

MODUL PRAKTIKUM ALGORITMA DAN PEMROGRAMAN

REVISI KEDUA



Disusun Oleh:

Shandi Noris, M.Kom

**TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG**

**Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566
Tangerang Selatan – Banten**

TATA TERTIB PRAKTIKUM

- 1) Peserta praktikum sudah terdaftar sebagai mahasiswa Teknik Informatika Universitas Pamulang pada semester bersangkutan dengan nama sudah tertera pada lembar kehadiran dan bersedia menjalankan tata tertib yang sudah ditentukan.
- 2) Praktikum dilaksanakan sesuai dengan jadwal yang telah ditentukan.
- 3) Peserta praktikum wajib hadir tepat pada waktunya.
- 4) Peserta praktikum diharuskan menandatangani lembar kehadiran.
- 5) Semua peserta praktikum harus memiliki modul praktikum.
- 6) Selama praktikum berlangsung diwajibkan :
 - Memelihara suasana agar nyaman dan tenang
 - Tidak Membawa makanan dan minuman
 - Tidak Merokok dan mengotori ruangan
 - Tidak Bersuara dengan keras
 - Tidak Hilir mudik yang tidak perlu
 - Tidak Bertindak atau berbicara yang tidak ada hubungannya dengan kegiatan praktikum
 - Tidak diperkenankan bermain Game
 - Tidak mencoret-coret sarana yang ada
 - Tidak merubah, merusak, atau mengambil peralatan di Laboratorium
 - Meletakkan tas ditempat yang ditentukan oleh Pengajar
- 7) Pakaian yang dikenakan peserta praktikum sebagai berikut:
 - Pria
 - Celana Panjang, tidak robek
 - Sepatu Tertutup (bukan sepatu sandal)
 - Kemeja (Bukan kaos berkerah atau almamater atau sejenisnya)
 - Wanita
 - Kemeja (Bukan kaos berkerah atau almamater atau sejenisnya)
 - Sepatu tertutup (bukan Sepatu sandal)
 - Celana Panjang/Rok

Sanksi-Sanksi :

- 1) Jika Keterlambatan lebih dari 20 menit tanpa alasan yang dapat diterima, maka peserta praktikum tidak diijinkan mengikuti praktikum.
- 2) Jika peserta praktikum merusak peralatan praktikum akibat melakukan perbuatan yang tidak berkaitan dengan praktikum/prosedur praktikum maka yang bersangkutan harus mengganti kerugian akibat perbuatannya.
- 3) Jika Tata tertib tidak ditaati maka peserta praktikum dapat diberi sanksi tegas serta dapat diberi nilai E (Tidak Lulus).

PERTEMUAN I

PENGANTAR

TUJUAN PRAKTIKUM

- a) Peserta dapat melaksanakan praktikum, membuat laporan praktikum dengan benar dan mengikuti tata tertib dalam melaksanakan praktikum.

ATURAN PRAKTIKUM

a) Mengikuti Tata tertib praktikum

b) Penilaian :

- 1) Kehadiran : 30%
 - Minimal kehadiran adalah 11 kali (ujian praktikum termasuk kehadiran). Jika kurang dari 11 kali maka peserta praktikum dinyatakan tidak lulus.
 - Absen Jalan diperbolehkan jika disertai alasan yang kuat (seperti surat dokter, surat dinas, atau sejenisnya) dengan batas maksimal 3 kali. Lembar Absen Jalan diberikan kepada pengajar pada saat Ujian Praktikum.
- 2) Tugas/Laporan Praktikum : 35%

Nilai Tugas/Laporan terdiri dari: tugas/quiz, laporan awal, dan laporan akhir.
- 3) Ujian Praktikum: 35%

Ujian praktikum dilaksanakan pada pertemuan ke-14 atau pertemuan akhir.
- 4) Salah satu atau lebih dari komponen nilai di atas tidak ada, maka peserta praktikum dinyatakan tidak lulus.

c) Membuat Laporan Praktikum.

- 1) Laporan Praktikum terdiri dari Laporan Awal dan Laporan Akhir.
- 2) Laporan Awal
 - Isi dari laporan awal adalah menulis ulang setiap Modul sesuai dengan pertemuan berikutnya, dan menjawab soal Tugas Pendahuluan.
- 3) Laporan Akhir
 - Isi dari laporan akhir adalah Menulis kesimpulan pada Modul yang telah dipraktekkan dan menjawab soal Tugas Akhir.
- 4) Tata Cara Mengumpulkan Laporan
 - Laporan dikumpulkan mulai pertemuan ke-2.

- Pada pertemuan ke-2, hanya laporan awal yang dikumpulkan.
- Pada pertemuan ke-3 sampai dengan pertemuan ke-13, laporan yang dikumpulkan adalah laporan awal dan laporan akhir.
- Pada pertemuan ke-14 atau pertemuan akhir atau pada saat ujian praktikum, hanya laporan akhir yang dikumpulkan.
- Contoh:
 - Pada Pertemuan ke-2, peserta praktikum harus membawa laporan awal yang isinya adalah penulisan ulang modul pertemuan ke-2 dan menjawab soal tugas pendahuluan pertemuan ke-2.
 - Pada Pertemuan ke-3, peserta praktikum harus membawa laporan awal yang isinya adalah penulisan ulang modul pertemuan ke-3 dan menjawab soal tugas pendahuluan pertemuan ke-3, serta membawa laporan akhir pertemuan ke-2 dan menjawab soal tugas akhir pertemuan ke-2.
 - Pada pertemuan ke-4 sampai dengan pertemuan ke-13, cara mengumpulkan laporannya sama dengan pada pertemuan ke-3.
 - Pada pertemuan ke-14 atau pada saat ujian praktikum, peserta praktikum hanya membawa Laporan akhir pertemuan ke-13 dan menjawab soal tugas akhir pertemuan ke-13.
- Contoh jika peserta praktikum Tidak Hadir:
 - Jika pada pertemuan ke-4 peserta praktikum tidak hadir dikarenakan sakit dan pada pertemuan ke-5 peserta praktikum hadir, maka peserta praktikum harus membawa laporan awal yang isinya adalah penulisan ulang modul pertemuan ke-5 dan menjawab soal tugas pendahuluan pertemuan ke-5, serta membawa laporan akhir pertemuan ke-3 dan menjawab soal tugas akhir pertemuan ke-3.

5) Format laporan praktikum :

- Isi Laporan ditulis tangan pada kertas A4.
- Halaman depan Laporan Awal dan Laporan Akhir boleh diprint atau ditulis tangan, contoh seperti gambar di bawah ini.

Contoh halaman Depan (diPrint) :

LAPORAN AWAL

ALGORITMA DAN PEMROGRAMAN

LAPORAN KE-1



Disusun Oleh :

Nama : Cecep Gorbacep

NIM : 123456789

Kelas : II-A Malam

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566
Tangerang Selatan – Banten

Contoh halaman Depan (diPrint) :

LAPORAN AKHIR

ALGORITMA DAN PEMROGRAMAN

LAPORAN KE-1



Disusun Oleh :

Nama : Cecep Gorbacep

NIM : 123456789

Kelas : II-A Malam

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

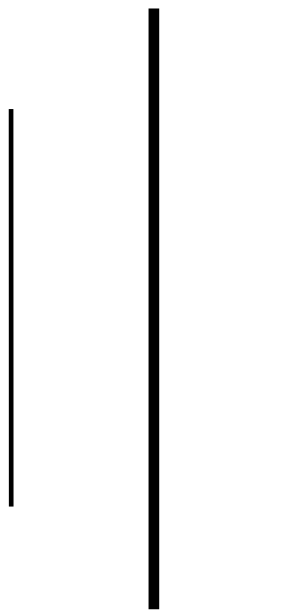
Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566
Tangerang Selatan – Banten

Contoh halaman Depan (diTulis) :

LAPORAN AWAL

ALGORITMA DAN PEMROGRAMAN

LAPORAN KE-1



Disusun Oleh :

Nama : Cecep Gorbacep

NIM : 123456789

Kelas : II-A Malam

TEKNIK INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS PAMULANG

Jl. Surya Kencana No. 1 Pamulang Telp (021)7412566, Fax. (021)7412566
Tangerang Selatan – Banten

PERTEMUAN II

DASAR PEMROGRAMAN C/C++

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar penggunaan bahasa pemrograman C/C++ dan editornya.
- b) Mahasiswa dapat menjelaskan dan menggunakan tipe data dan operator pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Bahasa C++ merupakan bahasa pemrograman tingkat tinggi, yakni yang memakai kata-kata dan pernyataan yang mudah dimengerti manusia, meskipun masih jauh berbeda dengan bahasa manusia sesungguhnya. Program C++ merupakan program yang berbentuk fungsi-fungsi. Main() merupakan nama dari suatu fungsi yang harus ada di program C++ dan diletakkan di bagian tertentu yang menunjukkan sebagai *compiler* dimana awal dari suatu program. Selain itu main() hanya dapat digunakan sekali saja dalam satu program.

b) Tipe Data

1) Tipe Integer

Tipe integer atau bilangan bulat dapat dideklarasikan dengan menggunakan kata kunci (*keyword*) atau kata cadangan (*reserved word*), yaitu *short* atau *long*. Perhatikan tabel berikut ini.

