MODUL PRAKTIKUM
ALGORITMA DAN PEMROGRAMAN

Tim Dosen:
Akmal, S.Si., MT
Mira Suryani, S.Pd., M.Kom

Program Studi S-1 Teknik Informatika
Departemen Ilmu Komputer
Fakultas Matematika dan Ilmu Pengetahuan Alam
Universitas Padjadjaran
2018

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah melimpahkan rahmat dan karunia-Nya sehingga Buku Panduan Praktikum kuliah ini bisa diselesaikan.Buku Panduan Praktikum ini merupakan penunjang untuk mata kuliah Algoritma dan Pemrograman untuk digunakan di Program Studi S-1 Teknik Informatika, Departemen Ilmu Komputer, FMIPA Unpad. Buku ini berisi panduan percobaan serta tugas yang harus diselesaikan oleh peserta praktikum. Diharapkan mahasiswa bisa membuat dan mengembangkan program yang modular dan bisa mengolah data dengan berbagai macam data yang yang sederhana ataupun yang bentukan seperti *array*, *record* dan *file*. Implementasi dalam praktikum menggunakan bahasa C++ dengan *compiler* MinGW Developer Studio berdasarkan algoritma yang bersesuaian.

Terimakasih yang sebesar-besarnya penulis sampaikan kepada Koordinator Program Studi serta para asisten di Laboratorium Departemen Ilmu Komputer atas semua bantuan dan dorongan sehingga Buku Panduan Praktikum ini bisa diselesaikan. Penulis menyadari bahwa Buku ini masih banyak kekurangan. Oleh sebab itu masukan dari para pembaca sangat diharapkan untuk penyempurnaan dimasa yang akan datang.

Bandung, 1 Agustus 2018
Akmal,S.Si, MT dan Mira Suryani, S.Pd., M.Kom

DAFTAR ISI

BAB 1. PENDAHULUAN ALGORITMA DAN PEMROGRAMAN

Pada Bab 1 ini, peserta perkuliahan akan mempelajari konsep dasar mengenai algoritma, pemrograman, menuliskan program C++ dengan menggunakan aplikasi MinGW Developer Studio, dan mengerjakan tugas latihan.

1.1. Algoritma

Algoritma biasanya didefinisikan sebagai rangkaian terurut langkah-langkah yang logis dan sistematis yang disusun untuk menyelesaikan suatu masalah. Tujuan algoritma adalah memberikan petunjuk tentang langkah-langkah logika penyelesaian masalah dalam bentuk yang mudah dipahami nalar manusia sebagai acuan yang membantu dalam mengembangkan program komputer. Pemahaman terhadap algoritma akan mencegah sejak dini kemungkinan terjadinya kesalahan logika pada program komputer yang dikembangkan.

Penulisan (Notasi) algoritma:

Ada tiga macam bentuk notasi algoritma antara lain:

– Uraian deskriptif

– Diagram-alir (*flowchart*)

– *Pseudocode* (kode semu)

a. Uraian Deskriptif

Contoh: menyelesaikan permasalahan menghitung luas dan keliling suatu lingkaran

Algoritma Hitung_Luas_dan_Keliling_Lingkaran

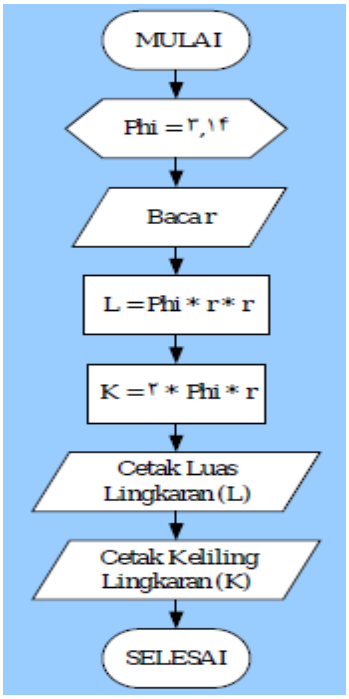
Deskripsi:

- Masukkan jari-jari lingkaran (r)
- Hitung luas lingkaran dengan rumus L = p * r2
- Hitung keliling lingkaran dengan rumus K = 2 * p * r
- Tampilkan luas lingkaran
- Tampilkan keliling lingkaran

b. Diagram Alir / *Flow Chart*

Flowchart adalah gambaran dalam bentuk diagram alir dari algoritma algoritma dalam suatu program, yang menyatakan arah alur program tersebut.

Contoh: menghitung luas dan keliling lingkaran yang algoritmanya dinotasikan dalam bentuk diagram alir (*flowchart*).



Gambar 1. Flowchart mencari luas dan keliling lingkaran.

c. Kode Semu / *Pseudo Code*

Notasi yang menyerupai notasi Bahasa pemrograman tingkat tinggi, misalnya Bahasa Pascal dan C.

Struktur algoritma dibagi ke dalam beberapa bagian, diantaranya:

1. Bagian kepala (*header*)
2. Bagian Deklarasi (definisi *variable*)
3. Bagian Deskripsi (rincian langkah)

Contoh:

```
Algoritma Luas_persegi_panjang
{Menghitung sebuah luas persegipanjang apabila panjang dan lebar persegipanjang tersebut diberikan}

Deklarasi
{Definisi nama peubah/variabel}
   float  panjang, lebar, luas

Deskripsi
   read(panjang,lebar)          //   bisa juga :  input   / baca
   luas ← panjang * lebar
   write(Luas)                  //  bisa juga  :  output   / tulis
```

1.2. Pemrograman

Program secara umum didefinisikan sebagai kumpulan instruksi atau perintah yang disusun sedemikian rupa sehingga mempunyai urutan nalar yang logis untuk menyelesaikan suatu persoalan yang dimengerti oleh komputer. Pemrograman adalah aktivitas yang berhubungan dengan pembuatan program dengan mengikuti kaidah bahasa pemrograman tertentu. Dalam konteks pemrograman terdapat sejumlah bahasa pemrograman seperti Pascal, Delphi, C, C++, C#, Java, dll. Dari contoh algoritma mencari luas persegi panjang bisa dihasilkan program dengan menggunakan beberapa Bahasa pemrograman seperti C++, sebagai berikut:

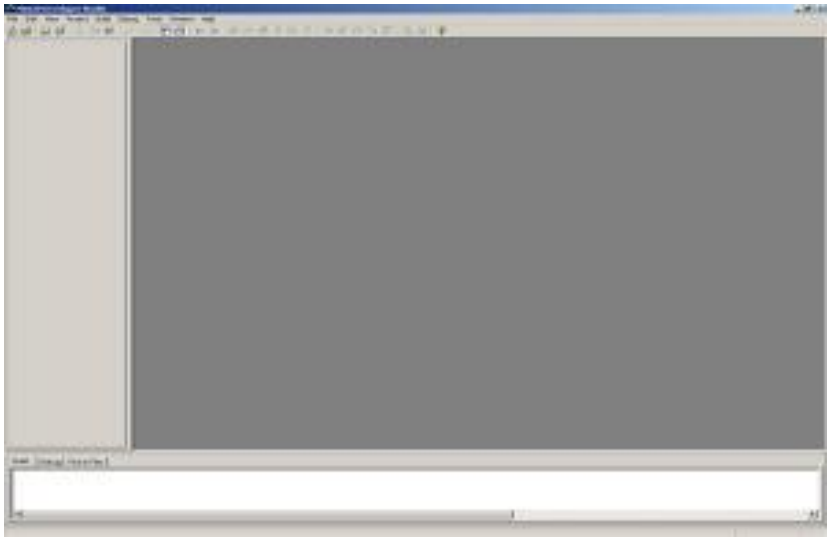
```
#include <iostream.h>

main(){
   float panjang, lebar, luas;
   cout << ”Panjang : ” ; cin >> panjang;
   cout << ”Lebar   : ” ; cin >> lebar;

   luas = panjang * lebar;
   cout << ”Luas   : ” << luas << endl;
}
```

