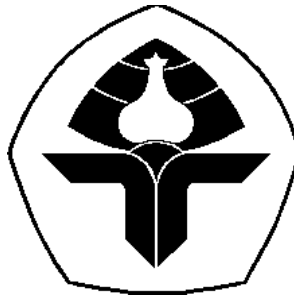


JOB SHEET

PEMROGRAMAN DASAR

MKK-13206



Disusun Oleh
Putu Indah Ciptayani, S.Kom. M.Cs

**PROGRAM STUDI MANAJEMEN INFORMATIKA
JURUSAN TEKNIK ELEKTRO
POLITEKNIK NEGERI BALI**

2018

| | | |
|--|--|--|
| POLITEKNIK NEGERI BALI | Job Sheet 1 Pengenalan Bahasa C | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 3 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 1 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami lingkungan kerja dan menggunakan struktur dasar Bahasa C.

C. Indikator

1. Mahasiswa mampu menggunakan IDE Bahasa C
2. Mahasiswa mampu menggunakan struktur dasar Bahasa C

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

1. Tahapan Penulisan Program

Pemrosesan program C umumnya melalui lima tahapan, yaitu :

- a) Tahap menulis kode sumber adalah proses penulisan program dengan menggunakan editor C dan kemudian disimpan ke dalam penyimpanan sekunder. File program C diberi nama dengan menambahkan ekstensi **.c**. Adapun hal yang perlu diperhatikan dalam penulisan bahasa C adalah sifat *case sensitive*, dimana huruf besar dan kecil dibedakan. Setiap akhir dari sebuah instruksi harus diakhiri tanda *semicolon* (;).
- b) Tahap berikutnya adalah melakukan kompilasi, dimana kompilator akan menerjemahkan program ke dalam kode bahasa mesin (*object code* = **.obj**). Di dalam sistem C, bagian program yang disebut *preprocessor* (dipanggil melalui *preprocessor directives*) akan diproses di awal sebelum tahap penerjemahan program dimulai.
- c) Proses *link* dilakukan untuk mengkombinasikan kode obyek (**obj**) dengan pustaka C yang tersedia agar menghasilkan file program

lengkap yang dapat dieksekusi oleh komputer, file yang dihasilkan mempunyai ekstensi **.exe**.

- d) Tahap selanjutnya adalah memanggil file program hasil langkah tiga ke dalam memori untuk dapat dijalankan oleh komputer.
- e) Tahap terakhir adalah menjalankan atau mengeksekusi program.

2. Fungsi

Program C pada hakekatnya tersusun atas sejumlah blok fungsi. Sebuah program minimal mengandung sebuah fungsi. Setiap fungsi terdiri dari satu atau beberapa pernyataan yang secara keseluruhan dimaksudkan untuk melaksanakan tugas khusus. Secara umum, fungsi mempunyai bentuk sebagai berikut :

```
nama-fungsi (daftar parameter)
deklarasi parameter;
{
    bagian isi fungsi
}
```

3. Fungsi main()

Fungsi **main()** harus ada pada program, sebab fungsi inilah yang menjadi titik awal dan akhir eksekusi program. Tanda { diawal fungsi menyatakan awal tubuh fungsi dan juga sebagai awal program dijalankan, sedangkan tanda } di akhir fungsi merupakan akhir bagian isi fungsi dan sekaligus adalah akhir dari eksekusi program.

```
int main()
{
    Isi fungsi main
}
```

4. Fungsi printf()

Fungsi **printf()** merupakan fungsi yang umum dipakai untuk menampilkan suatu keluaran pada layar peraga. Dalam bentuk umum, format **printf()** adalah sebagai berikut :

`printf(kontrol string, daftar pernyataan);`

String kontrol dapat berupa satu atau sejumlah karakter yang akan ditampilkan ataupun berupa penentu format yang akan mengatur penampilan dari argumen yang terletak pada **daftar argumen**. Mengenai penentu format diantaranya berupa :

%d Untuk menampilkan bilangan bulat (integer)
%f Untuk menampilkan bilangan pecahan
%c Untuk menampilkan sebuah karakter
%s Untuk menampilkan sebuah string

5. Preprosesor #include

#include merupakan salah satu jenis pengarah preprosesor yang dipakai untuk membaca file yang dinamakan file-judul (**header file**). File-judul yaitu file yang diantaranya berisi deklarasi fungsi dan definisi konstanta. File ini mempunyai ciri yaitu nama file-nya diakhiri dengan ekstensi .h , misalnya = stdio.h.

6. Komentar

Komentar atau comment adalah naskah program yang tidak akan diproses oleh *compiler*. Pada saat proses kompilasi berlangsung, teks program yang termasuk ke dalam komentar akan diabaikan oleh compiler. Kehadiran komentar di dalam program sangat dibutuhkan terutama jika program yang dibuat sudah masuk ke skala besar dan kompleks. Setidaknya ada 3 alasan mengapa komentar perlu ditulis :

1. Dokumentasi
2. Debugging
3. Maintenance

Bahasa C menyediakan dua cara menulis komentar :

1. Karakter “//” digunakan untuk mengawali penulisan komentar dalam satu baris. Karakter yang ditulis sampai akhir baris akan diperlakukan sebagai komentar. Cara ini hanya bisa diterapkan pada komentar satu baris. Jika cara ini akan diterapkan pada komentar beberapa baris, maka pada setiap baris komentar karakter “//” harus ditulis di awal komentar.
2. Karakter “/*” digunakan untuk mengawali penulisan komentar satu baris atau lebih, sampai dijumpai karakter “*/”. Cara ini memungkinkan kita menulis komentar lebih dari satu baris tanpa harus menulis tanda komentar berulang-ulang. Cukup awali komentar dengan menulis “/*” lalu akhiri komentar dengan menulis “*/”

7. Karakter Khusus

Bahasa C menyediakan beberapa karakter khusus, seperti :

- \a : untuk bunyi bell (alert)
- \b : mundur satu spasi (backspace)
- \n : ganti baris baru (new line)
- \r : ke kolom pertama, baris yang sama (carriage return)
- \t : tabulasi
- \0 : nilai kosong (null)
- \' : karakter petik tunggal
- \" : karakter petik ganda
- \\ : karakter garis miring

F. Keselamatan Kerja

1. Ikuti langkah kerja yang ada pada job sheet

2. Perhatikan keamanan instalasi listrik saat menghidupkan dan mematikan computer
3. Pastikan lingkungan kerja (meja, kursi dan computer) bersih, baik sebelum maupun sesudah penggunaan
4. Jika ada hal yang kurang jelas, tanyakan kepada dosen pengampu

G. Langkah Kerja

1. Program 1

- a) Bukalah aplikasi Bahasa C pada computer
- b) Ketikkan program berikut :

```
/*PROGRAM SEDERHANA UNTUK MENCETAK PESAN KE DALAM LAYAR*/
#include <stdio.h>

int main()
{
    //perintah untuk mencetak ke layar
    printf("HAI\n");
    printf("AYO BELAJAR BAHASA C");
    //nilai kembalian dari fungsi main
    return 0;
}
```

- c) Simpan pekerjaan Anda dengan nama prak1_1.c
- d) Lakukan kompilasi dan jalankan program
- e) Perhatikan hasil program Anda

2. Program 2

- a) Ketikkan program berikut :

```
#include <stdio.h>

int main()
{
    //mencetak string
    printf("Nim Mahasiswa\t: %s\n", "0410960044");
    printf("Nama Mahasiswa\t: %s\n", "Nadine");

    //mencetak integer
    printf("Umur \t\t: %d\n", 18);

    //mencetak float dan char
    printf("Nilai\t\t: %f, nilai huruf : %c\n", 90.5, 'A');
    return 0;
}
```

- b) Simpan pekerjaan Anda dengan nama prak1_2.c
- c) Lakukan kompilasi dan jalankan program
- d) Perhatikan hasil program Anda

H. Evaluasi

Kerjakan soal berikut dan kumpulkan hasilnya dalam bentuk hard copy pada pertemuan berikutnya!

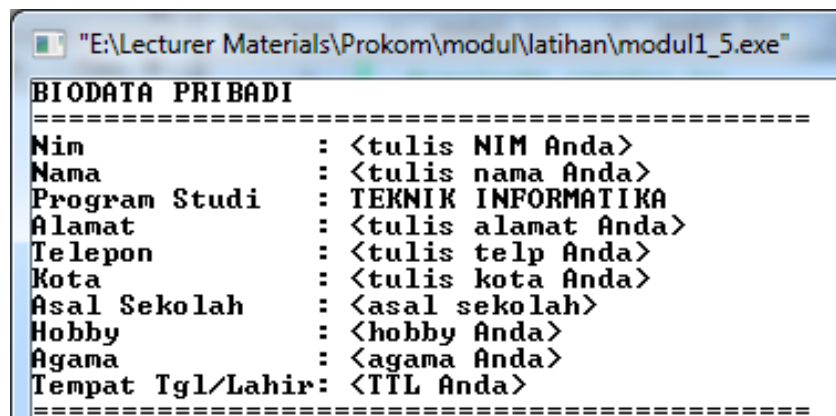
1. Cobalah cek kesalahan dari program berikut ini :

```
#include <stdio.h>

int Main()
{
    /Error Checking
    printf("%d Tali %d Uang\n", 1, 3);
    Printf("Dibawah ini adalah sebuah pepatah\n");
    printf("Ada %c Ada %s\n", "Gula", "Semut");
    return(0);
}
```

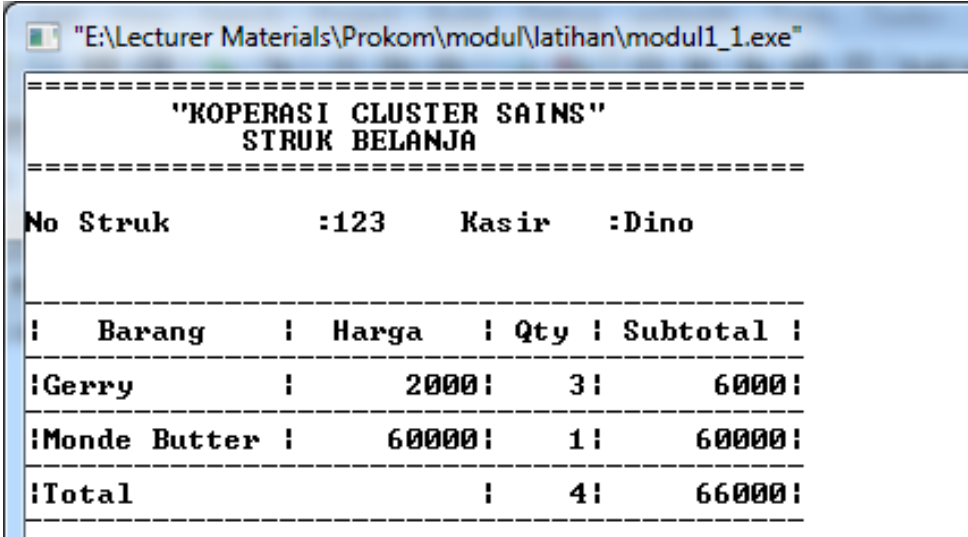
Q1. Apa saja kesalahan yang Anda temukan? Jelaskan penyebab kesalahan tersebut lalu tuliskan program yang benar!