Tabel 1. *Tipe integer*

Keyword	Besarnya Memori	Jangkauan Nilai
Int	16 bit	-32768 s/d 32768
Short int atau short	16 bit	-32768 s/d 32768
Long int atau long	32 bit	-2147483648 s/d 2147483648

Disamping tipe integer ini, dapat juga dideklarasikan tipe integer yang tidak bertanda (*unsigned integer*) dengan menggunakan kata kunci *unsigned*. Perhatikan tabel berikut ini.

Tabel 2. Tipe integer dengan kata kunci *unsigned*

Keyword	Besarnya Memori	Jangkauan Nilai
Unsigned Int	16 bit	0-65535
Unsigned short	16 bit	0-65535
Unsigned long	16 bit	0-4294967295

2) Tipe Pecahan

Tipe Pecahan atau *floating point* dapat dideklarasikan dengan menggunakan kata kunci (*keyword*) atau kata cadangan (*reserved*), yaitu *float*, *double* dan *long double*. Perhatikan tabel berikut ini.

Tabel 3. Tipe pecahan

Keyword	Besarnya Memori	Jangkauan Nilai
Float	32 bit	3.4E-038 s/d 3.4E+038
Double	64 bit	31.7E-038 s/d 1.7E+038
Long double	64 bit	31.7E-038 s/d 1.7E+038

c) Operator

Operator merupakan symbol atau kata yang dapat digunakan dalam program untuk melakukan suatu operasi atau manipulasi, seperti menjumlahkan dua buah nilai, memberikan nilai ke suatu variabel, membandingkan dua buah nilai, dan lain sebagainya. Tanda operator dalam bahasa C/C++ dapat dikelompokkan sebagai berikut :

1) Operator pengerjaan (*assignment operators*)

Operator pengerjaan berupa tanda sama dengan (=) yang digunakan untuk mengisi nilai yang berada sebelah kanannya variabel yang ditunjukkan di sebelah kirinya.

Contoh :

$$a = 5$$

$$a = b = c$$

maka nilai a dan b sama dengan c yaitu 5.

2) Operator aritmatika (*arithmetic operators*)

Operator aritmatika yang tergolong sebagai operator *binary* terlihat pada tabel berikut.

Tabel 4. *Tabel operator binary*

Operator	Keterangan
*	Perkalian
/	Pembagian
%	Modulus atau sisa pembagi
+	Penambahan
-	Pengurangan

Operator aritmatika yang tergolong sebagai operator *unary* terlihat pada tabel berikut.

Tabel 5 *Tabel operator unary*

Operator	Keterangan
+	Tanda plus
-	Tanda minus
++	Prefix : preincrement; Postfix: postincrement
--	Prefix : predecrement; Postfix: postdecrement

3) Operator peningkatan dan penurunan (*increment and decrement operators*)

C/C++ mempunyai operator yang terkenal, yaitu operator peningkatan (*increment operators*) yang menggunakan tanda operasi (++) dan operator penurunan yang menggunakan tanda operasi (--). Operator penambahan digunakan untuk menambahkan nilai 1 pada nilai sebelumnya dan operator penurunan digunakan untuk mengurangi nilai 1 pada nilai sebelumnya.

4) Operator *bitwise* (*bitwise operators*)

Untuk operator *bitwise* dalam bentuk bit, bahasa C++ menyediakan beberapa operator seperti yang diperlihatkan pada tabel di bawah ini.

Tabel 6 *Tabel operator bitwise*

Operator	Keterangan
<<	Shift left (pergeseran bit ke kiri)
>>	Shift right (pergeseran bit ke kanan)
&	Operasi bit AND
	Operasi bit OR
^	Operasi bit XOR
~	Operasi bit NOT

5) Operator hubungan (*relations operators*)

Operator hubungan (*relations operators*) digunakan untuk membandingkan dua elemen nilai dan akan dihasilkan nilai perbandingannya, yaitu betul (bernilai 1) atau salah (bernilai 0). Operator ini banyak digunakan pada penyelesaian suatu kondisi di statmen *IF*.

Tabel 7 *Tabel operator hubungan*

Operator	Keterangan
>	Lebih besar dari
>=	Lebih besar sama dengan
<	Lebih kecil
<=	Lebih kecil sama dengan
==	Sama dengan
!=	Tidak sama dengan

6) Operator logika (*logical operators*)

Operator logika biasanya berpasangan dengan operator hubungan. Operator logika dapat dilihat pada tabel berikut ini.

Tabel 8 *Tabel operator logika*

Operator	Keterangan
&&	Operator logika AND
	Operator logika OR
!	Operator logika NOT

7) Operator alamat (*address operators*)

C++ juga menyediakan dua buah operator alamat (*address operators*) yang berhubungan dengan penggunaan pointer. Perhatikan tabel berikut ini.

Tabel 9 *Tabel operator alamat*

Operator	Keterangan
&	Address of Operator
*	Indirection Operator

8) Operator koma (*comma operators*)

Operator koma (*comma operators*) digunakan untuk meletakkan beberapa ungkapan yang dipisahkan dengan tanda koma di dalam kurung buka dan kurung tutup.

9) Operator gabungan

C++ memungkinkan penulisan suatu operator pengerjaan (=) untuk digabung dengan operator lainnya yaitu dengan operator aritmetika, operator peningkatan dan penurunan serta operator *bitwise*.

Bentuk penulisannya sebagai berikut :

```
operator pengerjaan
variabel = variabel operator ungkapan
```

Dapat diubah dengan ditulis sebagai berikut :

```
operator lainnya
operator pengerjaan
variabel operator = ungkapan
```

TUGAS PRAKTIKUM

- a) Buatlah program menghitung luas dan keliling persegi panjang (simpan dengan nama lat2_1.cpp)**

```
#include <stdio.h>
main()
{
    int panjang, lebar, luas, keliling;
    printf ("Masukkan panjang :"); scanf ("%d", &panjang);
    printf ("Masukkan lebar :"); scanf ("%d", &lebar);
    luas=panjang*lebar;
    keliling=2*(panjang+lebar);
    printf ("Jadi luasnya adalah : %d\n",luas);
    printf ("Jadi kelilingnya adalah : %d\n",keliling);
}
```

- b) **Buatlah program menghitung luas dan volume Balok (simpan dengan nama lat2_2.cpp)**

```
#include <stdio.h>
main()
{
    int panjang, lebar, tinggi, luas, volume;
    printf ("Masukkan panjang :"); scanf ("%d",&panjang);
    printf ("Masukkan lebar :"); scanf ("%d",&lebar);
    printf ("Masukkan tinggi :"); scanf ("%d",&tinggi);
    luas=(2*panjang*lebar)+(2*panjang*tinggi)+(2*tinggi*lebar);
    volume = panjang*lebar*tinggi;
    printf ("Jadi luasnya adalah : %d\n",luas);
    printf ("Jadi volumenya adalah : %d\n",volume);
}
```

- c) **Buatlah program contoh penggunaan operator pengerjaan (simpan dengan nama lat2_3.cpp)**

```
#include <stdio.h>
main()
{
    int a,b,c;
    a=25;
    b=15;
    c=a+b;
    printf ("%d",c);
}
```

- d) **Buatlah program contoh penggunaan operator aritmatika (simpan dengan nama lat2_4.cpp)**

```
#include <stdio.h>
main()
{
    floatluas, pi, jari;
    pi = 3.1428;
    jari = 20.0;
    luas = pi*jari*jari;
    printf ("luas = %f ", luas);
}
```

- e) **Buatlah program contoh penggunaan operator hubungan (simpan dengan nama lat2_5.cpp)**

```
#include <stdio.h>
main()
{
    long int beli, disc, total;
    printf ("Masukkan Pebelian :");scanf ("%ld",&beli);
    if (beli>=100000)
    {
        disc=0.1*beli;
        total=beli-disc;
    }
    else
    {
        disc=0.05*beli;
        total=beli-disc;
    }
    printf ("Jadi Pembelian adalah : %ld\n",beli);
    printf ("Jadi Discount adalah : %ld\n",disc);
    printf ("Jadi Total Pembayaran adalah : %ld\n",total);
}
```

- f) **Buatlah program contoh penggunaan operator logika (simpan dengan nama lat2_6.cpp)**

```
#include <stdio.h>
main()
{
    int nilai, tugas;
    printf ("Masukkan sebuah nilai :");scanf ("%d",&nilai);
    printf ("Masukkan nilai tugas :");scanf ("%d",&tugas);
    if (nilai>70 && tugas>=75)
        printf ("Lulus\n");
    else
        printf ("Tidak Lulus\n");
}
```

- g) **Buatlah program contoh penggunaan operator koma (simpan dengan nama lat2_7.cpp)**

```
#include <stdio.h>
main()
{
    char kar1, kar2;
    (kar1='A', kar2='B');
    printf ("%c\n",kar1);
    printf ("%c\n",kar1);
}
```

TUGAS PENDAHULUAN

1. Tuliskan dan jelaskan kata-kata kunci pada bahasa C!
2. Tuliskan dan jelaskan fungsi-fungsi yang digunakan pada bahasa C!
3. Tuliskan dan jelaskan konstanta karakter escape (escape character) pada bahasa C!
4. Tuliskan dan jelaskan kode-kode format untuk fungsi printf()!

