kemudian menyebutkan nama variabelnya.

Sebelum nama variabel harus menggunakan simbol "&". Sehingga penulisannya seperti berikut :

```
char huruf;
```

```
scanf(" %c", &huruf);
```

```
//      ^      ^ nama variabel yang akan diberi nilai dari masukan
```

```
//      ^ perwakilan variabel ditulis dengan simbol format
```

```
int angka;
```

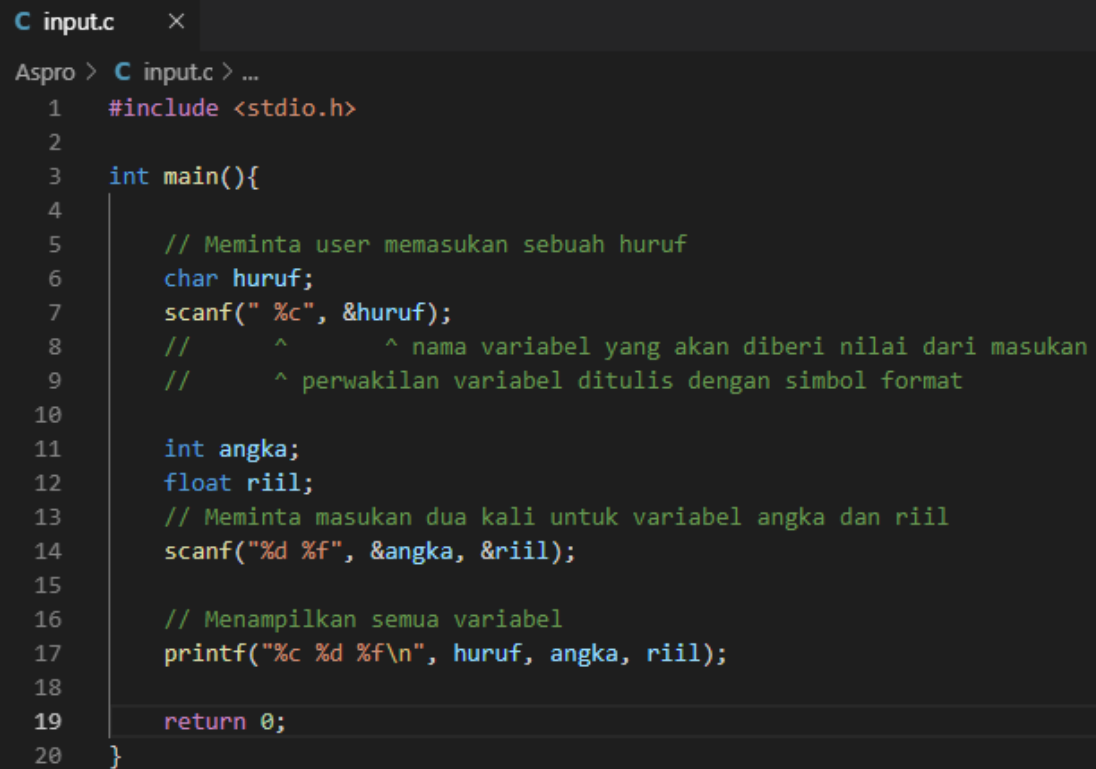
```
float riil;
```

```
// Meminta masukan dua kali untuk variabel angka dan riil
```

```
scanf("%d %f", &angka, &riil);
```

```
// Menampilkan semua variabel
```

```
printf("%c %d %f\n", huruf, angka, riil);
```



```
C input.c x
Aspro > C input.c > ...
1  #include <stdio.h>
2
3  int main(){
4
5      // Meminta user memasukan sebuah huruf
6      char huruf;
7      scanf(" %c", &huruf);
8      //      ^      ^ nama variabel yang akan diberi nilai dari masukan
9      //      ^ perwakilan variabel ditulis dengan simbol format
10
11     int angka;
12     float riil;
13     // Meminta masukan dua kali untuk variabel angka dan riil
14     scanf("%d %f", &angka, &riil);
15
16     // Menampilkan semua variabel
17     printf("%c %d %f\n", huruf, angka, riil);
18
19     return 0;
20 }
```

Menulis Program Kalkulator Sederhana

C kalkulator.c ×

Aspro > C kalkulator.c > ...

```
1  #include <stdio.h>
2
3  int main(){
4
5      /* Membuat variabel penampung bilangan pertama */
6      int bilangan_1;
7
8      /* Membuat variabel penampung bilangan kedua */
9      int bilangan_2;
10
11     // Membuat variabel penampung hasil operasi
12     int hasilTambah;
13     int hasilKurang;
14     int hasilKali;
15     int hasilBagi;
16     int hasilModulus;
17
18     /* Meminta pengguna untuk menginputkan (memasukan) bilangan pertama */
19     printf("Masukkan bilangan pertama : ");
20     scanf("%d", &bilangan_1);
21
22     /* Meminta pengguna untuk menginputkan (memasukan) bilangan kedua */
23     printf("Masukkan bilangan kedua : ");
24     scanf("%d", &bilangan_2);
25
26     // Proses operasi
27     hasilTambah = bilangan_1 + bilangan_2;
28     hasilKurang = bilangan_1 - bilangan_2;
29     hasilKali = bilangan_1 * bilangan_2;
30     hasilBagi = bilangan_1 / bilangan_2;
31     hasilModulus = bilangan_1 % bilangan_2;
32
33     // Menampilkan hasil operasi
34     printf("Pertambahan : %d\n", hasilTambah);
35     printf("Pengurangan : %d\n", hasilKurang);
36     printf("Perkalian : %d\n", hasilKali);
37     printf("Pembagian : %d\n", hasilBagi);
38     printf("Modulus : %d\n", hasilModulus);
39
40     return 0;
41
42 }
```


DAFTAR PUSTAKA

- Cambridge University Press. (2013). Definition of compiler in English. Diambil kembali dari Cambridge Dictionaries:
<https://dictionary.cambridge.org/dictionary/english/compiler>
- Tim Asisten Pemrograman Angkatan 9. (2018). Pengenalan: Peubah, Operasi, dan I/O. Modul Praktikum Algoritma dan Pemrograman 1. Bandung, Jawa Barat, Indonesia
- Sukamto, R. A. (2018). Logika Algoritma dan Pemrograman Dasar. Bandung: Modula