2. Cobalah untuk membuat program yang bisa menampilkan output berikut :



```
BIODATA PRIBADI
=====
Nim           : <tulis NIM Anda>
Nama          : <tulis nama Anda>
Program Studi : TEKNIK INFORMATIKA
Alamat        : <tulis alamat Anda>
Telepon       : <tulis telp Anda>
Kota          : <tulis kota Anda>
Asal Sekolah  : <asal sekolah>
Hobby         : <hobby Anda>
Agama        : <agama Anda>
Tempat Tgl/Lahir: <TTL Anda>
=====
```

3. Buatlah program untuk mencetak struk belanja seperti di bawah ini. Nomor Struk, Nama Kasir dan Item yang dibeli tidak boleh sama dengan contoh, silakan diganti sendiri.



The screenshot shows a Windows command prompt window with the title bar "E:\Lecturer Materials\Prokom\modul\latihan\modul1_1.exe". The output of the program is a receipt for "KOPERASI CLUSTER SAINS". It displays the receipt number as 123 and the cashier as Dino. Below this, there is a table of items purchased: Gerry (3 units at 2000 each, subtotal 6000), Monde Butter (1 unit at 60000, subtotal 60000), and a total of 4 units for 66000.

| "KOPERASI CLUSTER SAINS" | | | | |
|--------------------------|--------------|-----|-------|----------------|
| STRUK BELANJA | | | | |
| No Struk | : | 123 | Kasir | :Dino |
| ===== | | | | |
| | Barang | | Harga | Qty Subtotal |
| | Gerry | | 2000 | 3 6000 |
| | Monde Butter | | 60000 | 1 60000 |
| | Total | | | 4 66000 |

| | | |
|--|---|--|
| POLITEKNIK NEGERI BALI | Job Sheet 2 Tipe Data, Variabel, dan Konstanta | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 3 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 2 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami dan menggunakan berbagai tipe data dalam Bahasa C.

C. Indikator

1. Mahasiswa mampu menerapkan tipe data dalam Bahasa C
2. Mahasiswa mampu menggunakan sintaks keluaran dan masukan dengan berbagai tipe data

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

1. Tipe Data Dasar

Data merupakan suatu nilai yang bisa dinyatakan dalam bentuk konstanta atau variabel. Konstanta menyatakan nilai yang tetap, sedangkan variabel menyatakan nilai yang dapat diubah-ubah selama eksekusi berlangsung.

Data berdasarkan jenisnya dapat dibagi menjadi 5 kelompok, yang dapat dinyatakan sebagai tipe data dasar. Kelima tipe data dasar tersebut adalah sebagai berikut :

| Tipe | Total Bit | Kapasitas | Keterangan |
|--------|-----------|-----------------------|-----------------------------|
| Char | 8 | -128 s/d 127 | Karakter |
| Int | 16 | -32768 s/d 32767 | Bil. integer |
| Float | 32 | 3.4E-38 s/d 3.4E+38 | Bil. Real (presisi tunggal) |
| Double | 64 | 1.7E-308 s/d 1.7E_308 | Bil. Real (presisi ganda) |
| Void | 0 | - | Tak bertipe |

Untuk tipe data short int, long int, signed int dan unsigned int, maka ukuran memori yang diperlukan serta kawasan dari masing-masing tipe data adalah sebagai berikut:

| Tipe | Total bit | Kawasan | Keterangan |
|--------------|-----------|----------------------------|----------------------------|
| short int | 16 | -32768 s/d 32767 | short integer |
| long int | 32 | -2147483648 s/d 2147483647 | long integer |
| signed int | 32 | -2147483648 s/d 2147483647 | biasa disingkat dengan int |
| unsigned int | 32 | 0 s/d 4294967295 | bilangan int tak bertanda |

Catatan :

Ukuran dan kawasan dari masing-masing tipe data adalah bergantung pada jenis mesin yang digunakan (misalnya mesin 16 bit bisa jadi memberikan hasil berbeda dengan mesin 32 bit)

2. Variabel

Variabel adalah suatu pengenal (identifier) yang digunakan untuk mewakili suatu nilai tertentu di dalam proses program. Berbeda dengan konstanta yang nilainya selalu tetap, nilai dari suatu variable bisa diubah-ubah sesuai kebutuhan.

Nama dari suatu variable dapat ditentukan sendiri oleh pemrogram dengan aturan sebagai berikut :

- Pengenal harus diawali dengan huruf (A..Z, a..z) atau karakter garis bawah (_).
- Selanjutnya dapat berupa huruf, digit (0..9) atau karakter garis bawah atau tanda dollar (\$).
- Panjang pengenal boleh lebih dari 31 karakter, tapi hanya 31 karakter pertama yang akan dianggap berarti.
- Pengenal tidak boleh menggunakan nama yang tergolong sebagai kata-kata cadangan (reserved words) seperti int, if, while, dan sebagainya.

2.1 Mendeklarasikan Variabel

Variabel digunakan dalam program untuk menyimpan suatu nilai, dan nilai yang ada padanya dapat diubah-ubah selama eksekusi program berlangsung. Variabel yang akan digunakan dalam program harus dideklarasikan terlebih dahulu. Pengertian deklarasi disini berarti memesan memori dan menentukan jenis data yang bisa disimpan di dalamnya. Bentuk umum deklarasi variabel adalah sebagai berikut:

```
tipe daftar-variabel;
```

Pada pendeklarasian variabel, daftar variabel dapat berupa variabel atau beberapa variabel yang dipisahkan dengan koma.

Contoh:

```
Int var_bulat1;
```

```
Float var_pecahan1, var_pecahan2;
```

2.2 Memberikan nilai ke variabel

Untuk memberikan nilai ke variabel yang telah dideklarasikan, maka bentuk umum pernyataan yang digunakan adalah sebagai berikut:

```
nama_variabel = nilai;
```

Contoh:

```
int var_bulat = 10;
```

```
double var_pecahan = 10.5;
```

2.3 Inisialisasi variabel

Adakalanya dalam penulisan program, setelah dideklarasikan variabel langsung diberi nilai awal. Sebagai contoh yaitu variabel nilai:

```
int nilai;
```

```
nilai = 10;
```

Dua pernyataan diatas sebenarnya dapat disingkat melalui pendeklarasian

yang disertai penugasan nilai, sebagai berikut:

```
int nilai = 10;
```

3. Konstanta

Konstanta menyatakan nilai yang tetap. Berbeda dengan variabel, suatu konstanta tidak dideklarasikan. Namun seperti halnya variabel, konstanta juga memiliki tipe. Penulisan konstanta mempunyai aturan tersendiri, sesuai dengan tipe masing-masing.

- Konstanta karakter misalnya ditulis dengan diawali dan diakhiri dengan tanda petik tunggal, contohnya: 'A' dan '@'.
- Konstanta integer ditulis dengan tanda mengandung pemisah ribuan dan tak mengandung bagian pecahan, contohnya : -1 dan 32767.
- Konstanta real (float dan double) bisa mengandung pecahan (dengan tanda berupa titik) dan nilainya bisa ditulis dalam bentuk eksponensial (menggunakan tanda e) contohnya: 27.5 f (untuk

tipe float) atau 27.5 (untuk tipe double) dan 2.1e+5 (maksudnya 2,1x10⁵)

- Konstanta string merupakan deretan karakter yang diawali dan diakhiri dengan tanda petik ganda ("), contohnya : "Pemrograman Dasar C"

4. Fungsi printf()

Bentuk umum pernyataan printf():

```
printf("string kontrol", argumen1, argumen 2,...)
```

String kontrol dapat berupa keterangan yang akan ditampilkan pada layar beserta penentu format (seperti %d, %f, %c). Penentu format dipakai untuk memberi tahu kompiler jenis data yang akan ditampilkan. Argumen sesudah string kontrol (Argumen1, argumen2,...) adalah data yang akan ditampilkan ke layar. argumen ini dapat berupa variabel, konstanta, bahkan ungkapan. Misal:

```
printf("%d",20); //argumen berupa konstanta
printf("%d,a"); //argumen berupa variabel
printf("%d",a+20); //argumen berupa ungkapan
```

Berikut ini adalah tabel yang berisi penentu format pada printf() :

| | |
|----|--|
| %u | untuk menampilkan data bilangan tak bertanda (<i>unsigned</i>) dalam bentuk desimal. |
| %d | untuk menampilkan bilangan integer bertanda (<i>signed</i>) dalam bentuk desimal |
| %i | |
| %o | untuk menampilkan bilangan bulat tak bertanda dalam bentuk oktal. |
| %x | untuk menampilkan bilangan bulat tak bertanda dalam bentuk heksadesimal |
| %X | (%x → notasi yang dipakai : a, b, c, d, e dan f sedangkan %X → notasi yang dipakai : A, B, C, D, E dan F) |
| %f | untuk menampilkan bilangan real dalam notasi : dddd.ddddd |
| %e | untuk menampilkan bilangan real dalam notasi eksponensial |
| %E | |
| %g | untuk menampilkan bilangan real dalam bentuk notasi seperti %f, %E atau %F |
| %G | bergantung pada kepresisian data (digit 0 yang tak berarti tak akan ditampilkan) |
| l | merupakan awalan yang digunakan untuk %d, %u, %x, %X, %o untuk menyatakan long int (misal %ld). Jika diterapkan bersama %e, %E, %f, %F, %g atau %G akan menyatakan <i>double</i> |
| L | Merupakan awalan yang digunakan untuk %f, %e, %E, %g dan %G untuk menyatakan <i>long double</i> |
| h | Merupakan awalan yang digunakan untuk %d, %i, %o, %u, %x, atau %X, untuk menyatakan <i>short int</i> . |

5. Fungsi scanf()

Fungsi ini digunakan untuk memasukkan berbagai jenis data. Misalnya untuk memasukkan data jari-jari lingkaran pada program menghitung luas lingkaran.

```
scanf("%f", &radius);
```

Bentuk scanf() sesungguhnya menyerupai printf(). Fungsi ini melibatkan penentu format yang pada dasarnya sama digunakan pada printf(). Secara umum bentuk scanf() adalah sebagai berikut:

- scanf("string kontrol", daftar argumen);

F. Keselamatan Kerja

1. Ikuti langkah kerja yang ada pada job sheet
2. Perhatikan keamanan instalasi listrik saat menghidupkan dan mematikan computer
3. Pastikan lingkungan kerja (meja, kursi dan computer) bersih, baik sebelum maupun sesudah penggunaan
4. Jika ada hal yang kurang jelas, tanyakan kepada dosen pengampu

G. Langkah Kerja

1. Program 1

- a. Bukalah aplikasi Bahasa C pada computer
- b. Ketikkan program berikut :

```
//PROGRAM SEDERHANA UNTUK MENGENAL TIPE DATA
#include <stdio.h>
int main()
{
    int arus, hambatan, tegangan;
    arus = 10;
    hambatan = 500;
    tegangan = arus*hambatan;
    printf("Voltase = %d", tegangan);
    return 0;
}
```

- c. Simpan pekerjaan Anda dengan nama prak2_1.c
- d. Lakukan kompilasi dan jalankan program
- e. Perhatikan hasil program Anda

2. Program 2

- a. Ketikkan program berikut :

```
//PROGRAM SEDERHANA UNTUK MENGENAL TIPE DATA
#include <stdio.h>
int main()
{
    char nama_komponen[15];
    int jumlah, harga, bayar;
    printf("Nama      komponen      =      ");    scanf("%s",
        &nama_komponen);
    printf("Harga Komponen = "); scanf("%d", &harga);
    printf("jumlah beli = "); scanf("%d",&jumlah);
    bayar=jumlah*harga;
    printf("Yang harus dibayar sejumlah = %d\n\n",
```

```
        bayar);  
    return 0;  
}
```

- b. Simpan pekerjaan Anda dengan nama prak2_2.c
- c. Lakukan kompilasi dan jalankan program
- d. Perhatikan hasil program Anda

H. Evaluasi

Kerjakan soal berikut dan kumpulkan hasilnya dalam bentuk hard copy pada pertemuan berikutnya!