TUGAS AKHIR

1. Buatlah algoritma dan program untuk menghitung volume dan luas permukaan bola apabila masukkannya berupa jari-jari. Jari-jari yang dimasukkan bertipe float dan output atau keluaran yang dihasilkan adalah luas dan volume permukaan bola dengan tipenya adalah float.

Petunjuk:

$$\text{Volume} = \frac{4}{3} \times \pi \times r^3$$

$$\text{Luas} = 4 \times \pi \times r^2$$

Dengan:

$$\pi = 3,14$$

r = jari-jari bola

PERTEMUAN III

DECISION (IF)

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar penggunaan perintah IF pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan perintah IF pada bahasa pemrograman C/ C++.
- c) Mahasiswa dapat mengimplementasikan algoritma dalam pemecahan suatu masalah dengan berbagai alternatif jawaban yang tersedia dengan mengambil keputusan untuk memilih satu jawaban yang tepat.

TEORI DASAR

a) Pendahuluan

Perintah IF berguna untuk memilih dua atau lebih alternatif jawaban yang tersedia. Jika perintah IF tersebut terdiri dari dua atau lebih pernyataan, maka pernyataan tersebut harus berada di antara *behind* dan *end*.

Perintah IF memiliki bentuk umum :

```
if (kondisi)
    pernyataan
```

Bentuk di atas mempunyai arti jika kondisi benar, maka pernyataan di bawahnya akan dikerjakan. Jika perintah IF merupakan perintah yang menentukan pengambilan keputusan bahwa pernyataan itu akan dikerjakan atau tidak.

Pada perintah IF juga dapat pula diikuti oleh pernyataan majemuk.

Bentuknya adalah :

```
if (kondisi)
{
    Pernyataan1
    Pernyataan2
    .....
    .....
    .....
    Pernyataann
}
```


1) Perintah IF ELSE

Perintah IF..ELSE mempunyai bentuk umum sebagai berikut :

```
if (kondisi)
    Pernyataan1
else
    Pernyataan2
```

2) Perintah IF dalam IF

Perintah IF dalam IF sering disebut *nested-if*. Perintah ini mempunyai bentuk umum sebagai berikut:

```
if (kondisi1)
    if (kondisi2)
        Pernyataan1;
    else
        Pernyataan2;
else
    Pernyataan3;
```

Pernyataan IF dalam IF juga mempunyai bentuk yang majemuk yang sering disebut IF bertingkat. Bentuk umumnya adalah sebagai berikut:

```
if (kondisi1)
    Pernyataan1;
else if (kondisi2)
    Pernyataan2;
else if (kondisi3)
    Pernyataan3;
.....
.....
else
    pernyataann;
```

TUGAS PRAKTIKUM

- a) **Buatlah program contoh dasar Penggunaan perintah IF (simpan dengan nama lat3_1.cpp)**

```
#include <stdio.h>
main()
{
    int nilai;
    printf ("Masukkan sebuah nilai :");scanf ("%d",&nilai);
    if (nilai>70)
        printf ("Lulus\n");
}
```

- b) **Buatlah program contoh Penggunaan perintah IF-Else (simpan dengan nama lat3_2.cpp)**

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

main()
{
    long int kode, jumlah, sks, semester;
    char nama[100];
    printf ("Kode Masukkan: [1-2]:"); scanf ("%ld", &kode);
    printf ("Jumlah SKS per semester [20-25] : "); scanf ("%d", &sks);

    if (kode == 1)
    {
        strcpy(nama, "Teknik Informatika");
        jumlah = 30000*sks;
        semester = jumlah*2;
    }
    else if (kode == 2)
    {
        strcpy(nama, "Akuntansi");
        jumlah = 20000*sks;
        semester = jumlah*2;
    }
    else if (kode == 1)
    {
        strcpy(nama, "Manajemen");
        jumlah = 15000*sks;
        semester = jumlah*2;
    }
    printf("*=====*\n");
    printf("Nama Kursus : %s\n", nama);
    printf("Pembayaran dalam 1 semester: %ld\n", jumlah);
    printf("Pembayaran dalam 1 tahun: %ld\n", semester);
    getch();
}
```

- c) **Buatlah program “Berdasarkan Algoritma Berikut ini” (simpan dengan nama lat3_3.cpp)**

1. Masukkan nilai x
2. Jika $x > 0$, maka x adalah bilangan positif
3. Jika $x < 0$, maka x adalah bilangan negative
4. Jika $x = 0$, maka x adalah bilangan 0

- d) **Buatlah program “Berdasarkan Algoritma Berikut ini” (simpan dengan nama lat3_4.cpp)**

1. Masukkan nilai
2. Jika nilai ≥ 81 dan nilai ≤ 100 maka huruf=A
3. Jika nilai ≥ 71 dan nilai < 81 maka huruf=B
4. Jika nilai ≥ 61 dan nilai < 71 maka huruf=C
5. Jika nilai ≥ 51 dan nilai < 61 maka huruf=D
6. Jika nilai < 51 maka huruf=E
7. Tampilkan nilai angka dan huruf

TUGAS PENDAHULUAN

1. Jelaskan perbedaan perintah IF dan Switch!
2. Sebutkan dan jelaskan jenis-jenis perintah IF!
3. Mengapa anda sebagai programmer harus menggunakan perintah IF dalam program!
4. Buatlah contoh algoritma dan program sederhana menggunakan perintah IF dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program berdasarkan Algoritma Berikut ini!
 - 1) Masukkan kode
 - 2) Jika kode=1 maka hari=Senin
 - 3) Jika kode=2 maka hari=Selasa
 - 4) Jika kode=3 maka hari=Rabu
 - 5) Jika kode=4 maka hari=Kamis
 - 6) Jika kode=5 maka hari=Jumat
 - 7) Jika kode=6 maka hari=Sabtu
 - 8) Jika kode=7 maka hari=Minggu
 - 9) Jika kode yang dimasukkan salah maka pilihannya tidak ada
 - 10) Tampilkan kode dan hari

PERTEMUAN IV

DECISION (SWITCH)

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar penggunaan perintah Switch pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan perintah Switch pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Perintah SWITCH merupakan pernyataan yang di rancang untuk menangani pengambilan keputusan yang melibatkan sejumlah pilihan alternatif yang di antaranya untuk menggantikan pernyataan IF bertingkat.

Bentuk umumnya adalah sebagai berikut:

```
SWITCH(ekspresi)
{
    case konstanta1:
        pernyataan1;
        break;
    case konstanta2:
        pernyataan2;
        break;
    case konstanta3:
        pernyataan3;
        break;
    case konstanta4:
        pernyataan4;
        break;
    .....
    .....
    .....
    case konstanta:
        pernyataan;
        break;
    default:
        pernyataan n;
}
```

Dari bentuk umum diatas,dapat dijelaskan bahwa ekspresi bisa berupa ungkapan yang bernilai integer atau bertipe karakter. Setiap konstanta1, konstanta2,

konstanta3, konstanta4, sampai dengan konstanta n dapat berupa konstanta integer atau konstanta karakter

Setiap pernyataan1, pernyataan2, pernyataan3, pernyataan4 sampai pernyataann dapat sebuah atau beberapa pernyataan. Pengujian pada switch akan dimulai dari konstanta1. Apabila nilainya cocok dengan ekspresi maka pernyataan1 akan dijalankan. Kalau tidak cocok maka akan beralih ke pernyataan2 dan seterusnya sampai hasil yang diinginkan.

Perintah break menyebabkan eksekusi diarahkan ke akhir switch kalau nilai konstanta1 tidak sama dengan nilai ekspresi dan kemudian baru diteruskan pengujian dengan konstanta2 dan seterusnya. Jika sampai pada pengujian akhir tidak ada yang cocok maka default akan dijalankan.

TUGAS PRAKTIKUM

a) **Buatlah program contoh program switch (simpan dengan nama lat4_1.cpp)**