1.3. Menuliskan Program C++ dengan MinGW Developer Studio

Percobaan 1.1: Mencoba *compiler* dan membuat program persegi panjang

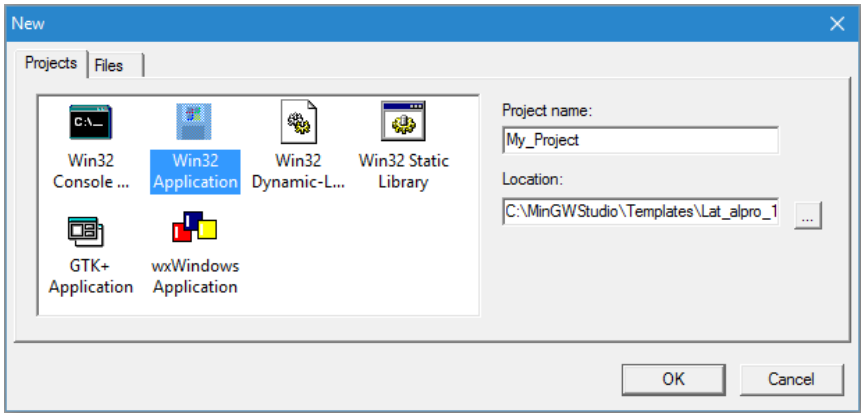
Langkah-langkahnya:

1. Bukalah software **MinGW Developer Studio.** Hasilnya adalah tampilan berikut ini



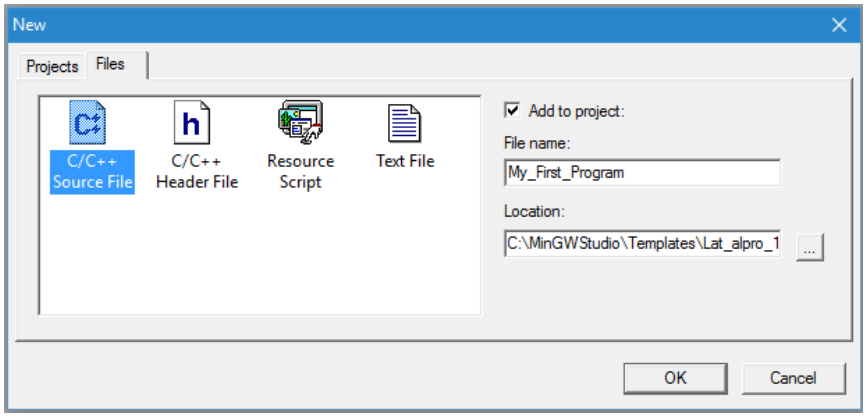
Gambar 2. Tampilan awal aplikasi MinGW Developer Studio.

2. Klik **File** >> **New**: Isikan **Project Name** sesuai keinginan anda misalnya My_Project. **Pilih Win32 Console Application >> Klik OK.**



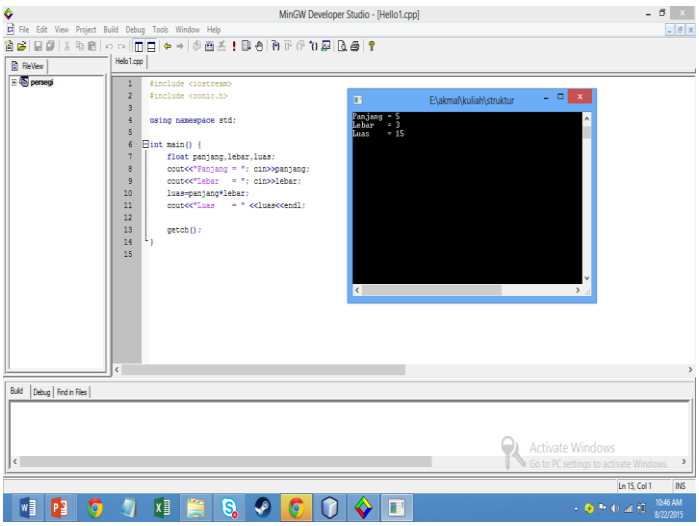
Gambar 3. Kotak dialog membuat *project* baru.

3. Klik **File** >> **New**, **Pilih C/C++ Source File** lalu isi **File Name** sesuai keinginan anda misalnya My_First_Program, lihat Gambar 4 berikut.



Gambar 4. Kotak dialog membuat file C/C++ baru.

4. Tuliskan kode program yang anda inginkan misalnya seperti ini dan pilih **Menu** >> **Compile** atau **tekan ctrl + F7**,



Gambar 5. Menjalankan kode program pada MinGW Developer Studio.

Jalankan program yang telah anda buat tadi cara nya dengan klik menu **Build** lalu pilih **Build and Execute** atau **tekan F8**, untuk melihat hasilnya.

TUGAS 1

1. Menyusun Algoritma

Dari permasalahan–permasalahan di bawah ini, susunlah algoritma untuk menyelesaikannya. Anda dapat menyusunnya dengan menggunakan *pseudo code* (a dan b) dan *flow chart* (c dan d).

a. Menghitung rata–rata dari 3 buah bilangan.
b. Dibaca dua buah nilai yang dihasilkan dari pengukuran arus (Ampere) dan Tahanan (Ohm), Hitunglah tegangan (Volt) yang dihasilkan.
c. Dibaca sebuah bilangan bulat (rupiah) yang positif, harus dihitung ekivalensinya dalam dollar ($) dan dituliskan hasilnya. Bagaimana dengan perubahan kurs yang sering terjadi? Apa yang harus diubah jika misalnya selain menghitung ekivalensi dalam $ juga harus dihitung ekivalensi dalam Yen, DM dan Euro?
d. Dibaca sebuah besaran riil, yang mewakili hasil pengukuran temperatur dalam derajat Celcius, Hitung ekivalensinya dalam derajat Fahrenheit, Rheamur dan Kelvin.

2. Pemrograman

Implementasikan program no a, b, c, dan d dengan menggunakan compiler C++.

BAB 2. PENGENALAN BAHASA C++

Pada bagian ini, peserta praktikum akan mengenal bahasa C++ dengan mempelajari dasar bahasa C++.

2.1 Dasar Bahasa C++

Susunan penulisan program dengan C++

```
/* Nama program    :
   Nama            : Akmal
   NPM             :
   Tanggal buat    :
   Deskripsi       :
**************************************************/
// deklarasi header file / Preprocessor directive
// deklarasi fungsi / void

main() {
  /* KAMUS */
  /* ALGORITMA */
}
```

Lakukan semua percobaan berikut ini dan periksa hasil yang ditampilkan. Tuliskan pada modul anda !