Buatlah program untuk:

1. Mencari sisi miring segitiga siku-siku berdasarkan input alas dan tinggi, dimana tinggi dan alas adalah bilangan bulat.
2. Mencari keliling dan luas lingkaran berdasarkan input jari-jarinya, dengan jari-jari berupa bilangan bulat
3. Mencari volume kubus berdasarkan input sisinya dengan sisi berupa bilangan riil

| | | |
|--|---|--|
| POLITEKNIK NEGERI BALI | Job Sheet 3 Operator | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 6 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 3 dan 4 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami dan menggunakan berbagai tipe data dan operator dalam Bahasa C.

C. Indikator

1. Mahasiswa mampu menerapkan berbagai operator dalam Bahasa C
2. Mahasiswa mampu menjelaskan keluaran dari program C yang melibatkan berbagai tipe data dan operator

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

Operator merupakan simbol atau karakter yang biasa dilibatkan dalam program untuk melakukan sesuatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai ke suatu variabel, membandingkan kesamaan dua buah nilai. Sebagian operator C tergolong sebagai operator binary, yaitu operator yang dikenakan terhadap dua buah nilai (*operand*). Contoh :

$a + b$

Simbol + merupakan operator untuk melakukan operasi penjumlahan dari kedua *operandnya* (yaitu a dan b). Karena operator penjumlahan melibatkan dua operator ini tergolong sebagai operator binary.

$-c$

Simbol - (minus) juga merupakan operator. Simbol ini termasuk sebagai operator unary, yaitu operator yang hanya memiliki sebuah operand (yaitu c pada contoh ini).

1. Operator Aritmatika

- Operator untuk operasi aritmatika yang tergolong sebagai operator binary adalah :
 - * : perkalian
 - / : pembagian
 - % : sisa pembagian
 - + : penjumlahan
 - : pengurangan
- Adapun operator yang tergolong sebagai operator unary.
 - : tanda minus
 - + : tanda plus

Contoh pemakaian operator aritmatika misalnya untuk memperoleh volume balok : panjang * lebar * tinggi

2. Operator Penurunan dan Peningkatan

Masih berkaitan dengan operasi aritmatika, C menyediakan operator yang disebut sebagai operator peningkatan dan operator penurunan, yaitu :

- ++ : operator peningkatan
- : operator penurunan

Operator peningkatan digunakan untuk menaikkan nilai variabel sebesar satu. Penempatan operator terhadap variabel dapat dilakukan di muka atau di belakangnya.

```
x = x+1;  
y = y-1;
```

Bisa ditulis menjadi :

```
++x;  
--y;
```

atau :

```
x++;  
y--;
```

bergantung pada kondisi yang dibutuhkan oleh pemrogram. Misalnya :

```
b = 2;  
a = b++ *3;
```

akan berbeda dengan

```
b = 2;  
a = ++b *3;
```

Perbedaan terletak pada nilai akhir a, dimana pada kasus pertama operasi perkalian dilakukan dahulu baru nilai b dinaikkan, sehingga nilai a adalah 75, sedangkan pada kasus kedua, nilai b dinaikkan dulu sebesar 1, baru dilakukan operasi perkalian, sehingga nilai a adalah 78.

3. Operator Penugasan

Operator penugasan (*assignment operator*) digunakan untuk memindahkan nilai dari suatu ungkapan (*expression*) ke suatu pengenal. Operator pengerjaan yang umum digunakan dalam bahasa pemrograman, termasuk bahasa C adalah operator sama dengan (=). Contohnya:

fahrenheit = celcius * 1.8 + 32;

Maka '=' adalah operator penugasan yang akan memberikan nilai dari ungkapan : celcius * 1.8 + 32 kepada variabel fahrenheit.

Bahasa C juga memungkinkan dibentuknya statemen penugasan menggunakan operator pengerjaan jamak dengan bentuk sebagai berikut :

pengenal1 = pengenal2 = ... = ungkapan ;

Misalnya :

a = b = 15;

maka nilai variabel 'a' akan sama dengan nilai variabel 'b' akan sama dengan 15.

4. Operator Kombinasi (Pemendekan)

C menyediakan operator yang dimaksudkan untuk memendekkan penulisan operasi penugasan semacam

x = x + 2;

y = y * 4;

menjadi

x += 2;

y *= 4;

Daftar berikut memberikan seluruh kemungkinan operator kombinasi dalam suatu pernyataan serta pernyataan padanannya.

| | |
|----------|-----------------------------|
| x += 2; | kependekan dari x = x + 2; |
| x -= 2; | kependekan dari x = x - 2; |
| x *= 2; | kependekan dari x = x * 2; |
| x /= 2; | kependekan dari x = x / 2; |
| x %= 2; | kependekan dari x = x % 2; |
| x <<= 2; | kependekan dari x = x << 2; |
| x >>= 2; | kependekan dari x = x >> 2; |
| x &= 2; | kependekan dari x = x & 2; |
| x = 2; | kependekan dari x = x 2; |
| x ^= 2; | kependekan dari x = x ^ 2; |

5. Prioritas Operator Aritmatika

Tabel di bawah ini memberikan penjelasan mengenai prioritas dari masing-masing operator. Operator yang mempunyai prioritas tinggi akan diutamakan dalam hal pengerjaan dibandingkan dengan operator yang memiliki prioritas lebih rendah.

| PRIORITAS | OPERATOR | URUTAN Pengerjaan |
|-----------|----------------------------|----------------------------------|
| Tertinggi | () | dari kiri ke kanan |
| | ! ++ -- + - | dari kanan ke kiri ^{*)} |
| | * / % | dari kiri ke kanan |
| | + - | dari kiri ke kanan ^{*)} |
| Terendah | = += -= *= /= %= | dari kanan ke kiri |

*) Bentuk **unary +** dan **unary -** memiliki prioritas yang lebih tinggi daripada bentuk **binary +** dan **binary -**

6. Operator Relasional

Operator relasional adalah operator yang mengecek hubungan antara 2 buah operand. Nilai kembalian dari operator ini adalah Boolean.

| Operator | Arti | Contoh |
|----------|------------------------------|--------------|
| == | Sama dengan | 3==2 (false) |
| > | Lebih besar | 2>3 (false) |
| < | Lebih kecil | 2<3 (true) |
| >= | Lebih besar atau sama dengan | 2>=2 (true) |
| <= | Lebih kecil atau sama dengan | 4<=3 (false) |
| != | Tidak sama dengan | 1!=2 (true) |

7. Operator Logika

Operator logika merupakan operator yang digunakan untuk menggabungkan pernyataan yang mengandung operator relasional.

| Operator | Arti | Contoh |
|----------|-------------|---------------------------|
| && | Logical And | ((3==2) && (2<3)) (false) |
| | Logical Or | ((3==2) (2<3)) (true) |
| ! | Logical Not | !(5>3) (false) |

8. Operator Bitwise

Operator ini beroperasi pada nilai bit data. Operator ini digunakan pada pemrograman level bit.

| Operator | Arti | Contoh |
|----------|---------------------|-----------------------------------|
| & | Bitwise AND | 2 & 3 (10 & 11 = 10) bernilai 2 |
| | Bitwise OR | 2 & 3 (10 11 = 11) bernilai 3 |
| ^ | Bitwise XOR | 2 ^ 3 (10 ^ 11 = 01) bernilai 1 |
| << | Bitwise SHIFT LEFT | 2 << 1 (10 << 1 = 100) bernilai 4 |
| >> | Bitwise SHIFT RIGHT | 2 >> 1 (10 >> 1 = 1) bernilai 1 |

F. Keselamatan Kerja

1. Ikuti langkah kerja yang ada pada job sheet
2. Perhatikan keamanan instalasi listrik saat menghidupkan dan mematikan computer
3. Pastikan lingkungan kerja (meja, kursi dan computer) bersih, baik sebelum maupun sesudah penggunaan
4. Jika ada hal yang kurang jelas, tanyakan kepada dosen pengampu

G. Langkah Kerja

a. Program 1

1. Bukalah aplikasi Bahasa C pada computer
2. Ketikkan program berikut :

```
//Menghitung keliling dan luas lingkaran
#include <stdio.h>
#define pi 3.14
void main()
{
    double radius, keliling, luas;
    printf("Masukkan jari-jari lingkaran : ");
    scanf("%lf",&radius);
    keliling = 2 * pi * radius;
    luas = 0.5 * pi * radius * radius;
    printf("\nData lingkaran\n");
    printf("Jari-jari = %8.2lf\n", radius);
    printf("Keliling = %8.2lf\n", keliling);
    printf("Luas = %8.2lf\n", luas);
}
```

3. Simpan pekerjaan Anda dengan nama prak3_1.c
4. Lakukan kompilasi dan jalankan program
5. Perhatikan hasil program Anda

b. Program 2

1. Ketikkan program berikut :

```
#include <stdio.h>
//#include <math.h>

void main()
{
    int a=1, b=5,c;

    c=a==b;

    printf("\n\n");
    printf("a = %d\n", a);
    printf("b = %d\n", b);
    printf("c = %d\n", c);
}
```

2. Simpan pekerjaan Anda dengan nama prak3_2.c
3. Lakukan kompilasi dan jalankan program
4. Perhatikan hasil program Anda

Q1. Berapakah output nilai c? Mengapa outputnya demikian?

Q2. Bagaimana jika pernyataan `c=a==b` diganti menjadi `c=a=b`? Apakah outputnya berbeda? Mengapa?

c. Program 3

1. Ketikkan program berikut :

```
#include <stdio.h>

void main()
{
    int a=7, b=5,c;

    c = a++ % --b;

    printf("\n\n");
    printf("a = %d\n", a);
    printf("b = %d\n", b);
    printf("c = %d\n", c);
}
```

2. Simpan pekerjaan Anda dengan nama prak3_3.c

3. Lakukan kompilasi dan jalankan program
4. Perhatikan hasil program Anda

Q3. Jelaskanlah mengapa output dari a = 8, b = 4 dan c=3!

Q4. Jika pernyataan c = a++ % --b diganti dengan c = ++a % --b, maka berapakah outputnya? Jelaskan mengapa demikian? Lalu bagaimana jika diganti dengan c = a++ % b-- atau c = ++a % b--?

H. Evaluasi

Kerjakan soal berikut dan kumpulkan hasilnya dalam bentuk hard copy pada pertemuan berikutnya!