```
#include <stdio.h>
Main()
{
    int kdhari;
    printf ("Masukan kode hari[1-7]:"); scanf("%d",&kdhari);
    switch (kdhari)
    {
        case 1:
            printf("Senin"); break;
        case 2:
            printf("Selasa"); break;
        case 3:
            printf("Rabu"); break;
        case 4:
            printf("Kamis"); break;
        case 5:
            printf("Jum'at"); break;
        case 6:
            printf("Sabtu"); break;
        case 7:
            printf("Minggu"); break;
        default:
            printf("Kode Tidak ada");
    }
}
```

- b) **Buatlah program “Berdasarkan Algoritma Berikut ini” (simpan dengan nama lat4_2.cpp)**

- 1) Membuat program Menu untuk menghitung Luas dan keliling Segitiga, Persegi Panjang, Lingkaran.
- 2) Masukkan pilihan berupa kode, yaitu S,P, dan L
- 3) Jika memilih S maka masuk menu Segitiga
- 4) Jika memilih P maka masuk menu Persegi Panjang
- 5) Jika memilih L maka masuk menu Lingkaran

TUGAS PENDAHULUAN

1. Jelaskan kelebihan perintah Switch dibandingkan perintah IF!
2. Mengapa anda sebagai programmer harus menggunakan perintah Switch dalam program!
3. Sebutkan dan jelaskan perintah-perintah pendukung perintah Switch!
4. Buatlah contoh algoritma dan program sederhana menggunakan perintah Switch dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah algoritma dan program untuk menentukan bonus yang akan diperoleh seorang pelanggan jika diberlakukan ketentuan berikut ini:

Pembelian	Bonus
100000 <= Pembelian <200000	Discout 5%
200000 <= Pembelian <300000	Tiket ke Yogya
300000 <= Pembelian <400000	Tiket ke Bali
400000 <= Pembelian <500000	Jam Tangan Rolex
Pembelian >= 500000	Tiket ke Swiss

PERTEMUAN V

LOOPING (FOR)

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar penggunaan perintah-perintah perulangan (looping) pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan perintah FOR pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Perintah pengulangan (looping) digunakan untuk melakukan suatu proses berulang-ulang, jika suatu kondisi dipenuhi atau sebaliknya. Pada prakteknya, proses pengulangan biasanya digunakan untuk beberapa hal, misalkan mengulang proses pemasukan data, mengulang proses perhitungan dan mengulang proses penampilan hasil pengolahan data.

b) For

Proses *looping* yang menggunakan perhitungan (counter) dapat dibuat dengan pernyataan FOR. Pernyataan ini digunakan bila anda sudah tahu berapakah anda akan mengulang satu atau beberapa pernyataan.

Perintah ini merupakan salah satu perintah pengulangan yang sering kali digunakan dan memiliki bentuk umum sebagai berikut:

```
for (ungkapan1; keadaan; ungkapan3)
{
    Pernyataan;
}
```

Bentuk diatas memiliki arti

1. Ungkapan1 adalah inisialisasi terhadap variable pengendali *looping*
2. Keadaan adalah kondisi untuk keluar dari *looping*
3. Ungkapan3 adalah pengaturan kenaikan atau penurunan nilai variable pengendali *looping*

TUGAS PRAKTIKUM

- a) **Buatlah program contoh penggunaan perintah for (simpan dengan nama lat5_1.cpp)**

```
#include<stdio.h>
main()
{
    int n ;
    for (n=1; n<=10; n++)
    {
        printf("%d\n", n);
    }
}
```

- b) **Buatlah program contoh penggunaan perintah for dengan continue & break (simpan dengan nama lat5_2.cpp)**

```
#include<stdio.h>
main()
{
    int i;
    for( i=0; i<10; i++)
    {
        if(i==4) continue;
        printf("bilangan :%d\n",i)
        if(i==6) break;
    }
}
```

- c) **Buatlah program contoh penggunaan perintah for dengan goto (simpan dengan nama lat5_3.cpp)**

```
#include<stdio.h>
main()
{
    int n=1;
    cetak:
    printf("c++ sangat mudah\n");
    if(n++<10)
        goto cetak;
}
```


- d) **Buatlah program menampilkan bilangan ganjil dan genap, Tampilan Output sebagai berikut (simpan dengan nama lat5_4.cpp)**

```
Ganjil : 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 .....  
Genap : 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 .....
```

TUGAS PENDAHULUAN

1. Jelaskan kelebihan dan kekurangan perintah-perintah perulangan (looping) pada bahasa C/C++!
2. Mengapa anda sebagai programmer harus menggunakan perintah for dalam program!
3. Jelaskan perbedaan operator aritmatika i++ dan ++i!
4. Buatlah contoh algoritma dan program sederhana menggunakan perintah For dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program menampilkan bilangan Fibonacci menggunakan perintah for, Tampilan Output sebagai berikut ini.

```
Fibonacci : 0 1 1 2 3 5 8 13 21 34 55 89 .....
```

PERTEMUAN VI LOOPING (WHILE)

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar penggunaan perintah-perintah perulangan (looping) pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan perintah WHILE pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Do..while

Perintah ini menyatakan pengulangan proses selama kondisi tertentu.

Bentuk umumnya adalah sebagai berikut:

```
do
Pernyataan;
While(keadaan);
    Atau
do
{
Pernyataan;
Pernyataan;
}
While(keadaan)
```

b) While

Perintah WHILE ini prinsipnya sama dengan perintah DO...WHILE hanya pengujian kondisinya terletak pada *loop*.

Bentuk umumnya adalah sebagai berikut:

```
while (keadaan)
pernyataan;
atau
while(keadaan)
{
    Pernyataan;
    Pernyataan;
}
```

Pernyataan dapat berupa pernyataan tunggal atau beberapa pernyataan yang dibatasi dengan tanda{ }. Pernyataan itu akan di jalankan bila kondisinya benar.

TUGAS PRAKTIKUM

- a) Buatlah program contoh penggunaan perintah do..while (simpan dengan nama lat6_1.cpp)

```
#include<stdio.h>
main()
{
    int counter;
    counter = 0;
    do
    {
        printf("c++\n");
        counter++;
    }
    while(counter<15);
}
```

- b) Buatlah program contoh penggunaan perintah while (simpan dengan nama lat6_2.cpp)

```
#include<stdio.h>
main()
{
    int i;
    i=10;
    while (i>0)
    {
        printf ("%d\n",i);
        i--;
    }
}
```

- c) **Buatlah program contoh penggunaan perintah do while (simpan dengan nama lat6_3.cpp)**

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

main()
{
    int nomer, bagi, sisa=0;
    char chrpilihan;
    do
    {
        printf("masukan suatu bilangan :"); scanf("%d", &nomer);
        printf("masukan bilangan pembagi (2/3/4/5): ");scanf("%d", &bagi);
        printf("bilangan yang di pilih : %d\n", nomer);
        printf("bilangan pembagi      : %d\n", bagi);
        sisa = nomer % bagi;
        if (sisa == 0)
        {
            printf("sisa bagi : tidak ada\n");
        }
        else
        {
            printf("sisa bagi: %d\n", sisa);
        }
        printf( "\napakah anda ingin meneruskan ? (y/n) :");
        scanf (" %c", &chrpilihan);
    }
    while(toupper(chrpilihan) == 'y');
}
```

- d) **Buatlah program menampilkan bilangan ganjil dan genap, Tampilan Output sebagai berikut (simpan dengan nama lat6_4.cpp)**

```
Ganjil : 1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 .....
Genap  : 2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 .....
```

TUGAS PENDAHULUAN

1. Mengapa anda sebagai programmer harus menggunakan perintah while atau do..while dalam program!

2. Jelaskan kelebihan dan kekurangan perintah while dan do..while!
3. Buatlah contoh algoritma dan program sederhana menggunakan perintah do..while dengan memakai Flowchart!
4. Buatlah contoh algoritma dan program sederhana menggunakan perintah while dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program menampilkan bilangan Fibonacci menggunakan perintah while, Tampilan Output sebagai berikut ini.

Fibonacci : 0 1 1 2 3 5 8 13 21 34 55 89
--

PERTEMUAN VII

FUNGSI

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar penggunaan Fungsi pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat membuat dan menggunakan Fungsi beserta strukturnya pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Dasar fungsi

Pada umumnya fungsi memerlukan masukan yang dinamakan argument atau parameter. Hasil akhir fungsi akan berupa sebuah nilai (nilai fungsi balik)

Adapun bentuk umumnya definisi sebuah fungsi adalah:

```
Penentu_tipe nama_fungsi (daftar parameter)
Deklarasi parameter
{
Tubuh fungsi
}
```

Penentu tipe berfungsi untuk menentukan tipe keluaran fungsi yang dapat berupa salah satu tipe data C++ yang berlaku, misalnya char int, default tipe fungsi yang tidak disebutkan dianggap sebagai int.

