

## Modul PDP-01

### **Petunjuk :**

1. Tulis semua program di bawah ini satu per satu
2. Lakukan kompilasi di bawah program satu per satu
3. Berapa Kali anda melakukan kompilasi
4. Tulis Kesalahan yang di temukan saat kompilasi
5. Format laporan adalah sebagai berikut :

No	Jumlah kompilasi	Kesalahan	Output
1	2	1. Salah sintax 2. Kurang titik koma	Hello World
2	1	1. Kurang tanda “,”	23
3	1	-	12 21
.			
.			
n			

### **No Program**

- 1
  - a. Utiliti Membuat File tipe data Boolean, dengan nama **boolean.h** dengan sembarang text editor, lalu tulis kode di bawah ini dan simpan dengan format file All File

```
/*File boolean.h*/
#ifndef BOOLEAN_H
#define BOOLEAN_H

#define true 1
#define false 0
#define boolean unsigned char

#endif
```
  - b. Untuk memakai file ini, maka simpan file ini dalam direktori yang sama dengan file source code anda, lalu sertakan `#include "boolean.h"` dalam hash include.
  - c. Setelah itu anda dapat mendeklarasikan tipe data Boolean seperti contoh di bawah ini, `boolean benar; boolean state;`
  - d. Uji dengan file driver, `driver.c`

```
#include "boolean.h"
int main () {
    printf("Men-Test File boolean.h \n");
    boolean betul=true;
    printf("variabel betul desimal: %d\n", betul);
    printf("variabel betul:%s\n",betul?"true":"false");
}
```
- 2

```
/* File hello.c */
void
main()
```

```

    {
        printf("hello\n ");
    }
3  /* File : hello1.c */
   /* menuliskan hello ke layar */
   /* pola ini merupakan standard yang dipakai di kelas */
   int
   main ()
   {
       /* KAMUS */
       /* ALGORITMA */
       printf ("hello\n");
       return 0;
   }
   TIPE DATA SEDERHANA : int, float, char dan string
3a /* contoh membaca integer dan float*/
   /* kemudian menuliskan nilai yang dibaca */
   int main () {
       /* Kamus */
       int a; float b;
       /* Program */
       printf ("Membaca dan menulis, ketik nilai integer: ");
       scanf ("%d", &a);
       printf ("Membaca dan menulis, ketik nilai float: ");
       scanf ("%d", &b);
       printf ("Nilai yang dibaca : %f \n", b);
       return 0;
   }
3b int main() {
   /* Kamus */
   char cc;
   /* Algoritma */
   printf ("hello\n");
   printf("baca 1 kar : ");scanf ("%c ", cc);
   printf ("%c", cc);
   printf ("bye \n");
   return 0;
}
3c /* deklarasi dan alokasi string, kemudian mengisinya
    dengan membaca
    */
    #include<string.h>
    int
    main () {
        /* Kamus */
        char *str;
        char *str1;

```

```

/* Program */
printf ("\nBaca string, maks 20 karakter: ");
/*Alokasi string sebesar 20 karakter*/
str = (char *) malloc (20 * sizeof (char));
printf("masukkan sebuah string, max 20 kar: ");
scanf("%s",str);
printf ("String yang dibaca : %s\n", str);
/*Alokasi string sebesar 20 karakter*/
str1 = (char *) malloc (20 * sizeof (char));
strcpy (str1, str);
printf ("String yang disalin : %s\n", str1);
return 0;
}
3d /* File : enum.c */
/* Deklarasi dan pemakaian type enumerasi */
int
main () {
/* KAMUS */
/* "type" */
enum hari { senin, Selasa, Rabu, Kamis, Jumat, Sabtu }
hariku; /* hariku : variabel */
enum { satu, dua, tiga } angka; /* variabel */
enum { KEYWORD = 01, EXTERNAL = 03, STATIC = 04 };
/* sekedar enumerasi "konstanta" bernama ",
mengelompokkan
*/
typedef enum { merah,putih,kuning } warna; /* nama type
*/
unsigned int flags;
warna w = kuning;
/* ALGORITMA */
angka = tiga;
printf ("Angka %d \n ", angka);
hariku = 0;
printf ("Hari %d \n ", hariku);
printf ("Masukkan sebuah angka [0..2] ");
scanf ("%d", &angka);
printf ("Angka %d \n ", angka);
flags = EXTERNAL;
printf ("flags %d \n ", flags);
printf ("Warna = %d\n", w);
return 0;
}
3e /* latihan array statis : mengisi dg assignment, menulis
*/
int main () {
/* Kamus */