1. Buatlah program untuk menghitung nilai akar-akar dari sebuah persamaan kuadrat, dimana input adalah nilai a, b dan c dari persamaan $ax^2+bx+c=0$, sedangkan rumus akar persamaan adalah $x_1 = \frac{-b+\sqrt{b^2-4ac}}{2a}$, $x_2 = \frac{-b-\sqrt{b^2-4ac}}{2a}$
Bantuan : Untuk pangkat dan akar, gunakan fungsi pow dengan bentuk pow(bilangan,pangkat)
2. Sebuah perusahaan kurir menetapkan tarif pengiriman sebesar 15000 per kilo untuk 1 kilo pertama, dan selanjutnya sebesar 10000/kg. Buatlah program untuk menerima input berupa total barang yang dikirim (dalam kilo) dan harga pengiriman.
3. Sebuah toko menjual kopi dalam kemasan tas. Buatlah program yang menerima input berupa berat tas (dalam kilo), jumlah pembelian tas, harga kopi per kilo. Keluarkan output berupa total pembayaran kopi, apabila diketahui pajak adalah sebesar 10% dari total belanja.
4. Buatlah program untuk melakukan konversi nilai detik menjadi jam, menit, detik. Misalnya user memberikan input detik sebesar 7300, maka keluaran program adalah 2 jam, 1 menit 40 detik.
5. Sebuah toko kopi menjual kopi dalam kemasan 2kg saja, yaitu seharga 50000 per kemasan. Ketika customer melakukan pemesanan, maka kopi yang dipesan akan dikirim dalam bentuk box. Toko memiliki 3 jenis box, yaitu box besar mampu menampung 20 kemasan, box sedang mampu menampung 10 kemasan, dan box kecil mampu menampung 5 kemasan. Harga box besar adalah 10000, box sedang 7500, dan box kecil adalah 5000. Dalam melakukan packing, toko menggunakan aturan bahwa box besar dan box sedang tidak boleh memiliki ruang kosong (harus penuh), sedangkan box kecil boleh saja kosong. Toko ingin meminimalkan penggunaan box, dengan cara melakukan pemilihan box mulai dari box terbesar dulu baru medium dan selanjutnya box terkecil. Misalnya jika customer memesan 52 kemasan kopi, maka akan digunakan 2 box besar, 1 box sedang dan 1 box kecil. Buatlah program untuk menerima input berupa total kemasan yang dipesan oleh seorang customer dan

menentukan berapa jumlah masing-masing tas dan biaya total yang harus dibayar, yaitu harga box dan harga kopi yang dibeli

| | | |
|--|---|--|
| POLITEKNIK NEGERI BALI | Job Sheet 4 Struktur Kendali Seleksi | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 12 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 5, 6, 7 dan 8 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami dan menerapkan percabangan dalam Bahasa C.

C. Indikator

1. Mahasiswa mampu memahami bahwa kasus yang diberikan adalah kasus percabangan atau bukan.
2. Mahasiswa mampu menggunakan perintah if dalam menyelesaikan kasus percabangan yang diberikan.
3. Mahasiswa mampu menggunakan perintah switch-case dalam menyelesaikan kasus percabangan yang diberikan.
4. Mahasiswa mampu memahami perbedaan perintah if dan switch-case.
5. Mampu menggunakan percabangan bersarang dalam sebuah program

D. Peralatan

1. Perangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

1. Struktur Penyeleksian

Seringkali instruksi - instruksi dilaksanakan bila suatu persyaratan dipenuhi, dalam struktur penyeleksian, suatu instruksi atau sekelompok instruksi dapat dilewati, sementara instruksi lainnya dijalankan.

a. Konstruksi if

Sintaks :

```
if (ekspresi boolean) statement;
```

Atau

```
If (ekspresi boolean){  
    statement1;  
    statement2;} Blok statement  
}
```

*Jika ekspresi boolean menghasilkan nilai TRUE, maka statement atau blok statement akan dilaksanakan.

b. Konstruksi if-else

Sintaks :

```
if (ekspresi boolean) statement1;  
else statement2;
```

Atau

```
if (ekspresi boolean){  
    statement1;  
    statement2;} Blok statement1  
}  
else  
{ statement3;  
  statement4;} Blok statement2  
}
```

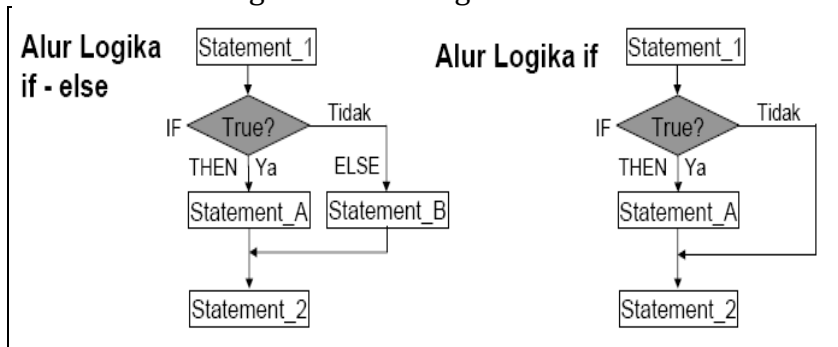
*Jika ekspresi boolean menghasilkan nilai TRUE, maka statement1 dan statement2 atau blok statement1 yang akan dilaksanakan, jika tidak (FALSE) maka statement3 dan statement4 atau blok statement2 yang akan dilaksanakan.

c. Konstruksi if-else dapat digunakan secara bertingkat (*nested*)

Sintaks:

```
if (ekspresi boolean1) statement1;  
else if (ekspresi boolean2) statement2;  
else if (ekspresi boolean3) statement3;  
.....  
else statementN;
```

Perbedaan alur logika if-else dengan if



d. Konstruksi switch-case

Konstruksi ini digunakan sebagai pengganti konstruksi if-else, bila konstruksi if-else bertingkat terlalu jauh, sehingga menjadi sulit dibaca.

Sintaks konstruksi switch-case:

```
switch (ekspresi)
{
    case constant_1 :
        statements_1; break;
    case constant_2 :
        statements_2; break;
    ...
    case constant_N :
        statements_N; break;
    default :
        statements;
}
```

F. Langkah Kerja

1. Program 1

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
#include <stdio.h>
int main()
{
    char nama[12];
    int nilai;
    printf("Program Mengetes Kondisi \n");
    printf("===== \n");
    printf("Masukkan Nilai="); scanf("%d", &nilai);
    if (nilai>=60)
    {
        printf("LULUS BAIK \n");
    }
    if ((nilai>=0) && (nilai<=59))
    {
        printf("TIDAK LULUS \n");
    }
}
```

- b. Simpanlah file Anda dengan nama prak4_1.c
c. Compile dan jalankan, lalu amati hasilnya!
d. Apakah yang dilakukan program di atas?

Q1. Apakah perbedaannya jika pernyataan if ((nilai>=0) && (nilai<=59)) diganti menjadi else if ((nilai>=0) && (nilai<=59))? Jelaskan manakah yang lebih efisien!

2. Program 2

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
#include <stdio.h>
void main()
{
    int paket,bayar,berat;

    printf("Jenis Pengiriman");
    printf("1. Dokumen");
    printf("\n2. Barang");
    printf("\nJenis Paket : ");scanf("%d",&paket);
    printf("\nBerat : ");scanf("%d",&berat);

    switch (paket) {
        case 1 :
            bayar = 10000*berat;
            printf("%d",bayar);
            break;
        case 2 :
            bayar = 15000*berat;
            printf("%d",bayar);
            break;
        default :
            printf("\nPilihan jenis barang salah");
    }
}
```

- b. Simpanlah file Anda dengan nama prak4_2.c
c. Compile dan jalankan, lalu amati hasilnya!
d. Apakah yang dilakukan program di atas?

3. Program 3

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
//Menghitung akar persamaan kuadrat
#include <stdio.h>
#include <math.h>
float a,b,c,d,x1,x2;
int main()
{
    /* Memasukkan data koef. a, b, dan c */
    printf("Masukan koef a : "); scanf("%f",&a);
    printf("Masukan koef b : "); scanf("%f",&b);
    printf("Masukan koef c : "); scanf("%f",&c);
    /* Menghitung Diskriminant */
    d = b*b - 4 * a * c;
```

```
printf("d = %f\n", d);
/* Menghitung x1 dan x2 serta
Menampilkan hasil */
if (d >= 0)
{
    x1 = (-b + sqrt(d)) / (2 * a);
    x2 = (-b - sqrt(d)) / (2 * a);
    printf("x1=%f\n x2=%f\n", x1, x2);
}
else
{
    printf("Akar Persamaan Imajiner");
}
}
```

- b. Simpanlah file Anda dengan nama prak4_3.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

Q2. Apakah kasus 2.3 bisa diselesaikan dengan case? Jelaskan jawaban Anda!

G. Evaluasi

1. Buatlah sebuah program untuk mengkonversi nilai ujian (nilai angka) menjadi nilai huruf

Tabel Nilai

| Nilai Akhir | Nilai Huruf |
|-------------|-----------------|
| 80 – 100 | A : Sangat Baik |
| 70 – 79 | B : Baik |
| 60 – 69 | C : Cukup |
| 0 – 59 | G : Gagal |

2. Buatlah sebuah program untuk menentukan nama hari berdasarkan inputan, contohnya sebagai berikut:
 - (1 = Senin)
 - (2 = Selasa)
 - (3 = Rabu)
 - (4 = Kamis)
 - (5 = Jumat)
 - (6 = Sabtu)
 - (7 = Minggu)
3. Buatlah program untuk mendeteksi bidang apakah bilangan yang diinputkan adalah bilangan nol, negatif dan positif

4. Buatlah program kalkulator, untuk menghitung dua bilangan yang diinputkan! Gunakan menu untuk memilih proses perhitungan yang dikehendaki user!