Sebuah fungsi dapat saja tidak mengandung parameter. Tentu saja untuk kondisi ini parameter juga tidak ada

Contoh:

```
Inisialisasi()
{
Return (0)
}
```

TUGAS PRAKTIKUM

- a) **Buatlah program contoh program parameter rekursi dan factorial (simpan dengan nama lat7_1.cpp)**

```

#include<stdio.h>
int factorial(int n)
{
    if (n==1)
    { return(1); }
    else
    { return (n*factorial(n-1)); }
}

main()
{
    int x;
    printf("mencari nilai faktorial\n");
    printf("masukkan nilai x:");scanf("%d",&x);
    printf("nilai factorial dari %d=%d\n",x,factorial(x));
}

```

- b) Buatlah program contoh program implimintasi fungsi (simpan dengan nama lat7_2.cpp)**

```

int findmax(int n1, int n2);
void printmax(int m);
main()
{
    int i =5;
    int j =7;
    int k;
    k =findmax(i,j);
    printmax(k);
}

int findmax(int n1, int n2)
{
    if(n1 > n2)
    { return n1; }
    else
    { return n2; }
}

void printmax(int m)
{
    printf ( " bilangan yang terbesar adalah :%d\n",m);
}

```

- c) **Buatlah program contoh program parameter formal dan actual (simpan dengan nama lat7_3.cpp)**

```
#include<stdio.h>
void tukar(int x, int y);
main()
{
    int a, b;
    a=99;
    b=11;
    printf("nilai sebelum pemanggilan fungsi :\n");
    printf("nilai a=%d nilai b=%d\n\n",a,b);
    tukar(a,b);
    printf("nilai sesudah pemanggilan fungsi :\n");
    printf("nilai a=%d nilai b=%d\n\n",a,b);
}

void tukar (int px, int py)
{
    int z;
    z=px;
    px=py;
    py=z;
    printf("nilai diakhir fungsi :\n");
    printf("nilai px=%d nilai py=%d\n\n",px,py);
}
```

- d) **Buatlah program contoh program fungsi rekursi untuk fibonacci (simpan dengan nama lat7_4.cpp)**

```
#include<stdio.h>
long fibonacci (long n)
{
    if (n==1 || n==2)
    { return 1; }
    else
    { return fibonacci (n-1) + fibonacci(n-2); }
}
main()
{
    int x;
    printf("masukan nilai fibonacci\n");
    printf("masukan nilai x:"); scanf("%d",&x);
    printf("nilai fibonacci dari %d=%d\n", x, fibonacci(x));
}
```


- e) **Buatlah program contoh program kali (simpan dengan nama lat7_5.cpp)**

```
#include<iostream.h>
int mult(int x, int y, int z);

int main()
{
    int x, y, z;
    cout << "masukan bilangan pertama :";
    cin>>x;
    cout << "masukan bilangan kedua :";
    cin>>y;
    cout << "masukan bilangan ketiga :";
    cin>>z;
    cout<< "hasil perkalian bilangan itu adalah " << mult(x, y, z);
    return 0;
}

int mult(int x, int y, int z)
{
    return x*y*z;
}
```

TUGAS PENDAHULUAN

1. Jelaskan yang dimaksud dengan rekursi pada Fungsi pada bahasa pemrograman C/C++!
2. Sebutkan dan jelaskan perintah-perintah pendukung Fungsi!
3. Jelaskan kelebihan menggunakan Fungsi pada bahasa pemrograman C/C++!
4. Buatlah contoh algoritma dan program sederhana menggunakan Fungsi dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program menampilkan bilangan Genap dan Ganjil menggunakan Fungsi!

PERTEMUAN VIII

ARRAY

TUJUAN PRAKTIKUM

- Mahasiswa dapat menjelaskan konsep dasar Array pada bahasa pemrograman C/C++.
- Mahasiswa dapat menjelaskan dan menggunakan jenis-jenis Array pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Array merupakan koleksi data dimana setiap elemen memakai nama dan tipe yang sama serta setiap elemen diakses dengan membedakan indeks *array*-nya.

Contoh :

Sebuah variabel bernama **c** yang mempunyai 7 lokasi memori yang semuanya bertipe **int**. Maka pendeklarasiannya adalah :

```
int c[7] ;
```

Nilai suatu variabel *array* dapat diinisialisasikan secara langsung pada saat deklarasi, misalnya :

```
int c[7] = { -45, 0, 6, 72, 1543, 43, 4 } ;
```

b) Array Dimensi Satu

Bentuknya :

```
Tipe nama_var[ukuran];
```

c) Array Dimensi Dua

Bentuknya :

```
Tipe nama_var[ukuran][ukuran];
```

Dan definisi variabel untuk setiap elemen tersebut adalah:

	0	1	2
0	b[0][0]	b[0][1]	b[0][2]
1	b[1][0]	b[1][1]	b[1][2]

d) Array Dimensi Banyak

Bentuknya :

```
Tipe nama_var[ukuran][ukuran][ukuran][ukuran] dst.;
```

Array ini seperti *array* dimensi dua tetapi dapat memiliki ukuran yang lebih besar. Sebenarnya *array* dimensi ini tidak terlalu sering digunakan, tetapi sewaktu-waktu apabila dimensi yang dibutuhkan banyak, maka *array* ini sangat memegang peranan yang penting.

Contoh :

```
Static int dataktip [2] [7] [8] [5];
```

TUGAS PRAKTIKUM

- a) **Buatlah program contoh penggunaan Array (simpan dengan nama lat8_1.cpp)**

```
#include <stdio.h>
#define SIZE 12

main()
{
    int a [ SIZE ]= {1, 3, 5, 4, 7, 2, 99, 16, 45, 67, 89, 45};
    int indeks, total = 0;

    for (indeks=0 ; indeks<=SIZE-1 ; indeks++)
    {
        total += a [indeks];
    }
    printf ("\nTotal Setiap Elemen Array adalah %d" , total);
    return 0;
}
```

- b) **Buatlah program contoh penggunaan Array (simpan dengan nama lat8_2.cpp)**

```
#include <stdio.h>
main()
{
    int square [100];
    int i, k;
    for (i=0 ; i<10 ; i++)
    {
        k = i+1;
        square[i] = k*k;
        printf ("\nPangkat dari %d adalah %d", k, square [i]);
    }
    return 0;
```

- c) **Buatlah program contoh penggunaan Array (simpan dengan nama lat8_3.cpp)**

```
#include <stdio.h>
void printArray (int [ ][3]);
main()
{
    int matrik1 [2][3] = {{1,2,3}, {4,5,6}},
        matrik2 [2][3] = {1,2,3,4,5},
        matrik3 [2][3] = { {1,2}, {4} };
    printArray (matrik1);
    printArray (matrik2);
    printArray (matrik3);
    return 0;
}

void printArray ( int a [ ][3])
{
    int i, j;
    for (i=0 ; i<=1 ; i++)
    {
        for (j=0 ; j<=2 ; j++)
            printf ("%d", a [i] [ j ] );
        printf ("\n");
    }
}
```

- d) **Buatlah program menampilkan matriks 3x3, Tampilan Output sebagai berikut (simpan dengan nama lat8_4.cpp)**

```
Input baris1kolom1 = 1
Input baris1kolom2 = 3
Input baris1kolom3 = 4
Input baris2kolom1 = 6
Input baris2kolom2 = 7
Input baris2kolom3 = 8
Input baris3kolom1 = 0
Input baris3kolom2 = 2
Input baris3kolom3 = 2
```

```
Matrik 3x3 = 1  3  4
              6  7  8
              0  2  2
```

TUGAS PENDAHULUAN

1. Jelaskan yang dimaksud dengan Array!
2. Sebutkan dan jelaskan jenis-jenis Array!
3. Jelaskan kelebihan menggunakan Array pada bahasa pemrograman C/C++!
4. Buatlah contoh algoritma dan program sederhana menggunakan Array dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program menampilkan bilangan matriks 3x3 menggunakan Array!

PERTEMUAN IX

STRING

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar String pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan jenis-jenis String pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

String merupakan bentuk data yang biasa dipakai dalam pemrograman guna menampung dan memanipulasi data teks. Didalam Turbo C++, *string* bukan merupakan tipe data tersendiri tetapi hanya merupakan jenis khusus dari *array*.

String juga bisa dipergunakan dalam konstanta dan variabel.

b) Fungsi Pada String

Pada Turbo C++ disediakan sejumlah fungsi makro yang didefinisikan dengan praprosesor `#define`.

Makro-makro yang didefinisikan dalam *type.h* adalah :

1) **Isalnum ()**

Fungsi makro ini akan menghasilkan benar (bukan nol) kalau argumennya adalah huruf besar, huruf kecil dan karakter digit '0' sampai '9'.

Bentuknya adalah sebagai berikut:

```
Tipe isalnum ( tipe argument ) ;
```

Contoh:

```
Tipe isalnum ( int d ) ;
```

2) **Isalpha ()**

Fungsi makro ini akan menghasilkan benar (bukan nol) kalau argumennya adalah huruf bukan angka.