```

```

int i;
int tab[10]; /* Cara mengacu elemen ke-i: tab[i] */
int N;
/* Program */
N = 5;
printf ("Isi dan print tabel untuk indeks 1..5 \n");
/* isi dengan assignment */
for (i = 1; i <= N; i++) {
    tab[i] = i;
};
/* traversal: print */
for (i = 1; i <= N; i++) {
    printf ("i=%d tab[i]=%d \n", i, tab[i]);
};
return 0;
}
3f /* contoh pendefinisian dan pengisian struktur: Titik,
mhs, meter */
int main () {
/* Kamus */
/* cara 1 */
struct {
    char nama[20];
    int nim;
    int nilai;
} Mhs; /* nama variabel berupa struct */
/* cara 2 */
/* tag, utk menyebut struct ... */
struct meter {
    int m;
    int cm;
};
struct meter M1; /* M1 adalah variabel */
/* cara 3 : INI yang membuat TYPE BARU */
typedef struct {
    float x;
    float y;
} Point; /* nama type */
Point P1;
Point P3;
/* latihan baca */
/* Algoritma */
printf ("Contoh mengisi struktur dg assignment : \n");
printf ("Titik P1, dengan P1.x dan P1.y:\n");
P1.x = 1.1;
P1.y = 2.5;
printf ("P1.x = %4.1f\nP1.y = %4.1f\n", P1.x, P1.y);

```

```

printf ("Baca Titik P3\n");
scanf ("%f %f", &P3.x, &P3.y);
printf ("P3.x = %f \nP3.y = %f \n", P3.x, P3.y);
strcpy (Mhs.nama, "Juliette");
Mhs.nim = 7473;
Mhs.nilai = 80;
printf ("Hasil assignment thd Mhs \n");
printf ("Nama = %s\nNim = %d\nNilai = %d\n", Mhs.nama,
Mhs.nim, Mhs.nilai);
return 0;
}

```

#### **ASSIGNMENT, INPUT & OUTPUT**

- 4    

```

/* File : ASIGN.C */
/* Assignment nilai integer dan print */
int
main ()
{ /* Kamus */
  int i;
  /* Program */
  printf ("hello\n");
  i = 5;
  printf ("Ini nilai i : %d \n", i);
  return 0;
}

```
  
- 5    

```

/* File asgddl.c */
int main()
{ /* Kamus */
  float f;
  long double fl1;
  /* Algoritma */
  f= 20.0f;
  fl1=10.0L;
  return 0;
}

```
  
- 6    

```

/* File : ASIGNi.C */
/* Assignement dan print */
#include <limits.h>
int
main ()
{
  /* Kamus */
  int i;

```

```

    long int ii;
/* Program */
printf ("hello\n");
i = 1234;
ii = 123456;
printf ("Ini nilai i=1234 = : %d \n", i);
printf ("Ini nilai ii=123456 : : %10d \n", ii);
/* print nilai batas integer */
printf ("Min & Max integer: %d,%d \n", INT_MIN,INT_MAX);
printf ("Max long integer : %ld, %ld \n", LONG_MAX);
return 0;
}
7 /* File : asign2.c */
/* Deskripsi : */
/* Program ini berisi contoh sederhana untuk */
/* mendefinisikan */
/* variabel-variabel bilangan bulat */
/*(short int, int, long int),*/
/* karakter, bilangan riil, */
/*-----*/
int
main ()
{
/* KAMUS */
short ks = 1;
int ki = 1;
long kl = 10000;
char c = 65; /* inisialisasi karakter dg integer*/
char cl = 'Z'; /* inisialisasi karakter dg karakter */
float x = 1.55;
/* Algoritma */
/* penulisan karakter sebagai karakter */
printf ("Karakter = %c\n", c);
printf ("Karakter = %c\n", cl);
/* penulisan karakter sebagai integer */
printf ("Karakter = %d\n", c);
printf ("Karakter = %d\n", cl);
printf ("Bilangan integer (short) = %d\n", ks);
printf ("\t\t(int) = %d\n", ki);
printf ("\t\t(long) = %ld\n", kl); /* perhatikan format
%ld */
printf ("Bilangan Real = %f8.3\n", x);
return 0;
}