Percobaan 2.1: Komentar dan Hello

```
/*     Nama program        :  hello.cpp
       Nama                :  Akmal
       NPM                 :  1234
       Tanggal buat        :  17 Agustus
       Deskripsi           :  Pencetakan Hello Unpad 2 kali
       ------------------------------------------------------ */
       #include <iostream.h>
       using namespace std;
       main(){
              cout << ”Hello, Unpad.” <<endl;
              cout << ”Hello, ” << ”Unpad ”;
              cout << ”Bandung dan Jatinangor”;
       }

       ------------------------------------------------------
       #include <iostream.h>
       main() {
              int n = 66;
              cout<<n<<endl;
       }
```

Output:

```
{ 66
```

- Program diatas memperlihatkan 15 token, yaitu main, (,), {, int, n, =, 66, ;, cout, <<, endl, return, 0 dan }.
- Token n adalah suatu variable.
- Token 66 adalah suatu konstanta bertipe integer
- Token int, return dan endl adalah suatu keyword
- Token = dan << adalah operator
- Token (,), {, ;, dan } adalah tanda baca
- Baris pertama berisi suatu prepocessor directive yang bukan bagian sebenarnya dari program.

Percobaan 2.2: Konstanta, Variable dan Pencetakan

```
#include <iostream.h>
main(){
      const float phi = 3.14;
      float jari_jari, luas, keliling;
      jari_jari = 7.0;
      luas =  phi * jari_jari * jari_jari;
      keliling = 2 * phi * jari_jari;
      cout << " Luas Lingkaran      = " << luas <<endl;
      cout << " Keliling Lingkaran  = " << keliling;
}
```

Percobaan 2.3: Konstanta, Tipe data dan Ukurannya

Tabel 1. Daftar tipe data.

Tipe data	Byte	Batasan
char	1	Bilangan bulat / ASCII antara -128 s.d. 127
unsigned char	1	Bilangan bulat antara 0 s.d. 255
short	2	Bilangan bulat antara -32.768 s.d. 32.767 (-2^{15} s.d 2^{15}-1)
Unsigned short	2	Bilangan bulat antara 0 s.d. 65.535 (0 s.d 2^{16}-1)
int	4	Bilangan bulat antara -2.147.483.648 s.d. 2.147.483.647 (-2^{31} s.d 2^{31}-1)
unsigned int / unsigned	4	Bilangan bulat antara 0 s.d 2^{32}-1
long int	4	Bilangan bulat antara -2^{31} s.d 2^{31} -1

Tipe data	Byte	Batasan
unsigned long int	4	Bilangan bulat antara 0 s.d 2^{32}-1
float	4	Bilangan real antara - 3.4 E+38 s.d. 3.4E+38 (7 digit presisi)
double	8	Bilangan real antara -1.7E+308 s.d. 1.7E+308 (15 digit presisi)

```
#include <limits.h>
main() {
    cout<<17;                  //  hasil = ?
    cout<<017;                 //    ??
    cout<<0x17;                //    ??
    cout<<"Ukuran tipe integer: "<<sizeof(int)<<endl;
    cout <<"Bilangan minimum char: "<<CHAR_MIN<<endl;
    cout<< "Bilangan minimum Uchar: "<<UCHAR_MAX<<endl;
    cout <<"Bilangan maximum int : "<<INT_MAX;
}
```

Latihan:

Coba untuk semua tipe data, tampilkan nilai terendah dan nilai tertinggi serta ukuran semua tipe!!

Percobaan 2.4: Literal / String

```
#include <string.h>
main(){
      cout<<"abc\ndef");            // hasil = ??
      cout<<"abc\tdef");            //    ??
      cout<<"\"Halo\"");            //    ??
      cout<<"Panjang kata Unpad ="<< strlen("Unpad");

      cout<<strlen("Selamat Pagi. \n")<<endl;
      cout<<strlen("Selamat Pagi. ")<<endl;
      cout<<strlen("Selamat")<<endl;
      cout<<strlen("S")<<endl;
      cout<<strlen("")<<endl;
}
```

2.2 Operator

Percobaan 2.5 : Operator Aritmetika

```
#include <iostream.h>
void main ()
{     int m=82, n = 26;
      cout << m<<" + "<<n<<" = "<<m+n<<endl;
      cout << m<<" - "<<n<<" = "<<m+n<<endl;
      cout << m<<" * "<<n<<" = "<<m+n<<endl;
      cout << m<<" / "<<n<<" = "<<m+n<<endl;
      cout << m<<" % "<<n<<" = "<<m+n<<endl;
      cout <<" - "<<m<<" = "<<-m<<endl;
}
```

Percobaan 2.6: Operator *Increment* dan *Decrement*

```
#include <iostream.h>
void main () {
      int m=44, n = 66;
      cout <<"m = "<<m<<", n = "<<n<<endl;
      ++m;
      --n;
      cout <<"m = "<<m<<", n = "<<n<<endl;
      m++;
      n--;
      cout <<"m = "<<m<<", n = "<<n<<endl;
}
```

Percobaan 2.7: Contoh Kasus Lain Operator *Increment*

```
#include <iostream.h>
main () {
      int m=66, n;
      n = ++m;
      cout <<"m = "<<m<<", n = "<<n<<endl;
      n=m++;
      cout <<"m = "<<m<<", n = "<<n<<endl;
      cout <<"m = "<<m++<<endl;
      cout <<"m = "<<m<<endl;
      cout <<"m = "<<++m<<endl;
      cout <<"m = "<<m<<endl;
}
```

Percobaan 2.8: Contoh Kasus Lain Operator *Increment* dan *Decrement*

```
#include <iostream.h>
main (){
      int m=5, n;
      n = ++m * --m;
      cout <<"m = "<<m<<", n = "<<n<<endl;
      cout <<++m<< " "<<++m<<" "<<++m<<endl;
}
```

Percobaan 2.9: Operator Bitwise

```
#include <iostream.h>
main ()
{     int m=82, n=26;
      cout <<m<<" << 2"<<" = "<<(m<2)<<endl;
      cout <<m<<" >> 2"<<" = "<<(m>2)<<endl;
      cout <<m<<" & "<<n<<" = "<<(m&n)<<endl;
      cout <<m<<" ! "<<n<<" = "<<(m!n)<<endl;
      cout <<m<<" ^ "<<n<<" = "<<(m^n)<<endl;
      cout <<"~"<<m<<"<< = "<<~m<<endl;
}
```

Percobaan 2.10: Type Casting

```
main(){
      int jumlah=10,nData=3;
      float rata;
      cout<<(jumlah/nData);
      rata= float (jumlah) / nData;
      cout<<rata;
}
```

Percobaan 2.11: I/O Manipulator

```
#include <iostream.h>
main(){
    double pi = 3.141592654;
    // Tampilan default: left justified, presisi 6.
    cout << pi << endl;
    // Ubah dg precision 4, lebar field 12, isi dg #
    cout.precision(4);
    cout.width(12);
    cout.fill('#');
    cout << pi << endl;
    // Ubah presisi ke 10
    cout.precision(10);
    cout << pi << endl;
}
```

Percobaan 2.12: Contoh Kasus Lain I/O Manipulator

```
#include <iostream.h>
#include <iomanip.h>
#include <conio.h>
main(){
    int num = 37;
    double pi = 3.141592654;
    cout << "hex: " << hex << num << endl;
    cout << "oct: " << oct << num << endl;
    cout << "dec: " << dec << num << endl;
    cout << setw(8) << num <<endl;
    cout << setw(8) << setfill(' ') << num << endl;
    cout << "Pi: " << setprecision(10) << pi << endl;
}
```