```
KALKULATOR

-----> MENU <-----
  1. PENJUMLAHAN
  2. PENGURANGAN
  3. PERKALIAN
  4. PEMBAGIAN

Pilih Nomor (1 - 4) : ...
```

| | | |
|--|--|--|
| POLITEKNIK NEGERI BALI | Job Sheet 5 Struktur Kendali Perulangan | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 15 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 9, 10, 11, 12 dan 13 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami dan menerapkan perulangan dalam Bahasa C untuk menyelesaikan kasus tertentu.

C. Indikator

1. Mahasiswa mampu memahami bahwa kasus yang diberikan adalah kasus perulangan atau bukan.
2. Mahasiswa mampu menggunakan perulangan for untuk menyelesaikan suatu kasus.
3. Mahasiswa mampu menggunakan perulangan while untuk menyelesaikan suatu kasus.
4. Mahasiswa mampu menggunakan perulangan do-while untuk menyelesaikan suatu kasus.
5. Mahasiswa memahami perbedaan antara perulangan for, while dan do-while, dan mampu menentukan bentuk perulangan yang paling tepat pada kasus yang diberikan.
6. Mahasiswa mampu menggunakan break, continue dan goto dalam program Bahasa C.
7. Mahasiswa mampu menggunakan perulangan bersarang untuk menyelesaikan suatu kasus.

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

Mengulang suatu proses merupakan tindakan yang banyak dijumpai dalam pemrograman. Pada semua bahasa pemrograman, pengulangan proses ditangani

dengan suatu mekanisme yang disebut *loop*. Dengan menggunakan *loop*, suatu proses yang berulang misalnya menampilkan tulisan yang sama seratus kali pada layar dapat diimplementasikan dengan kode program yang pendek.

1. Pernyataan *for*

Pernyataan pertama yang digunakan untuk keperluan pengulangan proses adalah pernyataan *for*. Bentuk pernyataan ini :

```
for (ungkapan1; ungkapan2; ungkapan3)
    pernyataan;
```

Kegunaan dari masing-masing ungkapan pada pernyataan *for*.

- Ungkapan1 : digunakan untuk memberikan inisialisasi terhadap variabel pengendali *loop*.
- Ungkapan2 : dipakai sebagai kondisi untuk keluar dari *loop*.
- Ungkapan3 : dipakai sebagai pengatur kenaikan nilai variabel pengendali *loop*.

Ketiga ungkapan dalam *for* tersebut harus dipisahkan dengan tanda titik koma (;). Dalam hal ini pernyataan bisa berupa pernyataan tunggal maupun jamak. Jika pernyataannya berbentuk jamak, maka pernyataan-pernyataan tersebut harus diletakkan di antara kurung kurawal buka ({) dan kurung kurawal tutup (}), sehingga formatnya menjadi :

```
for (ungkapan1; ungkapan2; ungkapan3)
{
    pernyataan;
    pernyataan;
    .
    .
    .
}
```

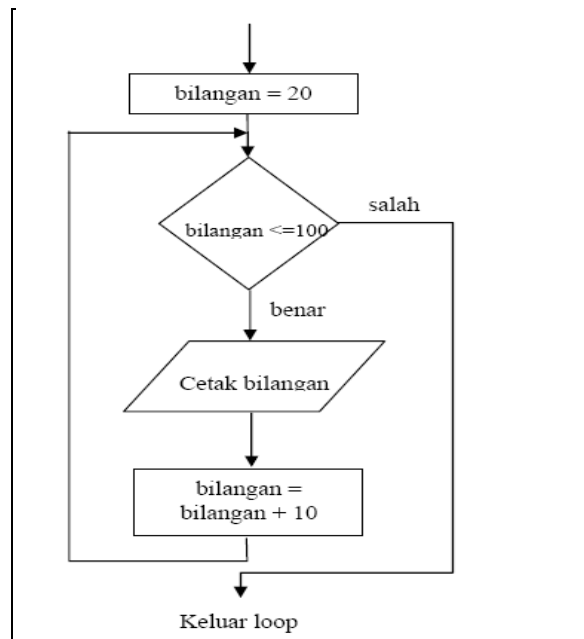
Contoh penggunaan *for*, misalnya untuk menampilkan deretan angka sebagai berikut :

```
20
30
40
50
.
.
.
100
```

Untuk keperluan ini, pernyataan *for* yang digunakan berupa :

```
for (bilangan = 20; bilangan <= 100; bilangan += 10)
    printf("%d\n", bilangan);
```

Kalau digambarkan dalam bentuk diagram alir, akan terlihat sbb :



a. Perulangan positif

Merupakan perulangan yang peningkatannya positif untuk variabel pengontrol perulangannya (misal i++)

```
#include <stdio.h>
int main()
{
    int i;
    for(i=-2;i<=3;i++)
        printf("%d\n",i);
}
```

Keterangan

Pada program di atas, variabel i dimulai dari -2 dan perulangan dihentikan setelah nilai 3 dengan peningkatan nilai 1, sehingga variabel akan bernilai -2,-1,0,1,2 dan 3.

b. Perulangan negatif

Contoh:

```
#include <stdio.h>
main()
{
    int i;
    for (i=5;i>=0;i--)
        printf("%d\n",i);
}
```

Keterangan

Pada program di atas, variabel *i* dimulai dari 5 dan perulangan dihentikan setelah nilai 0 dengan penurunan nilai *i* sebesar 1, sehingga variabel akan bernilai 5, 4, 3, 2 dan 1.

2. Statement while

Format penulisan perulangan dengan **while** adalah sebagai berikut:

while (kondisi) pernyataan

while akan mengecek terlebih dahulu kondisi, jika bernilai benar maka pernyataan akan dijalankan, jika salah maka pernyataan tidak dijalankan.

```
#include<stdio.h>

int main ()
{
    int i, n;
    printf("Masukkan nilai : ");
    scanf("%d",&n);
    i = 0;
    while ( i <= n ) {
        printf("%d , " , i );
        i++;
    }
    return 0;
}
```

Pada program di atas, *i* akan dicetak jika input *n* lebih dari atau sama dengan nol, kemudian nilai *i* akan terus dinaikkan melalui perintah *i++*. Pencetakan nilai *i* akan dihentikan apabila nilai *i* akhirnya telah melebihi nilai input *n*. Perlu diingat, apabila nilai input *n* ternyata di bawah 0, maka tidak akan ada nilai yang dicetak program sama sekali.

3. Statement do-while

Format penulisan perulangan dengan menggunakan **do...while** adalah sebagai berikut:

do pernyataan **while** (kondisi);

Berbeda dengan perulangan **while**, pada perulangan **do-while** ini pernyataan akan dijalankan terlebih dahulu kemudian baru dicek kondisinya jika benar maka akan terus dan jika salah maka akan berhenti.

```
#include <stdio.h>

int main ()
{
    short n;
    long jumlah = 0;
    do {
        printf("Masukkan bilangan (0 untuk keluar) : ");
        scanf("%d", &n);
        jumlah = jumlah + n;
        printf ("Jumlah : %d \n", jumlah);
    } while (n != 0);

    printf("Terimakasih.");
    return 0;
}
```

Pada program di atas, program akan terus meminta input dan menjumlahkan input dari user, sebelum user memberikan input 0. Terlihat pada program bahwa semua perintah di dalam perulangan minimal dieksekusi sekali, baru dilakukan pengecekan untuk menentukan apakah harus keluar dari perulangan atau melanjutkan perulangan.

F. Langkah Kerja

1. Program 1

- Buatlah sebuah file dan salinlah program di bawah ini!

```
#include <stdio.h>

main()
{
    long int total;
    int panjang,i;
    printf("Masukkan panjang deret : ");
    scanf("%d",&panjang);
    total =0;
    for (i=0;i<=panjang;i++)
        total +=pow(2,i);
    printf("Total : %ld",total);
}
```

- Simpanlah file Anda dengan nama prak5_1.c
- Compile dan jalankan, lalu amati hasilnya!
- Apakah yang dilakukan program di atas?

Q1. Apakah yang dilakukan program di atas? Bisakah jika program itu diubah menggunakan bentuk while, dan do-while? Jika bisa, buatlah!

2. Program 2

- Buatlah sebuah file dan salinlah program di bawah ini!


```
#include <stdio.h>

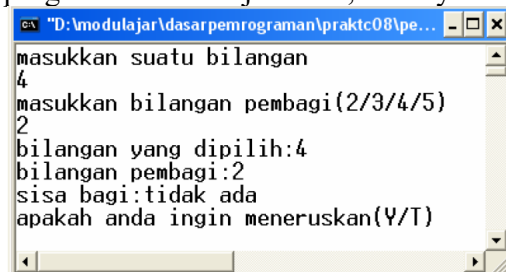
main()
{
    int i,n;
    printf("Masukkan nilai n : ");
    scanf("%d",&n);
    i=0;
    do
    {
        printf("%d, ",i);
        i++;
    }while(i<=n);
}
```

- b. Simpanlah file Anda dengan nama prak5_2.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

Q2. Apakah yang dilakukan program di atas? Apakah perbedaan dan persamaan program tersebut dengan program pada bagian 1.2? Tunjukkan persamaan dan perbedaannya dengan contoh input lalu jelaskan!

G. Evaluasi

1. Buatlah program untuk menampilkan N barisan bilangan ganjil. Input adalah bilangan N. Outputnya adalah : 1,3,5,...,N. Contoh, untuk input N=10, maka outputnya adalah 1,3,5,7,9.
2. Buatlah program untuk menampilkan karakter bintang '*' sejumlah n buah. Input adalah bilangan n. Outputnya adalah : *****...(berjumlah n buah). Contoh, untuk input n = 10, maka outputnya : *****
3. Buatlah program untuk menentukan sisa hasil pembagian antara bilangan yang dimasukkan dengan bilangan pembagi . Apabila sisa baginya=0 maka dicetak tidak ada dan kalau ada sisa baginya, maka sisa bagi tersebut ditampilkan. Bila program tersebut dijalankan, hasilnya adalah sebagai berikut:



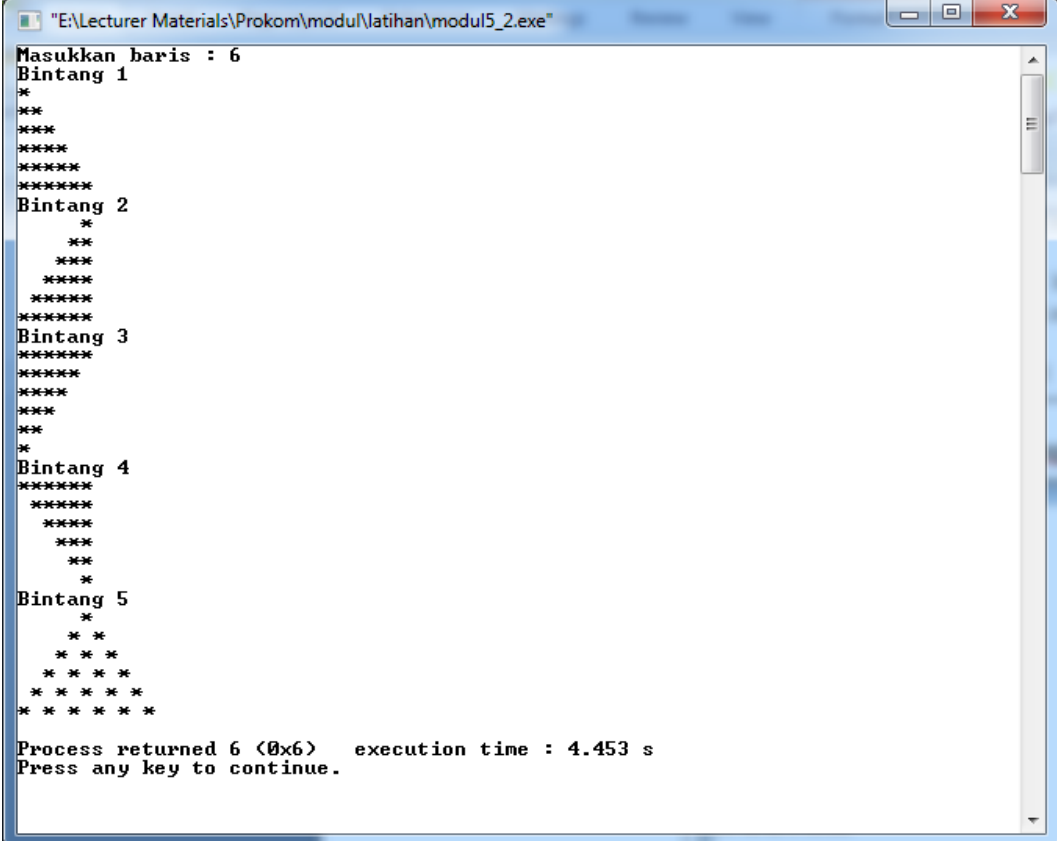
4. Gunakan loop **while** untuk membuat program yang dapat mencari total angka yang dimasukkan dengan tampilan sebagai berikut :

```
Masukkan bilangan ke-1 : 5
Mau memasukkan data lagi [y/t] ? y

Masukkan bilangan ke-2 : 3
Mau memasukkan data lagi [y/t] ? t

Total bilangan = 8
```

5. Buatlah program untuk menampilkan output seperti berikut ini!