```

## OPERATOR & EKSPRESI

```

8 /* File : incr.c */

```

```

/* Efek dari operator ++ */
int
main ()
{
/* Kamus */
int i, j;
/* Program */
i = 3;
j = i++;
printf ("Nilai i : %d\nNilai j : %d\n", ++i, j);
return 0;
}
9 /* File : oper1.c */
/* pemakaian beberapa operator terhadap bit */
int
main ()
{
/* KAMUS */
int n = 10; /* 1010 */
int x = 1; /* 1 */
int y = 2; /* 10 */
/* ALGORITMA */
printf ("n = %d \n", n);
printf ("x = %d \n", x);
printf ("y = %d \n", y);
printf ("n & 8 = %d \n", n & 8); /* 1010 AND 1000 */
printf ("x & ~ 8 = %d \n", x & ~8); /* 1 AND 0111 */
printf ("y << 2 = %d \n", y << 2); /* 10 ==> 1000 = 8 */
printf ("y >> 3 = %d \n", y >>3); /* 10 ==> 0000 = 0 */
return 0;
}
10 /* File : oprator.c */
/* Contoh pengoperasian variabel bertipe dasar */
int main ()
#include "boolean.h"
{ /* Kamus */
boolean Bool1, Bool2, TF ;
int i, j, hsl ;
float x, y, res;
/* algoritma */
printf ("Utk program ini, baca teksnya dan tambahkan
output");
Bool1 = true; Bool2 = false;
TF = Bool1 && Bool2 ; /* Boolean AND */
TF = Bool1 || Bool2 ; /* Boolean OR */
TF = ! Bool1 ; /* NOT */
TF = Bool1 ^Bool2; /* XOR */

```

```

/* operasi numerik */
i = 5; j = 2 ;
hsl = i+j; hsl = i - j; hsl = i / j; hsl = i * j;
hsl = i /j ; /* pembagian bulat */
hsl = i%j ; /* sisa. modulo */
/* operasi numerik */
x = 5.0 ; y = 2.0 ;
res = x + y; res = x - y; res = x / y; res = x * y;
/* operasi relasional numerik */
TF = (i==j); TF = (i!=j);
TF = (i < j); TF = (i > j); TF = (i <= j); TF = (i >= j);
/* operasi relasional numerik */
TF = (x != y);
TF = (x < y); TF = (x > y); TF = (x <= y); TF = (x >= y);
return 0;
}
11 /* File : oper2.c */
/* pemakaian beberapa operator terhadap RELATIONAL DAN
bit */
int main ()
{ /* KAMUS */
char i, j;
/* ALGORITMA */
i = 3; /* 00000011 dalam biner */
j = 4; /* 00000100 dalam biner */
printf ("i = %d \n", i);
printf ("j = %d \n", j);
printf (" i && j = %d \n", i && j); /* 1:op logik : true
and
true=> true */
printf (" i & j = %d \n", i & j); /* 0: 00000000 dalam
biner */
printf (" i|| j = %d \n", i || j); /* 1 : OR LOJIK, True
or true
=> true */
printf (" i| j = %d \n", i | j); /* 7: 00000111 biner */
printf (" i^j = %d \n", i ^ j); /* 7: 00000111 biner */
printf (" ~i = %d \n", ~i); /* -4: 11111100 biner */
return 0;
}
12 /* File : exp.c */
/* pemakaian operator kondisional */
int
main ()
{
/* KAMUS */
int x = 1;

```



```

int y = 2;
/* ALGORITMA */
printf ("x = %d \n", x);
printf ("y = %d \n", y);
printf ("hasil ekspresi = (x<y)?x:y = %d \n", (x < y) ? x
: y);
return 0;
}

```

```

13  /* File : expl.c */
    /* pembagian integer, casting */
    int main ()
    { /* KAMUS */
      int x = 1;
      int y = 2;
      float fx;
      float fy;
      /* ALGORITMA */
      printf ("x/y (format integer) = %d \n", x/y);
      printf ("x/y (format float) = %f \n", x/y);
      /* supaya hasilnya tidak nol */
      fx=x;
      fy=y;
      printf ("x/y (format integer) = %d \n", fx/fy);
      printf ("x/y (format float) = %f \n", fx/fy);
      /* casting */
      printf ("float(x)/float(y) (format integer) = %d \n",
      (float)x/(float)y);
      printf ("float(x)/float(y) (format float) = %f \n",
      (float)x/(float)y);
      x = 10;
      y = 3;
      printf ("x/y (format integer) = %d \n", x/y);
      printf ("x/y (format float) = %f \n", x/y);
      return 0;
    }

```

```

14  /* File : oper3.c */
    /* Operator terner */
    /* Ekspresi ditulis sebagai makro */
    #define max(a,b) ((a>b) ? a: b)
    int
    main ()
    {
      /* KAMUS */

```

```
int i = 0; /* perhatikan int i,j=0 bukan seperti ini */
int j = 0;
char c = 8;
char d = 10;
char e = max (c, d);
int k = max (i, j);
/* ALGORITMA */
printf ("Nilai e = %d \n", e);
printf ("Nilai k = %d \n", k);
i = 2;
j = 3;
k = max (i++, j++);
printf ("Nilai k = %d \n", k);
return 0;
}
```