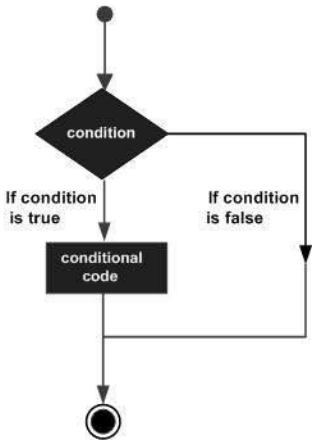
TUGAS 2

1. Buatlah program untuk menghitung luas bangunan geometri (bujursangkar, lingkaran, segitiga dan trapesium). Data masukkan dibaca dari piranti masukkan dan luas bangun ditampilkan sebagai keluaran.
2. Buatlah program untuk menghitung harga total suatu barang, dimana jumlah barangnya 5, harga per unit 5203.02.
3. Buatlah program untuk penggunaan operasi aritmatika yaitu penjumlahan, pembagian, perkalian, dan pengurangan dengan variabel yang diinputkan.
4. Buatlah program yang dapat membaca nama dan jam kerja pegawai. Kemudian, hitunglah honor pegawai tersebut jika upahnya perjam adalah Rp. 5.000,- Perhatikan bahwa upah perjam setiap pegawai tidak sama, dan perubahan upah tidak sesering perubahan kurs.
5. Terdapat sebuah program yang dapat membaca 3 buah bilangan bulat yang mewakili tiga buah tahanan dalam Ohm: R1, R2 dan R3. Kemudian hitung dan tuliskanlah tahanan total yang dihasilkan jika ketiganya dipasang seri dan parallel!

BAB 3. PEMILIHAN/SELEKSI

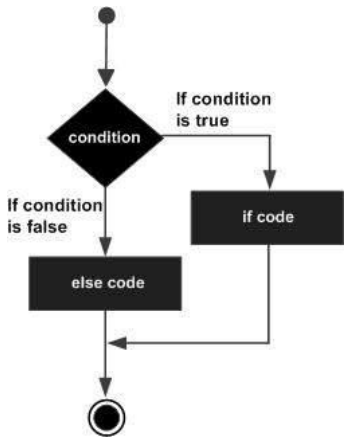
Struktur kontrol pemilihan adalah pernyataan yang mengizinkan *user* untuk memilih dan mengeksekusi blok kode spesifik dan mengabaikan blok kode yang lain.

Percobaan 3.1 : Analisa Kasus Tunggal (If)



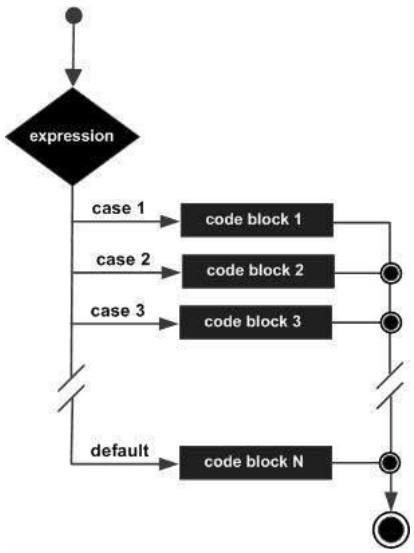

```
main(){
       int n,d;
        cout<<"Masukkan bil. Pertama :"; cin>>n;
        cout<<"Masukkan bil. Kedua   :"; cin>>d;
       if(n%d==0){
          cout<< n <<"  habis dibagi oleh "<<d<<endl;
        }
}
```

Percobaan 3.2 : Analisa 2 Kasus Komplementer (If Else)



```
main(){
     int i = 5;
     if(i%2==0){
        cout<<"i = " << i << ” adalah bilangan genap;
     }
     else {
        cout<<"i = "<< i <<” adalah bilangan ganjil";
     }
}
```

Percobaan 3.3: Analisa Banyak Kasus



Menggunakan if .. else if ..

```
      main() {
           int gol = 2;
           if(gol==1){
             cout<<"Gaji = 100");
           }
           else if(gol==2){
             cout<<"Gaji = 200 ");
           }
           else if(gol==3){
             cout<<"Gaji = 300");
           }
           else{
             cout<<"Golongan Salah");
           }
      }
```

Menggunakan Switch

```
main() {
    int gol = 2;
    switch (gol) {
       case 1  :  cout<<"Gaji = 100");
                  break;
       case 2  :  cout<<"Gaji = 200 ");
                  break;
       case 3  :  cout<<"Gaji = 300");
                  break;
       default :  cout<<"Golongan Salah");
                  break;
    }
}
```

Percobaan 3.4: Kasus Bersarang (Nested If)

```
main() {
      int a,b,c,max;
      cout<<”Masukkan 3 buah bilangan : “;
      cin >> a >> b >> c;
      if (a > b)
         if (a > c) max = a;         //a>b and a>c
         else  max = c;              //c>=a > b
      else
         if (b > c) max = b;         //b>=a and b>c
         else  max = c;              // c>= b >=a

      cout<<"Maksimum adalah : "<<max << endl;
}
```

Percobaan 3.5 : Kondisi Gabungan

Dalam hal ini menggunakan operator logical untuk mengkombinasikan kondisi yang akan diperiksa.

Tabel 2. Tabel kebenaran dari operator logika

X	Y	AND (&&)	OR (||)	! X
0	0	0	0	1
0	1	0	1	1
1	0	0	1	0

X	Y	AND (&&)	OR (‖)	! X
1	1	1	1	0

Operator And (&&)

```
main() {
      int a,b,c;
      cout<<"Masukkan 3 buah bilangan : ";
      cin >> a >> b >> c;
      cout<<"Nilai tertinggi adalah : ";
      if(a>=b && a>=c) cout << a << endl;
      if(b>=a && b>=c) cout << b << endl;
      if(c>=a && c>=b) cout << c << endl;
}
```

Operator Or (‖)

```
main() {
      char jawab;
      cout << "Yakin (y/t) : ";
      cin >> jawab;
      if (jawab=='Y' || jawab=='y')
          cout <<" OK, anda yakin" << endl;
      else
          cout <<" Maaf, anda tidak yakin.."<<endl;
}
```

Percobaan 3.6 : Operator Kondisional

Contoh 1

```
#include <iostream.h>
void main()

      char status;
      int grade = 80;
      //mendapatkan status pelajar
      status = (grade >= 60)? 'L' : 'G';
      //print status
      cout<< status ;
      }
```

Contoh 2

```
#include <iostream.h>
void main()
{
      int m = 26,n=82;
      int min = m < n ? m:n;
      cout <<Bilangan terkecil adalah”<<min<<endl;
}
```

Percobaan 3.7: Mencari Akar Persamaan Kuadrat : $AX^2 + BX + C = 0$.

```
#include <iostream.h>
#include <math.h>
main() {
   float a,b,c,disk,x1,x2;
   cout<<"Masukkan nilai a: "; cin>>a;
   cout<<"Masukkan nilai b: "; cin>>b;
   cout<<"Masukkan nilai c: "; cin>>c;

   if (a == 0) {
      cout<<"\nBukan persamaan kuadrat ..." ;
      return(0);
   }

   disk = b*b - 4*a*c;

   if (disk == 0){
         x1 = -b / (2*a);
         cout << "Dua akar real kembar"<<endl;
         cout << "x1 = x2 = "<< x1 <<endl;
   }
   else if (disk > 0){
         x1=(-b + sqrt(disk)) / (2*a);
         x2=(-b - sqrt(disk)) / (2*a);
         cout << "Dua akar real berlainan "<<endl;
         cout << "x1 = "<< x1 <<endl;
         cout << "x2 = "<< x2 <<endl;

   }
   else if (disk < 0){
      x1 = -b / (2*a);
      x2 = sqrt(-disk) / (2*a);
      cout << "Dua akar imajiner berlainan "<<endl;
      cout << "x1 = "<< x1 <<" + " << x2 << "i"<<endl;
      cout << "x2 = "<< x1 <<" - " << x2 << "i"<<endl;
   }
}
```

TUGAS 3

1. Buatlah program untuk menseleksi suatu bilangan dengan ketentuan sebagai berikut :

   0<=nilai <30 : Nilai rendah

   30<=nilai < 60 : Nilai sedang

   60<=nilai<=100 : Nilai tinggi

2. Buatlah sebuah program yang dapat menerima input tiga nilai ujian dari user (Quiz, UTS dan UAS) dan hitung nilai akhir dari nilai tersebut. Tampilkan output rata-rata dari tiga ujian. Tampilkan juga kata LULUS pada output jika nilai rata-rata lebih besar atau sama dengan 60, selain itu beri output: GAGAL.