```
"E:\Lecturer Materials\Prokom\modul\latihan\modul5_2.exe"
Masukkan baris : 6
Bintang 1
*****
Bintang 2
  *****
Bintang 3
    *****
Bintang 4
      *****
Bintang 5
        *****
*****
Process returned 6 (0x6)   execution time : 4.453 s
Press any key to continue.
```

| | | |
|--|---|--|
| POLITEKNIK NEGERI BALI | Job Sheet 6 Fungsi & Sub Program | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika Jurusan : Teknik Elektro | | Waktu : 12 x 50 Menit |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | Pertemuan 14, 15, 16 dan 17 |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami dan menerapkan fungsi dan sub program dalam Bahasa C untuk membuat program menjadi lebih rapi dan efisien.

C. Indikator

1. Mahasiswa mampu membuat fungsi dalam penyelesaian suatu permasalahan.
2. Mahasiswa mampu menentukan nilai kembalian suatu fungsi
3. Mahasiswa mampu menerapkan parameter dalam suatu fungsi
4. Mahasiswa dapat melakukan pemanggilan fungsi
5. Mahasiswa mampu menjelaskan perbedaan variabel lokal dan global
6. Mahasiswa mampu menggunakan metode melewati argumen dengan tepat
7. Mahasiswa mampu menyusun fungsi rekursi dari suatu permasalahan yang diberikan.

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

1. Fungsi

Fungsi merupakan suatu bagian dari program yang dimaksudkan untuk mengerjakan suatu tugas tertentu dan letaknya terpisah dari program yang memanggilmnya.

Keuntungan menggunakan fungsi :

- Program besar dapat dipisah menjadi program-program kecil.
- Dapat dikerjakan oleh beberapa orang sehingga koordinasi mudah.
- Kemudahan dalam mencari kesalahan-kesalahan karena alur logika jelas dan kesalahan dapat dilokalisasi dalam suatu modul tertentu saja.

- Modifikasi program dapat dilakukan pada suatu modul tertentu saja tanpa mengganggu program keseluruhan.
- Mempermudah dokumentasi.
- Reusability: Suatu fungsi dapat digunakan kembali oleh program atau fungsi lain

Cara mendefinisikan fungsi adalah sebagai berikut :

```
tipe_fungsi nama-fungsi(tipe_data parameter1, tipe_data parameter2,...)
{
    Perintah-perintah dalam fungsi
    .....
    .....
}
```

Dengan :

Tipe_fungsi : void atau tipe data

Parameter : argumen yang dikirimkan dari program pemanggil kepada fungsi

Ingat :

Sebuah fungsi dengan tipe void tidak memberikan nilai kembalian (tidak ada return value), selain itu harus memberikan nilai kembalian yang dinyatakan dalam return. Fungsi tanpa tipe dianggap bertipe integer.

2. Pass by value dan pass by reference

Pengiriman parameter dari program pemanggil kepada fungsi bisa dilakukan melalui 2 cara yaitu :

- Pass by value

Pengiriman parameter hanya mengirimkan nilai dari datanya, bukan alamat memori letak datanya, sehingga perubahan nilai di fungsi tidak akan merubah nilai asli di bagian program yang memanggil fungsi walaupun keduanya menggunakan nama variabel yang sama.

Contoh :

```
void passByValue(int a)
{
    a=10;
}
main()
{
    int a=8;
    passByValue(a);
    printf("Nilai a : %d", a);
}
```

Program di atas menghasilkan output 8, karena perubahan nilai a menjadi 10 hanya terjadi di dalam fungsi dan tidak mempengaruhi nilai a di dalam program utama.

- Pass by reference

Pengiriman parameter dengan mengirimkan memori dari parameter tersebut, bukan nilai dari datanya, sehingga perubahan nilai di fungsi akan merubah nilai asli di bagian program yang memanggil fungsi karena nilai yang diubah di dalam fungsi adalah nilai data pada alamat memori yang dikirim.

Contoh :

```
void passByReference (int *a)
{
    *a=10;
}
main()
{
    int a=8;
    passByReference(&a);
    printf("Nilai a : %d", a);
}
```

Program di atas menghasilkan output 10, karena perubahan nilai a di dalam fungsi berarti memodifikasi nilai variabel dalam memori a yang dikirim dari program utama.

3. Fungsi Rekursi

Rekursi adalah suatu proses dari fungsi yang memanggil dirinya sendiri secara berulang-ulang. Karena proses dilakukan berulang, maka harus ada suatu kondisi yang mengakhiri prosesnya. Jika tidak maka proses tidak terhenti sampai memori yang digunakan tidak dapat menampung lagi

Bentuk umum:

```
int faktorial(int n)
{
    if (n<=1) {
        return 1;
    }
    else {
        return n* faktorial(n-1);
    }
}
```

Kondisi berhenti ketika n adalah 1 atau di bawahnya

Bagian yang memanggil dirinya dengan parameter baru

Penelusuran :

Faktorial(5) = 5* faktorial(5-1) = 5*faktorial(4)
= 5* 4* faktorial(4-1) = 20 * faktorial(3)
= 20 * 3* faktorial(3-1) = 60* faktorial(2)
= 60 * 2* faktorial(2-1) = 120* faktorial(1)
= 120 * 1 = 120

F. Langkah Kerja

1. Program 1

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include <stdio.h>
2
3  void tukar(int bil1,int bil2)
4  {
5      int temp=bil1;
6      bil1=bil2;
7      bil2=temp;
8      printf("Bilangan setelah ditukar :\n");
9      printf("Bilangan 1 : %d\n",bil1);
10     printf("Bilangan 2 : %d\n",bil2);
11
12 }
13
14 main()
15 {
16     int bilangan1, bilangan2;
17     printf("Masukkan bilangan 1 : ");scanf("%d",&bilangan1);
18     printf("Masukkan bilangan 2 : ");scanf("%d",&bilangan2);
19     tukar(bilangan1,bilangan2);
20
21 }
```

- b. Simpanlah file Anda dengan nama prak6_1.c
c. Compile dan jalankan, lalu amati hasilnya!
d. Apakah yang dilakukan program di atas?

2. Program 2

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include<stdio.h>
2
3  int getGaji(int gajiPokok,int bonus,int tunjangan);
4
5  main()
6  {
7
8      //pemanggilan 1
9      printf("Gaji pokok %d, bonus %d, tunjangan %d, sehingga gaji total sebesar %d\n",
10             2500000,5000000,0,getGaji(2500000,500000,0));
11
12     //pemanggilan 2
13     int gp,bns,tunj;
14     printf("Masukkan gaji pokok : ");scanf("%d",&gp);
15     printf("Masukkan bonus : ");scanf("%d",&bns);
16     printf("Masukkan tunjangan : ");scanf("%d",&tunj);
17     printf("Gaji pokok %d, bonus %d, tunjangan %d, sehingga gaji total sebesar %d",
18            gp,bns,tunj,getGaji(gp,bns,tunj));
19 }
20
21 int getGaji(int gajiPokok,int bonus,int tunjangan)
22 {
23     return gajiPokok+bonus+tunjangan;
24 }
```

- b. Simpanlah file Anda dengan nama prak6_2.c
c. Compile dan jalankan, lalu amati hasilnya!
d. Apakah yang dilakukan program di atas?

3. Program 3

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include <stdio.h>
2
3  void tukar(int *bil1, int *bil2);
4
5  main()
6  {
7      int bilangan1, bilangan2;
8      printf("Masukkan bilangan 1 : "); scanf("%d", &bilangan1);
9      printf("Masukkan bilangan 2 : "); scanf("%d", &bilangan2);
10     printf("Bilangan setelah ditukar :\n");
11     tukar(&bilangan1, &bilangan2);
12     printf("Bilangan 1 : %d\n", bilangan1);
13     printf("Bilangan 2 : %d\n", bilangan2);
14 }
15
16 void tukar(int *bil1, int *bil2)
17 {
18     int temp=*bil1;
19     *bil1=*bil2;
20     *bil2=temp;
21 }
22
```

- b. Simpanlah file Anda dengan nama prak6_3.c
c. Compile dan jalankan, lalu amati hasilnya!
d. Apakah yang dilakukan program di atas?

Q1. Apakah perbedaan program a dengan program b? Bagaimana jika tanda bintang dalam fungsi dihilangkan? Jelaskan dengan lebih rinci kegunaan * dan tanda & pada parameter fungsi! Dalam kondisi seperti apakah pass by reference digunakan? Jelaskan dengan contoh selain program penukaran!

4. Program 4

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include <stdio.h>
2
3  int jumlah(int i);
4
5  main()
6  {
7      int bil;
8      do
9      {
10         printf("Masukkan bilangan bulat (minimal 0) : \n");
11         scanf("%d",&bil);
12     }
13     while (bil<0);
14
15     printf("Total : %d",jumlah(bil));
16 }
17
18 int jumlah(int i)
19 {
20     if (i<1)
21         return 0;
22     else
23         return jumlah(i-1)+i;
24 }
25 }
```

- b. Simpanlah file Anda dengan nama prak6_4.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

5. Program 4

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
#include <stdio.h>

int tabung(int saldo);
int tarik(int saldo);
void cekSaldo(int saldo);

main()
{
    int pilihan,saldo=0;
    do
    {
        system("cls");
        printf("\nMenu\n");
        printf("1. Tabung\n");
        printf("2. Tarik\n");
        printf("3. Cek Saldo\n");
        printf("0. Selesai\n");
        printf("Masukkan pilihan : ");
        scanf("%d",&pilihan);
        switch(pilihan)
```



```
        {
            case 1 : saldo=tabung(saldo);break;
            case 2 : saldo=tarik(saldo);break;
            case 3 : cekSaldo(saldo);break;
            case 0 : break;
            default : printf("Pilihan salah!");
        }
        printf("\nTekan sembarang tombol dari keyboard\n");
        getch();
    }
    while (pilihan!=0);
}

int tabung(int saldo)
{
    int jumlahTabung;
    printf("\nMasukkan tabungan : ");
    scanf("%d",&jumlahTabung);
    saldo +=jumlahTabung;
    printf("\nSaldo Anda : %d",saldo);
    return saldo;
}

int tarik(int saldo)
{
    int jumlahTarik;
    printf("\nMasukkan penarikan : ");
    scanf("%d",&jumlahTarik);
    if (jumlahTarik<=saldo)
    {
        saldo -=jumlahTarik;
        printf("\nSaldo Anda : %d",saldo);
    }
    else
    {
        printf("\nSaldo Anda tidak mencukupi. Saldo saat ini
: %d",saldo);
    }
    return saldo;
}

void cekSaldo(int saldo)
{
    printf("\nSaldo Anda : %d",saldo);
}
```

- b. Simpanlah file Anda dengan nama prak6_5.c
 - c. Compile dan jalankan, lalu amati hasilnya!
-

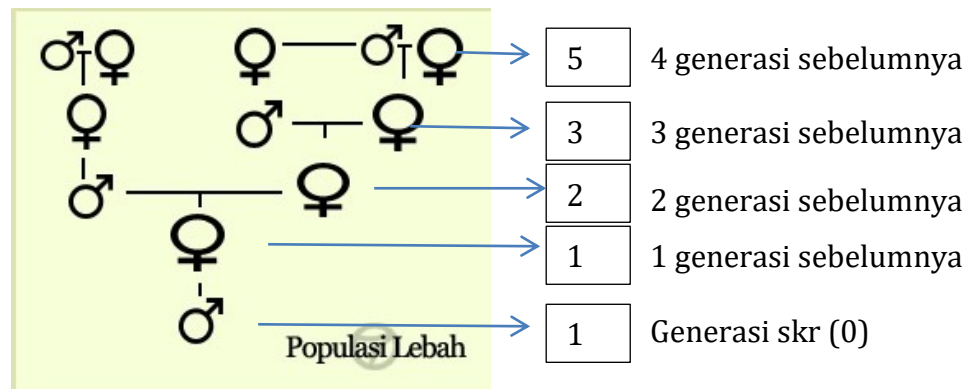
d. Apakah yang dilakukan program di atas?