Bentuknya adalah sebagai berikut:

```
Tipe isalpha ( tipe argument ) ;
```

3) Isdigit ()

Fungsi makro ini akan menghasilkan benar (bukan nol) kalau argumennya adalah karakter digit '0' sampai '9'.

Bentuknya adalah sebagai berikut:

```
Tipe isdigit ( tipe argument ) ;
```

4) Islower ()

Fungsimakro ini akan menghasilkan benar (bukan nol) kalau argumennya adalah huruf kecil.

Bentuknya adalah sebagai berikut:

```
Tipe islower ( tipe argument ) ;
```

5) Isupper ()

Fungsimakro ini akan menghasilkan benar (bukan nol) kalau argumennya adalah huruf besar.

Bentuknya adalah sebagai berikut:

```
Tipe isuper ( tipe argument ) ;
```

6) Tolower ()

Fungsi makro ini akan menghasilkan huruf kecil jika diisi huruf besar, jika diisi bukan huruf besar maka akan menghasilkan sama dengan argmen itu sendiri.

Bentuknya adalah sebagai berikut:

```
Tipe tolowe ( tipe argument ) ;
```

7) Toupper ()

Fungsi makro ini kebalikan dari *tolower* yaitu akan menghasilkan huruf besar jika diisi huruf kecil, jika diisi bukan huruf besar maka akan menghasilkan sama dengan argmen itu sendiri.

Bentuknya adalah sebagai berikut:

```
Tipe toupper ( tipe argument ) ;
```

8) Strcpy ()

Fungsi ini dipakai untuk menyalin string asal ke variabel string tujuan, variabel tujuan harus memiliki kuran yang dapat menampung seluruh karakter string asal. Bentuknya adalah sebagai berikut:

```
Strcpy ( tujuan, asal) ;
```

9) Strlen ()

Fungsi ini dipakai untuk menghitung jumlah karakter argumennya.

Bentuknya adalah sebagai berikut:

```
char S [ ] ="Hallo"  
printf ("Panjangnya :%d\n, strlen (S));
```

maka hasilnya adalah 5

10) Strcat ()

Fungsi ini dipakai untuk menambah string asal ke bagian akhir string tujuan.

Bentuknya adalah sebagai berikut:

```
Strcat ( tujuan, asal) ;
```

11) Strcmp ()

Fungsi ini dipakai untuk membandingkan string str1 dengan string str2. Hasil dari fungsi ini bertipe int berupa nilai negatif (str1 kurang dari str2), nol(str1=str2), positif (str1 lebih besar dari str2).

Nilai absolute hasil fungsi (kecuali bernilai nol) menyatakan selisih ASCII dari karakter yang menyebabkan str1 berbeda dengan str2. Perbandingan dilakukan untuk karakter pada posisi yang sama dari str1 dan str2, dimulai dari karakter yang paling kiri.

Jadi huruf 'A' lebih kecil dari 'B' dan 'B' lebih kecil dari 'C' dan seterusnya.

12) Strlwr ()

Fungsi ini dipakai untuk mengubah setiap huruf capital atau huruf besar ke huruf kecil di dalam *string*.

Bentuknya adalah sebagai berikut:

```
Car ket [ ] = "TURBO C++"  
Strlwr (ket);  
Hasilnya : turbo c++
```


13) Strupr ()

Fungsi ini akan mengubah semua huruf kecil pada string menjadi huruf besar.

Bentuknya adalah sebagai berikut:

```
Car ket [ ] = "turbo c++"  
Strlwr (ket);  
Hasilnya : TURBO C++
```

TUGAS PRAKTIKUM

- a) **Buatlah program contoh program string dengan *isalpha* (simpan dengan nama lat9_1.cpp)**

```
#include <stdio.h>  
#include <ctype.h>  
int main (void)  
{  
    char c = 'C';  
  
    if (isalpha (c))  
        printf ("%c is alphabetical \n",c);  
    else printf ("%c is not alphabeical\n",c);  
    return 0;  
}
```

- b) **Buatlah program contoh program string dengan *isdigit* (simpan dengan nama lat9_2.cpp)**

```
#include <stdio.h>  
#include <ctype.h>  
int main (void)  
{  
    char c = 'C';  
  
    if (isdigit (c))  
        printf ("%c is a digit\n",c);  
    else printf ("%c is not a digit\n",c);  
    return 0;  
}
```

- c) **Buatlah program contoh program string dengan *islower* (simpan dengan nama lat9_3.cpp)**

```
#include <stdio.h>
#include <ctype.h>
int main (void)
{
    char c = 'C';

    if (islower (c))
        printf ("%c is lower case character \n",c);
    else printf ("%c is not lower case character\n",c);
    return 0;
}
```

- d) **Buatlah program contoh program string dengan *isupper* (simpan dengan nama lat9_4.cpp)**

```
#include <stdio.h>
#include <ctype.h>
int main (void)
{
    char c = 'C';

    if (isupper (c))
        printf ("%c is an uppercase character \n",c);
    else printf ("%c is not an uppercase character\n",c);
    return 0;
}
```

- e) **Buatlah program contoh program string dengan *strcpy* (simpan dengan nama lat9_5.cpp)**

```
#include <string.h>
#include <stdio.h>
int main (void)
{
    char str1 [10] = {"abcdefghi"};
    char str2 [10];

    strcpy (str2, str1);
    printf ("%s\n",str2);
    return 0;
}
```

- f) **Buatlah program contoh program string dengan *strcat* (simpan dengan nama lat9_6.cpp)**

```
#include <string.h>
#include <stdio.h>
int main (void)
{
    char destination [25];
    char blank [10]= {" "}, c [10]= {"C++"}, Borland [10] = {"Borland"};

    strcpy (destination, Borland);
    strcat (destination, blank);
    strcat (destination, c);

    printf ("%s\n", destination);
    return 0;
}
```

- g) **Buatlah program contoh penggunaan String (simpan dengan nama lat9_7.cpp)**

```
#include <iostream>
#include <stdlib.h>
#include <cstring>
char *entry, letter, choice [2];
int ascii, len, binary [8], total;
void prog ();

int main()
{
    prog ();
    return 0;
}

void prog()
{
    entry = new char [501];
    cout<<"Masukkan kalimat yang ingin dikonversi (sampai 500 huruf) : ";
    cin.getline(entry, 500);
    len = strlen (entry);
```

```

for (int i = 0; i<len; i++)
{
    total = 0;
    letter = entry[i];
    ascii = letter;
    while(ascii>0)
    {
        if ((ascii%2)==0)
        {
            binary [total] = 0;
            ascii = ascii/2;
            total++;
        }
        else
        {
            binary [total] = 1;
            ascii = ascii/2;
            total++;
        }
    }
    total--;
    while (total>=0)
    {
        cout<<binary[total];
        total--;
    }
}
delete[ ] entry;
cout<<endl<<"Akan diulangi lagi (1 =ya, 2= tidak) ? : ";
cin.getline(choice, 3);
if (choice [0]=='1')
{
    prog();
}
else
{
    exit(0);
}
}

```

TUGAS PENDAHULUAN

1. Jelaskan jenis-jenis fungsi String pada bahasa pemrograman C/C++!
2. Sebagai programmer, mengapa anda menggunakan fungsi String!
3. Buatlah contoh program sederhana mengcopy beberapa karakter menggunakan array!
4. Buatlah contoh algoritma dan program sederhana menggunakan salah satu Fungsi string untuk menghitung jumlah karakter dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program yang dapat menginputkan nama serta hobbi dan jawaban komputer sama dengan inputan yang dimasukkan!

PERTEMUAN X

POINTER

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar Pointer pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan Pointer pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Pointer adalah variabel yang berisi alamat memori sebagai penilaiannya dan berbeda dengan variabel biasa yang berisi nilai tertentu. Dengan kata lain, pointer berisi alamat dari variabel yang mempunyai nilai tertentu.

Adapun bentuk umum dari pernyataan variabel pointer dalam C++ adalah:

```
Type *variabel name;
```

Dengan:

- 1) Type adalah tipe dasar pointer
- 2) Variabel name adalah nama variabel pointer
- 3) * adalah operator memori yang fungsinya untuk mengembalikan nilai variabel pada alamatnya yang ditentukan oleh operand

b) Operator pointer

Ada beberapa operator yang bisa digunakan dalam pointer. Operator-operator itu adalah : operator alamat (yang dilambangkan dengan symbol &) adalah operator unary yang mengembalikan alamat dari operandnya.

Sebagai contoh, diasumsikan deklarasi sebagai berikut :

```
int y =5 ;  
int *yptr;
```

Maka pernyataan :

```
Yptr = &y;
```

Pernyataan ini mengandung arti bahwa alamat dari variabel y ditunjukkan kepada variabel y pointer yptr.

c) Pointer aritmetik

Hanya 4 operator aritmetik dapat digunakan pada pointer yaitu ++,==,+ dan -. Dalam hal ini asumsi untuk tipe data integer adalah 32 bit.