   Nilai Akhir = 25% * Quiz + 35% * UTS + 40% * UAS

   Tentukan dan tampilkan hasil huruf mutu dengan ketentuan :

   0 <= nilai akhir < 45 : HM = “E”

   45 <= nilai akhir < 55 : HM = “D”

   55 <= nilai akhir < 68 : HM = “C”

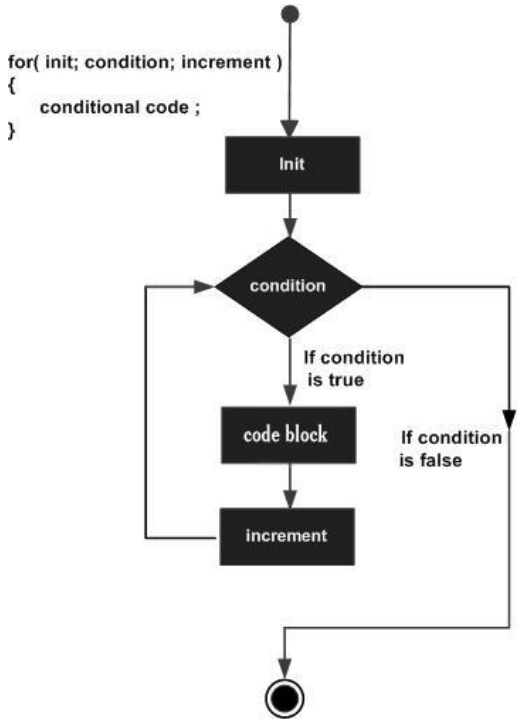
   68 <= nilai akhir < 80 : HM = “B”

   80 <= nilai akhir <= 100 : HM = “A”

BAB 4. PENGULANGAN/LOOPING

Terdapat 3 bentuk skema pengulangan yaitu **for**, **while** dan **do .. while** pada pemrograman. Perulangan for dengan while memiliki kemiripan yaitu melakukan aksi minimal nol kali, sedangkan do while melakukan aksi minimal satu kali. Kadang-kadang dikatakan juga bahwa for identik dengan while. Setiap bentuk pengulangan memiliki karakteristik tertentu dan para programer harus bisa membedakan kapan setiap bentuk tersebut digunakan sehingga pengerjaan program (algoritma) menjadi efektif dan efisien.

Percobaan 4.1: Pengulangan dengan Statement " for "



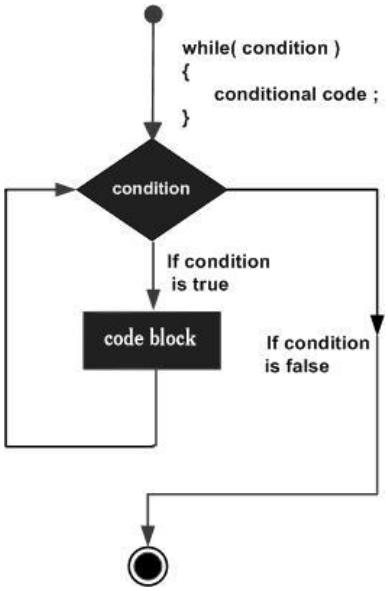
Mencetak Angka

```
main()
            for(int i=0; i<4; i++){
                  cout << i << endl;
            }
            for(int i=3;i>0;i--){
                  cout<<i<<endl;
            }
            for(int i=0; i <= 10 ; i += 2){
                  cout<<i << " ";
            }
            cout << endl;
            for(int i=10; i >= 0 ; i -= 2){
                  cout<<i << " ";
            }
  }
```

Menghitung Sigma(i) = 1 + 2 + 3 + ... + n (i = 1.. n) dengan for

```
main() {
   int  n, sigma = 0;
   cout << "Masukkan bilangan integer positif"; cin >> n;
   for (int i = 1; i <= n; i++) {
        sigma += i;
   }
   cout << "Jumlah dari << n <<" bil. pertama adalah : " <<
     sigma << endl;
}
```

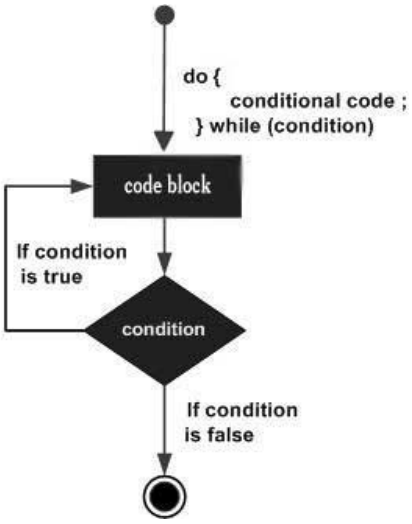
Percobaan 4.2: Pengulangan dengan Statement " while "



Menghitung Sigma(i) = 1 + 2 + 3 + ... + n (i = 1.. n) dengan while

```
main() {
    int i=1, n, sigma = 0;
    cout << "Masukkan bilangan integer positif"; cin >> n;
    while (i<=n) {
        sigma += i;
        i++;
    }
    cout << "Jumlahnya dari << n <<" bil. pertama adalah : "
       << sigma << endl;
}
```

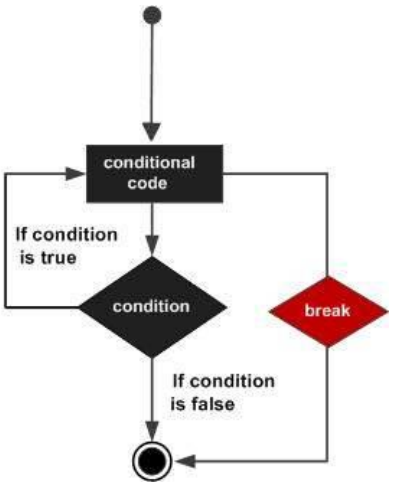
Percobaan 4.3: Pengulangan Dengan Statement ” do while ”



Program menghitung fungsi faktorial n! = (n) (n-1) ... (3) (2) (1)

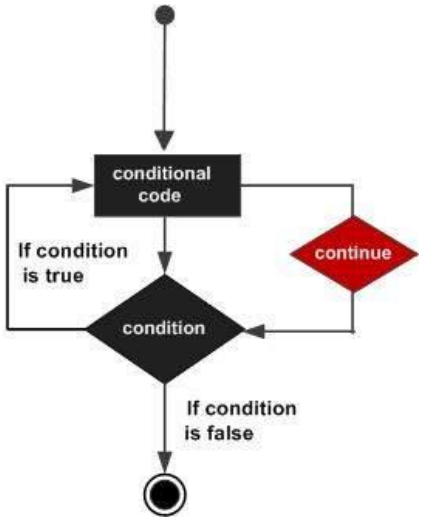
```
main() {
   int n, f = 1;
   cout << "Masukkan bil integer positif : "; cin >> n;
   do {
      f *= n;
      n--;
   } while (n >= 1);
   cout << n << " faktorial adalah : " << f << endl;
}
```