Q2. Apabila fungsi tabung dan tarik bertipe void (tidak memberikan nilai kembalian), apa yang harus dimodifikasi agar program tetap berjalan dengan benar? Atau tidak ada yang perlu dimodifikasi? Jelaskan jawaban Anda!

G. Evaluasi

1. Buatlah fungsi rekursi untuk kasus berikut :

Lebah adalah hewan yang hidup berkoloni. Ada fakta unik tentang lebah yang tidak diketahui semua orang. Lebah jantan diproduksi dari sel telur ratu yang tidak terbuahi (artinya lebah jantan tidak memiliki Ayah), sedangkan lebah betina dihasilkan dari sel telur yang sudah dibuahi (lebah betina memiliki Ayah dan Ibu). Jika ditelusuri pohon nenek moyang dari seekor lebah jantan, adalah sebagai berikut :



Contoh Program :

```
Masukkan generasi penelusuran nenek moyang : 1
Jumlah lebah pada 1 generasi sebelumnya adalah : 1
```

```
Masukkan generasi penelusuran nenek moyang : 2
Jumlah lebah pada 2 generasi sebelumnya adalah : 2
```

```
Masukkan generasi penelusuran nenek moyang : 4
Jumlah lebah pada 4 generasi sebelumnya adalah : 5
```

```
Masukkan generasi penelusuran nenek moyang : 5
Jumlah lebah pada 5 generasi sebelumnya adalah : 8
```


| | | |
|--|---|--|
| POLITEKNIK NEGERI BALI | Job Sheet 7 Array Satu Dimensi | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika Jurusan : Teknik Elektro | | Waktu : 6 x 50 Menit |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | Pertemuan 18 dan 19 |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu menggunakan variabel berindeks (*array*) satu dimensi dalam Bahasa C.

C. Indikator

1. Mahasiswa mampu melakukan deklarasi array baik satu dimensi
2. Mahasiswa mampu melakukan insialisasi array satu dimensi dengan berbagai cara
3. Mahasiswa mampu melakukan pengaksesan array satu dimensi

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

Variabel array (larik) adalah tipe terstruktur yang terdiri dari sejumlah komponen yang mempunyai tipe yang sama. Suatu array mempunyai jumlah komponen yang tetap. Banyaknya komponen dalam suatu larik ditunjukkan oleh suatu indeks yang berfungsi membedakan anggota dalam suatu array.

Pendeklarasian sebuah array adalah sebagai berikut :

```
Tipe_data nama_variabel[panjang];
```

Dimana :

- Tipe_data : tipe data dari variabel yang ingin diinput
- Nama_variabel : nama variabel array
- panjang : Panjang maksimal variabel array

Contoh :

```
int bilangan[4];
```

Source code di atas adalah source code untuk membuat sebuah array bernama bilangan dengan tipe integer dan panjang 4 elemen.

Pemberian nilai pada array adalah dengan cara memberikan nilai kepada setiap elemen yang ada di dalam array. Adapun pemberian nilai pada suatu elemen adalah sebagai berikut :

```
Nama_variabel[index] =nilai;
```

Dimana

- index : urutan elemen dari array (dimulai dari 0)
- Nilai : nilai yang akan diberikan kepada elemen array ke-index+1

Contoh :

```
bilangan[0]=8;  
bilangan[3]=2;
```

Contoh di atas memberikan nilai 8 kepada elemen array ke-1 dan memberikan nilai 2 pada elemen ke-4.

Pengaksesan elemen array adalah sebagai berikut :

```
Nama_variabel[index];
```

Dimana :

- index : urutan elemen dari array (dimulai dari 0)

Contoh :

```
Printf("%d",bilangan[0]);
```

Contoh di atas akan mencetak array bilangan pada elemen array pertama.

F. Langkah Kerja

1. Program 1

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include <stdio.h>
2
3  main()
4  {
5      int n;
6      printf("Masukkan panjang elemen array : ");
7      scanf("%d", &n);
8
9      //deklarasi array bertipe int dengan panjang n
10     int bil_array[n];
11     int i;
12     //menginput elemen array dengan perulangan
13     for (i=0; i<n; i++)
14     {
15         printf("Masukkan elemen ke-%d : ", i+1);
16         scanf("%d", &bil_array[i]);
17     }
18
19     //mencetak element array dengan perulangan
20     for (i=0; i<n; i++)
21         printf("Elemen ke-%d adalah %d \n", i+1, bil_array[i]);
22
23 }
24
```

- b. Simpanlah file Anda dengan nama prak7_1.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

2. Program 2

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include <stdio.h>
2  main()
3  {
4      int n;
5      printf("Masukkan panjang elemen array : ");
6      scanf("%d", &n);
7
8      //deklarasi array bertipe int dengan panjang n
9      int bil_array[n];
10     int i;
11     //menginput elemen array dengan perulangan
12     for (i=0; i<n; i++)
13     {
14         printf("Masukkan elemen ke-%d : ", i+1);
15         scanf("%d", &bil_array[i]);
16     }
17
18     int cari;
19     printf("Masukkan nilai yang ingin dicari : ");
20     scanf("%d", &cari);
21
22     //melakukan pencarian
23     for (i=0; i<n; i++)
24     {
25         if(cari==bil_array[i])
26         {
27             printf("Data ditemukan");
28             break;
29         }
30     }
31 }
```

- b. Simpanlah file Anda dengan nama prak7_2.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

Q1. Apakah yang terjadi jika `break` pada baris 28 dihilangkan? Apakah terdapat perbedaan pada output? Jika ada, jelaskan! Jika tidak, jelaskan untuk apa `break` tersebut!

G. Evaluasi

1. Buatlah program untuk menginput elemen array dan menentukan elemen terbesar di dalam array!

2. Buatlah program untuk melakukan input array 1 dimensi bertipe integer. Kemudian mintalah user untuk menginput sebuah threshold. Lalu tampilkanlah berapa jumlah data yang nilainya lebih besar daripada threshold!

Contoh :

Input Panjang Data : 4

Data ke-1 : 23

Data ke-2 : 4

Data ke-3 : 7

Data ke-4 : 90

Input Threshold : 7

Jumlah data di atas threshold : 2

3. Buatlah program untuk membalik nilai suatu array. Pastikan nilai array memang terbalik, bukan hanya tampilan saja!

Contoh :

Input Panjang Data : 4

Data ke-1 : 23

Data ke-2 : 4

Data ke-3 : 7

Data ke-4 : 1

Data setelah dibalik

Data ke-1 : 1

Data ke-2 : 7

Data ke-3 : 4

Data ke-4 : 23

4. Buatlah program untuk melakukan pencarian di dalam array, dimana program akan menampilkan indeks array dimana data ditemukan!

Contoh :

Input Panjang Data : 4

Data ke-1 : 23

Data ke-2 : 4

Data ke-3 : 7

Data ke-4 : 4

Input data yang dicari : 4

Data ditemukan pada index : 2, 4

| | | |
|--|--|--|
| POLITEKNIK NEGERI BALI | Job Sheet 8 Array Dua Dimensi | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 3 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 20 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu memahami dan menerapkan percabangan dalam Bahasa C.

C. Indikator

1. Mahasiswa mampu melakukan deklarasi array baik dua dimensi
2. Mahasiswa mampu melakukan insialisasi array dua dimensi dengan berbagai cara
3. Mahasiswa mampu melakukan pengaksesan array dua dimensi

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

Array 2 dimensi biasanya digunakan untuk merepresentasikan tabel. Informasi yang terkandung di dalam array, diatur dalam baris dan kolom.

Pendeklarasian array 2 dimensi adalah sebagai berikut :

```
Tipe_data nama_variabel[baris][kolom];
```

Dimana :

- Tipe_data : tipe data dari variabel yang ingin diinput
- Nama_variabel : nama variabel array
- Baris : Panjang maksimal baris variabel array
- Kolom : Panjang maksimal kolom dari setiap baris array

Contoh :

```
int bilangan[4][3];
```

Source code di atas adalah source code untuk membuat sebuah array 2 dimensi bernama bilangan dengan tipe integer dengan jumlah baris 4, dimana

setiap baris akan terdiri dari masing-masing 3 elemen anggota. Ilustrasinya bisa dinyatakan seperti dalam tabel dengan 4 baris dan 3 kolom seperti berikut :

| | | |
|---------------|---------------|---------------|
| Elemen ke 1,1 | Elemen ke 1,2 | Elemen ke 1,3 |
| Elemen ke 2,1 | Elemen ke 2,2 | Elemen ke 2,3 |
| Elemen ke 3,1 | Elemen ke 3,2 | Elemen ke 3,3 |
| Elemen ke 4,1 | Elemen ke 4,2 | Elemen ke 4,3 |

Pemberian nilai pada array adalah dengan cara memberikan nilai kepada setiap elemen yang ada di dalam array. Adapun pemberian nilai pada suatu elemen adalah sebagai berikut :

```
Nama variabel[row index][col index]=nilai;
```

Dimana

- row_index : urutan baris dari array (dimulai dari 0)
- col_index : urutan kolom dari array elemen utama ke row_index
- Nilai : nilai yang akan diberikan kepada elemen array ke-indeks+1

Contoh :

```
bilangan[0][1]=8;  
bilangan[3][0] = 2;
```

Contoh di atas memberikan nilai 8 kepada elemen array baris ke-1 dan kolom ke-2, dan memberikan nilai 2 pada baris ke-4 dan kolom ke-1.

Pengaksesan elemen array adalah sebagai berikut :

```
Nama variabel[row index][col index];
```

Dimana :

- row_index : urutan baris dari array (dimulai dari 0)
- col_index : urutan kolom dari array elemen utama ke row_index

Contoh :

```
printf("%d",bilangan[0][1]);
```

Contoh di atas akan mencetak array bilangan pada baris pertama kolom ke 2.