Perhatikan contoh berikut:

```
int *p1;//assume :p1==200
p1++;
p1--;
```

d) Array pointer

Pointer dapat di-array seperti tipe data yang lain dalam c++ integer. Sebagai contoh, untuk menyatakan sebuah array pi dari pointer sebanyak 10 buah data yang brtipe 10 integer, dapat dituliskan sebagai berikut:

```
int *pi[10];
```

Untuk menentukan alamat dari variabel integer disebut var ke elemen ke tiga dai pointer array, dapat di tuliskan sebagai berikut:

```
int var;
Pi [2]=&var
```

Pernyataan berikut member nilai var :

```
*pi[2]
```

TUGAS PRAKTIKUM

a) Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_1.cpp)

```
#include<iostream.h>
Int main(void)
{
int v =7, *p;
p = &v;
cout<<"nilai v ="<<v<<" dan *p ="<<*p<<"\n alamatnya ="<<p<<'\n';
}
```

- b) **Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_2.cpp)**

```
#include<iostream.h>
int main()
{
    int *ptr, num;
    ptr = & num;
    *ptr = 100;
    cout<< num <<" ";
    (*ptr)++;
    cout<<num <<" ";
    (*ptr)*2;
    cout <<num<<"\n";
    return 0;
}
```

- c) **Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_3.cpp)**

```
#include<iostream.h>
int main()
{
    int num[10];
    int *start, *end;
    start = num;
    end = &num[9];
    while(start != end)
    {
        cout<< "masukan bilangan : ";
        cin >> *start;
        start++;
    }
    return 0;
}
```


- d) **Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_4.cpp)**

```
#include<iostream.h>
int main()
{
    char *s;
    s = "pointer masih digunakan.";
    cout<< s;
    return 0;
}
```

- e) **Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_5.cpp)**

```
#include <iostream.h>
int main()
{
    int numbers[5], *p;
    p = numbers; *p=10;
    p++; *p=20;
    p = &numbers[2]; *p=30;
    p = numbers + 3; *p=40;
    p = numbers; *(p+4) = 50;
    for(int n=0; n <5; n++)
        cout<<numbers[n] <<" ";
    return 0;
}
```

- f) **Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_6.cpp)**

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
int main()
{
    int x, *p, **q;
    x =10; p = &x; q = &p;
    cout<< **q;
    return 0;
}
```

- g) **Buatlah program contoh penggunaan pointer (simpan dengan nama lat10_7.cpp)**

```
#include<iostream.h>
void duplicate(int&a, int&b, int&c)
{
    a*=2;
    b*=2;
    c*=2;
}
int main()
{
    int x=1, y=3, z=7;
    duplicate( x, y, z);
    cout<<"x=" << x << ", y="<<y<<"", z=" << z;
    return 0;
}
```

TUGAS PENDAHULUAN

1. Jelaskan yang dimaksud dengan Pointer!
2. Sebutkan dan jelaskan jenis-jenis Pointer!
3. Sebagai programmer, kapan anda menggunakan Pointer!
4. Buatlah contoh algoritma dan program sederhana menggunakan Pointer dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program untuk menghitung banyaknya karakter yang dimasukkan dengan menggunakan pointer!

PERTEMUAN XI

STRUCTURE

TUJUAN PRAKTIKUM

- Mahasiswa dapat menjelaskan konsep dasar Structure pada bahasa pemrograman C/C++.
- Mahasiswa dapat menjelaskan dan menggunakan Structure pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Struktur(structure) adalah sekumpulan variabel yang masing-masing tipe datanya berbeda, dan di kelompokkan kedalam satu nama. (Menurut pascal struktur juga dikenal sebagai *record*).Struktur membantu data-data yang rumit, khususbya dalam program yang besar, Karen sturktur memibiarkan sekelompok variabel diperlakukan sebagai satu unit.

Struktur ini sering digunakan untuk mendefinisikan suatu *record* data yang disimpan di dalam file. Struktur termasuk kedalam tipe data yang di bangkitkan (derived data type), yang disusun menggunakan objek dengan tipe yang lain.

Perhatikan definisi berikut:

```
Struct mhs
{
    Char *nama;
    Char *nim;
    Int uts, uas;
    Flot akhir;
    Char mutu;
}
```

b) Struktur dan fungsi

Operasi yang sering di terapkan pada struktur adalah proses menyalin atau menunjukan struktur sebagai unit, menggunakan alamatnya dan mengakses anggotanya. *Copy* dan *assignment* mencakup memberi argument ke fungsi dan menghasilkan nilai dari fungsinya juga.

Struktur bisa diletakan di awal oleh daftar *value* konstanta dan otomatis juga bisa ditempatkan di awal oleh operasi *assignment*. Sebuah struktur otomatis mungkin

juga diletakan di depan oleh tugas atau oleh panggilan fungsi yang menghasilkan struktur jenis yang tepat. Untuk menghubungkan nama struktur dan nama anggota digunakan symbol “,”

c) Array dalam struktur

Array disini fungsinya sangat penting untuk menyimpan nama dan bilangan bulat yang akan digunakan dalam proses perhitungan.

Supaya lebih memahami tentang pendefinisian dan penggunaan array dalam struktur perhatikan permasalahan berikut:

Anda diminta membuat algoritma dan program untuk menampilkan judul film dengan batas karakter yang dimasukkan adalah 50, dimana judul ini disimpan dalam array dan didefinisikan dalam struktur.

Algoritma dari permasalahan diatas adalah sebagai berikut :

- 1) Tentukan fungsi yang menampung judul film, dimana judul film tersebut ditampung dalam *array*
- 2) Masukan judul film
- 3) Masukan tahun film
- 4) Tampilkan film sesuai dengan yang diinputkan yang sesuai dengan fungsi pada langkah 1
- 5) Selesai

d) Pointer dalam struktur

Misalkan sebuah pointer yaitu *ptpelajar*, yang menunjukan kepada sebuah data yang mempunyai struktur PELAJAR seperti berikut:

```
Struct PELAJAR *ptpelajar;
```

Seperti pada pointer yang lain, deklarasi di atas tidak menyediakan sebarang tempat untuk record PELAJAR. Perlu dibuat record baru yang fungsinya menggunakan pointer. Misalakan *pelajar_baru*.

```
Ptpelajar = &pelajar_baru;
```

Dengan kondisi tersebut, pointer *ptpelajar* digunakan untuk mengganti tempat alamat pelajar baru, dan pointer *ptpelajar* ini ditunjukan dengan menggunakan symbol -> seperti berikut ini:

```
ptpelajar->nama = Khairul Ariffin;  
ptpelajar->kelas = 12345 ;
```

```
ptpelajar->fakultas = EKONOMI;
```

Sama dengan :

```
*ptpelajar.nama =Khairul Ariffin;  
*ptpelajar.kelas =12345;  
*ptpelajar.fakultas = EKONOMI;
```

TUGAS PRAKTIKUM

- a) **Buatlah program contoh penggunaan struktur (simpan dengan nama lat11_1.cpp)**

```
#include<iostream.h>  
#include<string.h>  
#include<stdlib.h>  
struct movies_t  
{  
    char title[50];  
    int year;  
} mine, yours;  
  
void printfmovie (movies_t movie);  
int main()  
{  
    char buffer [50];  
    strcpy (mine.title, "2001 A Space Odyssey");  
    mine.year = 1968;  
    cout<< "masukan judul film favorit: ";  
    cin.getline (yours.title,50);  
    cout<< " masukan tahun :";  
    cin.getline (buffer,50);  
    yours.year = atoi (buffer);  
    cout<< "judul film favorit yang ada:\n ";  
    printfmovie (mine);  
    cout<< " judul film favorit kamu adalah :\n ";  
    printfmovie (yours);  
    return 0;  
}  
  
void printfmovie (movies_t movie)  
{  
    cout<< movie.title;  
    cout<<" ( " << movie.year << ") \n";  
}
```

- b) **Buatlah program contoh penggunaan struktur (simpan dengan nama lat11_2.cpp)**

```
#include<stdio.h>
struct S1
{
    char *s;
    int i ;
    struct S1 *slp;
};
main( )
{
    static struct S1 a [ ]= {
        { "abcdef", 1, a+1 },
        { " ghijkl", 2, a+2},
        {" mnopqr", 3, a}
    };
    struct S1 *p = a;
    int i;
    printf("a [0].s = %s p->= %s a[2].slp->s=%s\n", a[0].s, p->s, a[2].slp->s);
    for (i=0; i<3; i++)
    {
        printf("--a[i].i = %d ", --a[i].i);
        printf("++a[i].s[3] = %c\n ", ++a[i].s[3]);
    }
}
```

- c) **Buatlah program contoh penggunaan struktur (simpan dengan nama lat11_3.cpp)**