Percobaan 4.4: Pernyataan break dan continue



Menghitung Sigma(i) = 1+2+3+...+n dengan while (1) dan break

```
main()  {
   int i=1, n, sigma = 0;
   cout << "Masukkan bilangan integer positif"; cin >> n;
   while (1) {
       if (i > n) break; // loop berhenti disini ketika i>n
       sigma +=i;
       i++;
    }
    cout << "Jumlahnya dari << n <<" bil. pertama adalah: " << sigma <<
   endl;
}
```




```
main()
{
   int  n;
   for (; ;) {
      cout << "Masukkan integer : ";  cin >> n;
      if (n % 2 ==0) continue;
      else if (n % 3 == 0) break;
      cout << "\tAkhir loop " << endl;
    }
   cout << "\tLuar loop" << endl;
}
```

Percobaan 4.5: Perulangan Bersarang (Nested Loop)

Mencetak bintang sebanyak 3 baris dan 5 kolom.

```
main() {
      for (int i=1; i<=3; i++) {
            for (int j=1; j<=5; j++) {
                  cout << " * ";
            }
            cout << endl;
      }
}
```

Menggunakan nested loop while

```
main() {
      int i=1;
      while(i<=3) {
            int j=1;
            while (j<=5 ) {
                  cout << " * ";
                  j++;
            }
            cout << endl;
            i++;
      }
}
```

Untuk menampilkan posisi koordinat i,j bisa diganti dengan perintah:

```
cout  << i << ",," <<  j << "  ";
```

Percobaan 4.6: Studi Kasus 1

Dengan memanfaatkan posisi koordinat dengan ukuran n x n maka bisa dibuat berbagai bentuk pola. Misalkan dari ukuran 5 x 5 didapatkan pola sebagai berikut :

1,1	1,2	1,3	1,4	1,5
2,1	2,2	2,3	2,4	2,5
3,1	3,2	3,3	3,4	3,5
4,1	4,2	4,3	4,4	4,5
5,1	5,2	5,3	5,4	5,5

Akan dibentuk susunan bintang pada posisi diagonal sbb

```
*           *

   *     *

      *

   *     *

*           *
```

Maka akan dicetak bintang sesuai koordinat pada kondisi baris=kolom dan baris+kolom=6 (i == j || i + j == 6)

```
main() {
   for (int i=1; i<=5; i++) {
      for (int j=1; j<=5; j++) {
         if (i==j || (i+j==6)){
            cout << " * ";
         }
         else {
            cout << "   ";
         }
      }
      cout << endl;
   }
}
```

Kasus berikut juga untuk mencetak bintang:

```
main() {
   for (int i=1; i<=7; i++) {
      for (int j=1; j<=7; j++)  {
         if ((i+j<=8 && i<=j)||(i+j>=8 && i>=j)){
            cout << " * ";

         }
         else {
            cout << "   ";
         }
      }
      cout << endl;
   }
}
```

Percobaan 4.7: Studi Kasus 2

Akan dibuat sebuah program untuk membalikkan nilai integer yang diinputkan. Misalkan dimasukkan suatu angka integer 12345 maka akan dihasilkan angka integer 54321. Untuk menyelesaikan permasalahan ini digunakan operator modulo 10 untuk memotong angka yang paling kanan (satuan) dan pembagian dengan nilai 10 untuk membuang angka yang dipotong. Angka satuan yang dipotong secara berulang dijumlahkan dengan bilangan penampung (hasil) yang dikalikan dengan 10.

```
main() {
   long m, d, n=0;
   cout << "Masukkan satu bilangan integer positif :";
   cin >> m;
   while (m > 0) {
      d = m % 10;
      m /= 10;
      n = 10*n + d;
   }
   cout << "Hasil pembalikan bilangan adalah : " << n;
}
```

TUGAS 4

1. Buatlah program untuk mencetak table sinus dan kosinus dengan peningkatan nilai 30 yang dimulai dari nilai 0. Perhatikan perhitungan dalam radian.
   Bentuk judul table adalah: X Sinus(X) Cosinus(X)
   Gunakan skema perulangan while
2. Buatlah program untuk menampilkan nilai dari 1/X, X^2, X^3 mulai dari nilai 1 sd batas menggunakan skema perulangan for.
3. Buatlah program untuk mencari jumlah dari

   $$1/X + 1/X^2 + 1/X^3 \cdots + 1/X^n$$

   Dimana X dan n bilangan integer yang diinputkan oleh user.
4. Buat program untuk menampilkan output sbb (dicetak angka dan asterik)

   Banyak baris : 4 (diinputkan dari keyboard)

   1 * * * *
   2 * * *
   3 * *
   4 *

   1. *
   2. * *
   3. * * *
   4. * * * *

   Gunakan skema nested loop dengan a. 2 buah for
   b. 2 buah while
5. Buat program untuk menampilkan output sbb

   ```
   *          * * *          *
   * *        * *          * * *
   * * *  atau *    atau  * * * * *
   * *        * *          * * *
   *          * * *          *
   ```

6. Berdasarkan posisi bintang dalam koordinat dari percobaan buatlah beberapa bentuk hasil eksplorasi anda. Minimal 4 bentuk yang dibuat.
7. Berdasarkan percobaan, buatlah program untuk:
   i. Mengubah bilangan decimal menjadi biner
   ii. Mengubah bilangan decimal menjadi octal
   iii. Mengubah bilangan biner menjadi bilangan decimal
   iv. Mengubah bilangan octal menjadi bilangan desimal

BAB 5. FUNGSI/FUNCTION

Terdapat lima konsep dasar mengenai fungsi dalam pemrograma, yaitu:

- Fungsi adalah objek (bagian program/rutin) yang mengerjakan suatu tugas tertentu dan digunakan untuk memodularkan program dengan suatu ciri mengembalikan suatu nilai (*return value*)
- Fungsi dapat digunakan untuk menghindari penulisan yang sama yang ditulis berulang-ulang.
- Semua variabel yang dideklarasikan dalam fungsi merupakan variabel lokal, yang hanya diketahui dalam fungsi bersangkutan
- Fungsi bisa memiliki parameter yang menyediakan komunikasi antara fungsi dengan bagian yang memanggil fungsi tersebut (Parameter formal dan Parameter aktual).
- Parameter bersifat lokal.