F. Langkah Kerja

1. Program 1

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
3  main()
4  {
5      int baris,kolom;
6      printf("Masukkan baris matriks : ");
7      scanf("%d",&baris);
8      printf("Masukkan kolom matriks : ");
9      scanf("%d",&kolom);
10     //deklarasi array bertipe int dengan elemen baris,kolom
11     int matriks[baris][kolom];
12     int i,j;
13     //menginput elemen array dengan perulangan
14     for (i=0;i<baris;i++)
15     {
16         for(j=0;j<kolom;j++)
17         {
18             printf("Masukkan elemen ke[%d,%d] : ",i+1,j+1);
19             scanf("%d",&matriks[i][j]);
20         }
21     }
22     //mencetak element array dengan perulangan
23     printf("Matriks :\n")
24     for (i=0;i<baris;i++)
25     {
26         for(j=0;j<kolom;j++)
27         {
28             printf("%d ",matriks[i][j]);
29         }
30         printf("\n");
31     }
32 }
```

- Simpanlah file Anda dengan nama prak8_1.c
- Compile dan jalankan, lalu amati hasilnya!
- Apakah yang dilakukan program di atas?

G. Evaluasi

- Buatlah program untuk melakukan penambahan 2 buah matriks!
- Buatlah program untuk melakukan transpose matriks!
- Buatlah program untuk mengalikan matriks! Baris dan kolom dari masing-masing matriks diinput oleh user. Ingat syarat perkalian matriks adalah kolom dari matriks pertama harus sama dengan jumlah kolom dari matriks kedua.
- Buatlah program untuk mengubah sebuah array 2 dimensi menjadi array 1 dimensi, dengan cara sebagai berikut :

Diinputkan array 2 dimensi dengan elemen sebagai berikut :

| | | | |
|---|----|---|---|
| 3 | 5 | 2 | 1 |
| 7 | 4 | 1 | 9 |
| 8 | 12 | 0 | 3 |

Cara mengubahnya adalah baca ke samping lalu lanjutkan ke bawah, sehingga bentuk array 1 dimensi adalah :

| | | | | | | | | | | | |
|---------|---|---|---|---------|---|---|---|---------|----|---|---|
| 3 | 5 | 2 | 1 | 7 | 4 | 1 | 9 | 8 | 12 | 0 | 3 |
| Baris 1 | | | | Baris 2 | | | | Baris 3 | | | |

Jumlah baris dan kolom untuk array adalah sesuai dengan input pengguna

| | | |
|--|-------------------------------|--|
| POLITEKNIK NEGERI BALI | Job Sheet 9 String | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 6 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 21 dan 22 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu menggunakan string dan memanipulasinya dalam Bahasa C.

C. Indikator

1. Mahasiswa mampu melakukan deklarasi dan input string
2. Mahasiswa mampu menggunakan fungsi-fungsi string yang disediakan oleh bahasa C

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

String merupakan untai karakter. Dalam pemrograman C, tidak terdapat tipe data string. Di dalam C, string dianggap sebagai sebuah array dengan tipe karakter.

Cara mendeklarasikan sebuah string adalah sebagai berikut :

```
char nama_variabel[panjang];
```

Program C menyediakan beberapa fungsi khusus untuk memanipulasi string. Semua fungsi khusus tersebut berada di dalam library string.h, sehingga untuk menggunakan fungsi-fungsi tersebut, maka pemrogram harus menambahkan library string.h ke dalam programnya.

Beberapa fungsi string yang umum digunakan adalah :

| Fungsi | Kegunaan |
|---------------------|--|
| strcpy(dest,src) | Mengkopi string src ke dalam string dest |
| strncpy(dest,src,n) | Mengkopi string src sepanjang n karakter ke dalam string |

| | |
|------------------|---|
| | dest |
| strcat(s1,s2) | Menggabungkan string s1 dan s2 dengan hasil s1 kemudian dilanjutkan dengan s2. Hasil disimpan pada s1. |
| strncat(s1,s2,n) | Menggabungkan string s1 dan s2 sepanjang n karakter dengan hasil s1 (full) kemudian dilanjutkan dengan s2 sepanjang n karakter. Hasil disimpan pada s1. |
| strcmp(s1,s2) | Membandingkan string s1 dengan s2. Hasilnya 0 jika sama, dan bukan nol, jika berbeda |
| strlen(s) | Menghitung panjang karakter dari string s |

F. Langkah Kerja

1. Program 1

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include<stdio.h>
2
3  main()
4  {
5      char s[30];
6      printf("Masukkan s : ");
7      scanf("%s",&s);
8
9      printf("\n%s",s);
10 }
```

- b. Simpanlah file Anda dengan nama prak9_1.c
c. Compile dan jalankan, lalu amati hasilnya!
d. Apakah yang dilakukan program di atas?

Q1. Coba input sebuah string tanpa spasi, apa hasilnya? Kemudian inputkanlah string dengan spasi dan lihatlah apa hasilnya? Coba ganti line 7 dengan gets(s), kemudian inputlah string dengan spasi dan apa hasilnya? Jelaskan fungsi dari gets!

2. Program 2

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include<stdio.h>
2
3  main()
4  {
5      char s[30];
6      int a;
7      printf("Masukkan a : ");
8      scanf("%d",&a);
9      fflush(stdin);
10     printf("Masukkan s : ");
11     gets(s);
12
13     printf("\nNilai s : %s",s);
14     printf("\nNilai a : %d",a);
15
16 }
17
```

- b. Simpanlah file Anda dengan nama prak9_2.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

Q2. Coba hilangkan fflush(stdin) pada line 9, lalu jalankan program! Adakah perbedaan? Jelaskan fungsi stdin pada program di atas!

3. Program 3

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include<stdio.h>
2  #include<string.h>
3
4  main()
5  {
6      char s[30];
7      printf("Masukkan s : ");
8      gets(s);
9
10     printf("\n%s",s);
11
12     int i;
13     printf("String setelah dibalik : ");
14     for (i=strlen(s)-1;i>=0;i--)
15         printf("%c",s[i]);
16 }
17
```

- b. Simpanlah file Anda dengan nama prak9_3.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

4. Program 3

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
1  #include<stdio.h>
2  #include<string.h>
3
4  main()
5  {
6      char s1[30],s2[30],s3[100];
7      printf("Masukkan s1 : ");
8      gets(s1);
9
10     printf("Masukkan s2 : ");
11     gets(s2);
12
13     strcpy(s3,s1);
14     strcat(s3," ");
15     strcat(s3,s2);
16
17     printf("\ns1 : %s",s1);
18     printf("\ns2 : %s",s2);
19     printf("\ns3 : %s",s3);
20
21 }
```

- b. Simpanlah file Anda dengan nama prak9_4.c
- c. Compile dan jalankan, lalu amati hasilnya!

- d. Apakah yang dilakukan program di atas?

G. Evaluasi

1. Buatlah program untuk mengecek password yang diinputkan. Keluarkan pesan berhasil apabila password yang dimasukkan adalah kata oyi. Jika salah 3kali, maka program selesai
2. Buatlah program untuk menghitung panjang sebuah string, namun tidak termasuk spasi di dalamnya.

Contoh :

String : aku, Panjang : 3

String : aku senang, Panjang : 9

String : aku kamu dan dia, Panjang : 13

| | | |
|--|--------------------------------|--|
| POLITEKNIK NEGERI BALI | Job Sheet 10 Struct | Dosen Pengampu : Putu Indah Ciptayani, M.Cs |
| Bidang Studi : Manajemen Informatika | | Waktu : 6 x 50 Menit |
| Jurusan : Teknik Elektro | | Pertemuan 23 dan 24 |
| Kelas/Sem : IB/ Th Ajaran : 2018/2019 | | |

A. Standar Kompetensi

Mahasiswa diharapkan mampu menerjemahkan permasalahan dalam dunia nyata ke dalam algoritma dan mengimplementasikannya ke dalam Bahasa C setelah mempelajari mata kuliah ini.

B. Kompetensi Dasar

Setelah mempelajari materi ini, mahasiswa diharapkan mampu membuat program dengan melibatkan struktur dalam pemrogram C.

C. Indikator

1. Mahasiswa mampu melakukan deklarasi struktur Bahasa C dan memberinya nilai
2. Mahasiswa mampu melakukan pengaksesan sebuah struktur
3. Mahasiswa mampu membuat program yang berisi array dari struktur dalam Bahasa C

D. Peralatan

1. Seperangkat computer yang sudah terinstal compiler Bahasa C
2. Buku Ajar
3. Job sheet

E. Teori

Struktur merupakan sekelompok item yang memiliki sebuah nama. Setiap item di dalam struktur disebut sebagai anggota struktur. Anggota-anggota struktur dapat memiliki perbedaan tipe data.

Cara pendeklarasian sebuah struct adalah :

```
typedef struct nama_struct
{
    tipe_data Nama_variabel1;
    tipe_data Nama_variabel2;
    .....
    .....
    tipe_data Nama_variabeln;
} ;
```

```
typedef struct nama_struct Nama_variabel;
```

Nama_variabel pada baris terakhir merupakan sebuah variabel yang akan menyimpan data record yang dinyatakan di dalam struct di atasnya.

Cara mengakses sebuah variabel struct adalah sebagai berikut :

```
Nama_variabel.variabel
```

F. Langkah Kerja

1. Program 1

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
#include <stdio.h>

typedef struct mahasiswa
{
    char nim[10];
    char nama[30];
} ;

typedef struct mahasiswa mhs;

main()
{
    mhs mhs1;

    printf("Masukkan NIM mahasiswa : ");
    gets(mhs1.nim);
    printf("Masukkan nama mahasiswa : ");
    gets(mhs1.nama);

    printf("NIM mahasiswa : %s \n", mhs1.nim);
    printf("Nama mahasiswa : %s ", mhs1.nama);
}
```

- b. Simpanlah file Anda dengan nama prak10_1.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

2. Program 2

- a. Buatlah sebuah file dan salinlah program di bawah ini!

```
#include <stdio.h>
```

```
typedef struct mahasiswa
{
    char nim[10];
    char nama[30];
} ;

typedef struct mahasiswa mhs;

main()
{
    mhs data_mhs[4];
    int i;

    for (i=0;i<4;i++)
    {
        printf("Masukkan NIM mahasiswa ke-%d: ",i+1);
        gets(data_mhs[i].nim);
        printf("Masukkan nama mahasiswa ke-%d: ",i+1);
        gets(data_mhs[i].nama);
    }

    for (i=0;i<4;i++)
    {
        printf("NIM mahasiswa %d : %s \n",
            i+1, data_mhs[i].nim);
        printf("Nama mahasiswa %d : %s \n",
            i+1, data_mhs[i].nama);
    }
}
```

- b. Simpanlah file Anda dengan nama prak10_1.c
- c. Compile dan jalankan, lalu amati hasilnya!
- d. Apakah yang dilakukan program di atas?

G. Evaluasi

1. Buatlah sebuah program dengan array struct untuk menyimpan data 5 orang pegawai, dimana akan disimpan NIP, Nama, Gaji dan Jabatan pegawai!