```
#include<iostream.h>
#include<iomanip.h>
struct Employee_str
{
    char Name[25];
    int IdNum;
};
int main( )
{
    Employee_str employee[5];
    int loop;
    for(loop=0; loop<5; loop++)
    {
```

```

    cout<< "Enter name for employee" <<(loop+1)<<" : ";
    cin.getline(employee[loop].Name,80, '\n');
    cout<< "Enter ID #: ";
    cin>>employee[loop].IdNum;
    cin.ignore(80, '\n');
}
    cout<<"\nemployee data:\n";
    cout.flags(ios::left);
    cout<<setw(15)<<"Name"<<"\tID#\n_____\n";
    for(loop =0; loop<5; loop++)
    {
        cout<<setw(15)<<employee[loop].Name;
        cout<<"\t"<<employee[loop].IdNum<<endl;
    }
    cout<<endl;
}

```

- d) **Buatlah program untuk mencatat data siswa suatu sekolah yang terdiri dari field nama, alamat, telepon, dan jenis kelamin. Tampilan Output sebagai berikut. (simpan dengan nama lat11_4.cpp)**

```

Nama Siswa      : Andri
Alamat          : Klaten
Jenis Kelamin   : Pria
Telepon         : 0856712712

```

Mau tambah data lagi [Y/T] :

TUGAS PENDAHULUAN

1. Jelaskan yang dimaksud dengan Struktur pada bahasa C/C++!
2. Sebutkan dan jelaskan jenis-jenis Struktur!
3. Sebagai programmer, mengapa anda menggunakan Struktur!
4. Buatlah contoh algoritma dan program sederhana menggunakan Struktur dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program untuk mencatat data karyawan yang terdiri dari field jumlah karyawan, nama, gaji/bulan, serta status (belum menikah/menikah). Jika sudah menikah, tanya berapa anaknya. Tampilkan seluruh data dalam bentuk tabel!

PERTEMUAN XII

SORTING

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar Sorting pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan Sorting pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Sorting adalah suatu proses pengurutan data yang sebelumnya disusun secara acak atau tidak teratur menjadi urut dan teratur menjadi urut dan teratur menurut suatu aturan tertentu. Biasanya pengurutan terbagi menjadi 2 yaitu : ascending (pengurutan dari karakter/ angka kecil ke karakter / angka besar ke karakter/ angka kecil)

Ada banyak cara yang dapat dilakukan untuk melakukan proses pengurutan dari paling atas ke paling bawah atau sebaliknya.

Untuk melakukan proses pengurutan dapat menggunakan beberapa metode antara lain :

1) Bubble sort

Bubble sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya. Perbandingan alamatnya dapat dimulai dari data yang paling awal atau yang paling akhir. Apabila elemen yang sekarang (sebelumnya) lebih besar dari elemen berikutnya, maka posisi di tukar, kalau tidak posisinya tetap atau tidak perlu ditukar.

2) Selection sort

Selection sort adalah suatu metode pengurutan yang membandingkan elemen yang sekarang dengan elemen berikutnya sampai elemen terakhir. Jika ditemukan

elemen lain yang lebih kecil dari elemen sekarang maka di catat posisinya dan langsung ditukar.

3) Quick sort

Quick sort adalah suatu metode pengurutan yang membandingkan suatu elemen (*pivot*) dengan elemen yang lain dan menyusunkan sedemikian rupa sehingga elemen yang lain lebih kecil dari pada *pivot* teletak disebelah kiri *pivot* sedangkan elemen yang lebih besar dari *pivot* diletakan disebelah kanan *pivot*.

4) Merge sort

Merge sort adalah suatu metode pengurutan yang membandingkan elemen satu dengan elemen yang lain, apabila nilainya lebih kecil maka datanya ditampung di elemen yang lain lagi.

TUGAS PRAKTIKUM

- a) **Buatlah program contoh penggunaan sorting (simpan dengan nama lat12_1.cpp)**

```
#include <iostream.h>
#include <iomanip.h>
int main()
{
    int NumList[8] = {5, 34, 32, 25, 75, 42, 22, 2};
    int Swap;
    cout<<"Data sebelum diurutkan: \n";
    for(int ctr=0; ctr<8; ctr++)
    {
        cout<< setw(3) << NumList[ctr];
    }
    cout<<"\n\n";
    for ( int i=0; i<7; i++)
    for ( int ii=0; ii<7; ii++)
    if ( NumList[ii] > NumList [ii +1] )
    {
        Swap = NumList [ii];
        NumList [ ii ] = NumList[ ii +1 ];
        NumList [ ii + 1] = Swap;
    }
}
```

```

cout<<"Data setelah diurutkan : \n";
for (int iii=0; iii<8; iii++)
cout<< setw (3) << NumList [iii];
cout<<endl <<endl;
}

```

- b) Buatlah program contoh penggunaan sorting (simpan dengan nama lat12_2.cpp)**

```

#include<stdio.h>
#include <stdlib.h>
#include <string.h>
int sort_function (const void *a, const void *b);
char list[5][4] = {"cat", "car", "cab", "cap", "can"};
int main(void)
{
    int x;
    qsort (( void *)list, 5, sizeof(list[0]), sort_function);
    for (x = 0; x< 5; x++)
        printf("%s\n", list[x]);
    return 0;
}

int sort_function ( const void *a, const void *b)
{
    return ( strcmp ((char *) a, (char *)b) );
}

```

- c) Buatlah program contoh penggunaan sorting (simpan dengan nama lat12_3.cpp)**

```

#include <iostream>

void SwapMembers (int items[ ], int index1, int index2)
{
    int temp;
    temp=items[index1];
    items[index1] = items [index2];
    items[index2]=temp;
}

main()
{

```

```
int n, m;
int numbers [] ={ 102, 21, 83, 42, 11, 10, 9, 3, 20, 27, 15, 92, 2};
const int size = sizeof(numbers) / sizeof(numbers[0]);
for (n = size-1; n>0; n--)
for (m=0; m<n; m++)
    if(numbers [m] > numbers [m+1] )
        SwapMembers ( numbers, m, m+1 );
for (n=0; n<size;n++)
    cout<< numbers[n] << ", ";
return 0;
}
```

TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Sorting!
2. Sebutkan dan Jelaskan jenis-jenis Sorting pada bahasa C/C++!
3. Sebagai programmer, mengapa anda menggunakan Sorting!
4. Buatlah contoh algoritma dan program sederhana menggunakan Sorting dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program untuk mengurutkan data-data berikut ini dari yang terbesar ke yang terkecil!

4 , 8 , 5 , 9 , 6 , 2 , 7 , 5 , 9 , 5

PERTEMUAN XIII

SEARCHING

TUJUAN PRAKTIKUM

- a) Mahasiswa dapat menjelaskan konsep dasar Searching pada bahasa pemrograman C/C++.
- b) Mahasiswa dapat menjelaskan dan menggunakan Searching pada bahasa pemrograman C/ C++.

TEORI DASAR

a) Pendahuluan

Proses pencarian suatu elemen di dalam array disebut searching, ada 2 macam pencarian yaitu pencarian sekuensial(sequential searching) dan pencarian biner(binary searching). Perbedaannya terletak pada keadaan suatu elemen atau data yang berada pada array. Pencarian sekuensial digunakan apabila data dalam keadaan acak atau tidak urut. Sedangkan pencarian biner digunakan pada data yang sudah dalam keadaan urut.

b) Pencarian Sekuensial

Pencarian sekuensial menggunakan prinsip sebagai berikut : data yang ada pada suatu array dibandingkan satu persatuan dengan data yang dicari.

Pencarian ini dilakukan dengan melakukan suatu pengulangan dari 1 sampai semua data yang ada. Pada setiap kali pengulangan, dibandingkan data yang posisinya ke-i dengan data yang dicari atau maksud. Apabila sama, maka data tersebut telah ditemukan dan proses pengulangan dihentikan. Sebaliknya, kalau sampai pengulangan selesai dan data yang dicari tidak ditemukan, maka tersebut tidak ada.

Untuk memahami penggunaan pencarian sekuensial didalam suatu program, perhatikan masalah berikut ini.

a) **Buatlah program contoh penggunaan searching (simpan dengan nama lat13_1.cpp)**

61


```
        start=middle+1;
        middle=( start+end)/2;
    }
    while(start<=end && array [middle]!=element);
    if(array [middle]==element)
        position=middle;
    return position;
}
```

TUGAS PENDAHULUAN

1. Apa yang dimaksud dengan Searching!
2. Jelaskan kelebihan dan kekurangan Searching pada bahasa C/C++!
3. Sebagai programmer, mengapa anda menggunakan Searching!
4. Buatlah contoh algoritma dan program sederhana menggunakan Searching dengan memakai Flowchart!

TUGAS AKHIR

1. Buatlah program pencarian data-data, data-data yang akan dicari diinputkan dan kumpulan data-data yang akan dilakukan untuk proses pencarian juga diinputkan dan dimasukkan dalam array!