Percobaan 5.1: Penulisan Model Fungsi (*Program mencari kuadrad bil.)*

Model 1:

```
// Definisi Fungsi lengkap di atas main program
int pangkat2(int x){
       return ( x * x);
}

main() {
       int  n;
       cin >> n;
       cout << pangkat2(n) << endl;    // Testing fungsi
}
```

Model 2:

```
int pangkat2(int x);        //Header / deklarasi fungsi

main() {
       int n;
       cin >> n;
       cout << pangkat2(n) << endl;  //Testing fungsi
}

int pangkat2(int x) {        // definisi fungsi di bawah main
       return ( x * x);
}
```

Percobaan 5.2: Pengiriman Secara Nilai pada Fungsi void

Fungsi Cetak Bintang Sebanyak m Baris dan n Kolom

```
void cetakBintang(int baris, int kolom){
      cout << ”Pencetakan bintang : ” << endl;
      for (int i=1; i<=baris; i++) {
            for (int j=1; j<=kolom; j++) {
                  cout << " * ";
            }
            cout << endl;
      }
}

main() {
      int baris,kolom;
      cout << ”Banyak baris : ”; cin >> baris;
      cout << ”Banyak kolom : ”; cin >> kolom;
      cetakBintang(baris,kolom);   // pemanggilan
}
```

Percobaan 5.3 : Pengiriman Secara Acuan pada fungsi void

Fungsi untuk Swap (Tukar Bilangan)

```
void swap( float& x, float& y){
      float temp = x;
      x = y;
      y = temp;
}
main() {
      float bil1 = 2, bil2 = 3;
      cout << bil1 << ” , ” << bil2 << endl;   // 2   3
      swap(bil1,bil2);   // pemanggilan

      cout << ”Hasil Swap / Tukar : ”;
      cout << bil1 << ” , ” << bil2 << endl;   // 3   2
}
```

Percobaan 5.4: Fungsi Boolean

Fungsi untuk menentukan bilangan prima. Akan diperiksa suatu bilangan apakah bilangan prima atau bukan. Jika bilangan prima maka akan dikembalikan nilai 1 dan jika salah akan dikembalikan nilai 0.

```
#include <iostream.h>
#include <math.h>

using namespace std;

int isPrima(int bil){
   float sqrtBil = sqrt(float(bil));
   if (bil < 2) return 0;           // false
   if (bil == 2) return 1;          // true
   if (bil % 2 == 0) return 0;      // false
   for (int gjl=3; gjl <= sqrtBil; gjl+=2) {
       if (bil % gjl == 0) return 0;  //  false
   }
   return 1;                        // true
}
main() {
   cout << "Bilangan-bilangan prima" << endl;
   for (int n=1; n < 100; n++) {
       if (isPrima(n))
           cout << n << " " ;
   }
   cout << endl;
}
```

Percobaan 5.5: Studi Kasus: Mencari Luas Persegi Panjang.

Diketahui bahwa rumus untuk mencari luas persegi panjang adalah panjang * lebar. Akan dikembangkan suatu program modular dengan menggunakan pola input →proses → output, artinya akan dibuat fungsi untuk input data berbentuk fungsi void, fungsi untuk proses berbentuk fungsi dan fungsi void serta fungsi untuk output berupa pencetakan data.

```
void inputData(float& pjg, float& lbr);
float hitungLuas(float pjg,float lbr);
void cariLuas(float pjg,float lbr, float& hsl);
void cetakData(float pjg, float lbr, float hsl);

main() {
      float panjang,lebar,luas;
      inputData(panjang,lebar);
      luas=hitungLuas(panjang,lebar);  // fungsi
      cetakData(panjang,lebar,luas);

      cariLuas(panjang,lebar,luas);    // void
      cetakData(panjang,lebar,luas);
}

void inputData(float& pjg, float& lbr){
      cout<<"Input Panjang = "; cin>>pjg;
      cout<<"Input Lebar   = "; cin>>lbr;
}

float hitungLuas(float pjg,float lbr){
       return (pjg*lbr);
}

void cariLuas(float pjg,float lbr, float& hsl){
      hsl=pjg*lbr;
}

void cetakData(float pjg, float lbr, float hsl){
      cout<<"Panjang = " << pjg <<endl;
      cout<<"Lebar   = " << lbr <<endl;
      cout<<"Luas    = " << hsl <<endl;
}
```

TUGAS 5

1. Tentukan hasil program berikut dan perlihatkan langkah-langkahnya.

```
void fungsi(int a, int &b, int &c){
    b =  ++a;
    c +=  b--;
    a = b + c;;
    cout << a  << b << c;
}

main(){
     int  a=2, c=2, b=2, y=2;
     fungsi (c,a,b);
     cout << a << b << c << y;
     fungsi (a + b,c,y);
     cout  << a << b << c  << y;
}
```

2. Ubahlah program yang ada pada studi kasus 1 dan studi kasus 2 di bab 3 (perulangan) menjadi modular berbentuk fungsi.
3. Diketahui rumus faktorial n! = 1*2*3...* n

   Buatlah program yang terstruktur dan modular untuk mencari :

   a. Permutasi P(n,r) = n! / (n-r)!
   b. Kombinasi C(n,r) = n! / (n-r)! * r!
4. Pecah masalah menjadi beberapa fungsi antara lain : fungsi input, fungsi faktorial, fungsi permutasi / kombinasi, fungsi output, dan fungsi utama (main).
5. Buatlah fungsi untuk menghitung jumlah deret pecahan :

   1 – 1/3 + 1/5 – 1/7 + 1/9 + . . . ± 1/N

   N adalah bilangan bulat positif

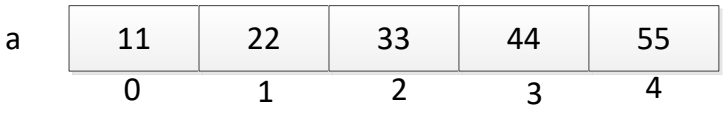
BAB 6. LARIK/ARRAY

Array adalah suatu tipe data yang menyimpan sekumpulan elemen data yang bertipe sama, dan memiliki indeks. Indeks array harus tipe data yang menyatakan keterurutan, misalnya *integer* atau karakter, dengan penyimpanan di memori secara kontinyu.

Setiap elemen array / larik satu dimensi ditulis dengan notasi :

$$\sum_{i=0}^{n-1} a[i] = a[0], a[1], a[2], a[3], a[4], \ldots, a[n-1]$$

Angka di dalam kurung siku menyatakan indeks array yang dimulai dari 0 sampai dengan (n-1) yang diinginkan.



Percobaan 6.1: Deklarasi Array

```
main() {
   int data[10];
   int n;

   cout<<"Banyak data : "; cin>>n;
   for (int i=0; i<n; i++) {
        cout<<"Data  : "; cin>>data[i]);
   }

   for(int i=0; i<n; i++) {
       cout<<”Data”<<i+1<<” = ”<<data[i];
   }
}
```

Percobaan 6.2 : Pendeklarasian Array Menggunakan Alias (Typedef)

```
typedef int larik[10];

main() {
   larik data;
   int n;

   cout<<"Banyak data : "; cin>>n;
   for (int i=0;i<n;i++) {
        cout<<"Data  : "; cin>>data[i]);
   }

   for(int i=0;i<n;i++) {
       cout<<”Data ”<<i+1<<” = ”<<data[i];
   }
}
```

Percobaan 6.3: Inisialisasi Array

```
main() {
   int data[]  = {11, 22, 33, 44, 55};

   for(int i=0;i<n;i++) {
       cout<<”Data ”<<i+1<<” = ”<<data[i] << “  “;
   }
}
```

Percobaan 6.4: Passing Array ke Fungsi

```
void banyakData(int& n);
void isiLarik(int a[], int n);
void printLarik(int a[], int n);

main() {
  int x[10];       // variabel array x sebanyak maks=10
  int n;
  banyakData(n);
  isiLarik(x,n);
  printLarik(x,n);
}

void banyakData(int& n){              // Input banyak data
   cout<<"Banyak data : "; cin>>n;
}

void isiLarik(int a[], int n){      // Input data larik
  for (int i=0;i<n;i++) {
     cout<<"Masukkan data ke- "<<(i+1)<<" : "; cin>>a[i];
  }
}

void printLarik(int a[], int n){    //Mencetak data larik
  cout <<"Data yang sudah dimasukkan" <<endl;
  cout <<"-----------------------------"<<endl;
  for(int i=0;i<n;i++) {
     cout <<"Data ke-"<<(i+1)<<" = "<< a[i] <<endl;
  }
}
```

Terlihat bahwa fungsi:

```
void isiLarik(int a[], int n)
```

dipanggil dengan cara :

```
int x[10];
isiLarik(x,n);
```

Percobaan 6.5 : Menggunakan Parameter di Fungsi Bertipe Array Alias

```
typedef int larik[10];

void banyakData(int& n);
void isiLarik(larik& a, int n);
void printLarik(larik a, int n);

main() {
  larik x;       // variabel array x
  int n;
  banyakData(n);
  isiLarik(x,n);
  printLarik(x,n);
}

void banyakData(int& n){             // Input banyak data
   cout<<"Banyak data : "; cin>>n;
}

void isiLarik(larik& a, int n){      // Input data larik
  for (int i=0;i<n;i++) {
     cout<<"Masukkan data ke- "<<(i+1)<<" : "; cin>>a[i];
  }
}

void printLarik(larik a, int n){    //Mencetak data larik
  cout <<"Data yang sudah dimasukkan" <<endl;
  cout <<"----------------------------"<<endl;
  for(int i=0;i<n;i++) {
     cout <<"Data ke-"<<(i+1)<<" = "<< a[i] <<endl;
  }
}
```

Terlihat bahwa fungsi:

```
void isiLarik(larik& a, int n)
```

dipanggil dengan cara :

```
larik x;
isiLarik(x,n);
```

Percobaan 6.6: Lengkapi untuk Mencari Nilai Rata-rata dan Nilai Maksimum

```
void cariRata(larik a, int n, float& rata){
  float jumlah=0;                // Ubah jadi fungsi ?
  for (int i=0;i<n;i++) {
      jumlah=jumlah+a[i];
  }
  rata=jumlah/n;
}


int maksimum (larik a, int n){   // Ubah jadi void?
      int maks = -999;
      for (int i = 0; i < n; i++) {
            if (maks < a[i]) {
                  maks = a[i];
            }
      }
      return (maks);
}
```

Percobaan 6.7: Operasi Input String Sebagai *array of character*

```
#include <iostream.h>
main (){
      char mybuffer [100];
      cout << "What's your name? ";
      cin.getline (mybuffer,100);
      cout << "Hello " << mybuffer << ".\n";
      cout << "Which is your favourite team? ";
      cin.getline (mybuffer,100);
      cout << "I like " << mybuffer << " too.\n";
}
```

Percobaan 6.8: Array 2 Dimensi (Matriks)

```
typedef int  matriks[10][10];


main() {
  matriks x;          // variabel array x bertipe matriks
  int nBaris, nKolom;


  banyakData(nBaris, nKolom)
  isiMatriks(x,nBaris, nKolom);
  cetakMatriks(x,nBaris, nKolom);
}
```

```
void banyakData(int& nBaris, int& nKolom){
       cout << "Banyak baris : "; cin >> nBaris;
       cout << "Banyak kolom : "; cin >> nKolom;
}
void isiMatriks(matriks& x, int nBaris, int nKolom){
    for (int i=0; i < nBaris; i++) {
       for (int j=0; j < nKolom; j++) {
              cout<<"Data ke-["<<i+1<<","<<j+1 <<"] : ";
              cin >> x[i,j];
       }
    }
}
void cetakMatriks(matriks x, int nBaris, int nKolom){
       cout << "\nPencetakan Matriks :"<<endl;
       for (int i=0; i < nBaris; i++) {
              for (int j=0; j < nKolom; j++) {
                     cout << x[i,j] << "  ";
              }
              cout << endl;
       }
}
```

Percobaan 6.9: Sorting

Sorting bisa didefinisikan sebagai suatu pengurutan data yang sebelumnya tersusun secara acak menjadi data yang tersusun secara teratur menurut aturan tertentu.Terdapat beberapa algoritma yang umumnya digunakan untuk melakukan sorting, antara lain: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, Quick Short, dan Shell Sort.

Kasus 1: Bubble Sort

```
void swap (int& x, int& y){
      int temp = x;
      x = y;
      y = temp;
}
void bubbleSort(larik& a, int n) {
      for (int i=n-1; i > 0; i--){
            for (int j=0; j < i; j++) {
                  if (a[j] > a[j+1] )
                        swap (a[j], a[j+1]);
            }
      }
}
```

Kasus 2: Insertion Sort

```
void insertion_sort (int arr[], int length){
                int j, temp;

        for (int i = 0; i < length; i++){
                j = i;

                while (j > 0 && arr[j] < arr[j-1]){
                          temp = arr[j];
                          arr[j] = arr[j-1];
                          arr[j-1] = temp;
                          j--;
                          }
                }
```

Kasus 3: Selection Sort

```
void selectSort(int arr[], int n)
{
//pos_min adalah posisi terpendek dari nilai minimum

        int pos_min,temp;

        for (int i=0; i < n-1; i++)
        {
            pos_min = i;

//set pos_min menjadi current index of array

                for (int j=i+1; j < n; j++)
                {

                if (arr[j] < arr[pos_min])
                    pos_min=j;
//pos_min akan menyimpan indeks yang mengandung nilai paling
minimum, hal ini dibutuhkan ketika proses swap terjadi
                }

//jika pos_min tidak lagi sama dengan i maka nilai yang lebih
kecil harus ditemukan, maka sebuah pertukaran harus terjadi
             if (pos_min != i)
             {
                   temp = arr[i];
                   arr[i] = arr[pos_min];
                   arr[pos_min] = temp;
             }
        }
```

Percobaan 6.10: *Linear Searching* (Pencarian)

```
main() {
   larik x;
   int n,kunci,found,lokasi;
   cout << "Kunci Pencarian data : " ; cin >> kunci;
   linearSearch(x, n, kunci, found, lokasi);
   if (found)
       cout << "Ditemukan di posisi : " << lokasi+1 ;
   else
        cout << "Tidak ditemukan";

}

void linearSearch(larik a, int n, int kunci, int& found,
                  int& lokasi){
      found = lokasi = 0;
      while (!found && lokasi < n) {
            if (a[lokasi] == kunci){
                  found = 1;
            }
            else {
                  lokasi=lokasi+1;
            }
      }
}
```

TUGAS 6

1. Buatlah program modular Matriks berikut:
   a. Penjumlahan 2 buah matriks
   b. Perkalian 2 buah matriks
   c. Transpose suatu matriks
   d. Mencari jumlah dari setiap baris dan kolom suatu matriks. Hasil penjumlahan disimpan dalam suatu array 1 dimensi

      Contoh :

      A = 1 0 4 jBaris = 5 jKolom = 8
      5 2 1 8 3
      2 1 2 5 7

      Fungsi yang diperlukan antara lain : main, inputMatriks, cetakMatriks, cariJumlahBaris, cariJumlah Kolom, cetakLarik.
2. Buatlah program modular untuk mencari nilai standar deviasi dari kumpulan data bertipe integer.

BAB 7. RECORD/STRUCTURE

Record adalah suatu tipe data yang merupakan kumpulan dari atribut-atribut (field) suatu objek. Pada record tipe elemen bisa berbeda-beda tidak seperti array yang mengharuskan mempunyai tipe elemen yang sama.

Atribut 1	Atribut 2	Atribut 3	. . .	Atribut N

Deklarasi record / structure

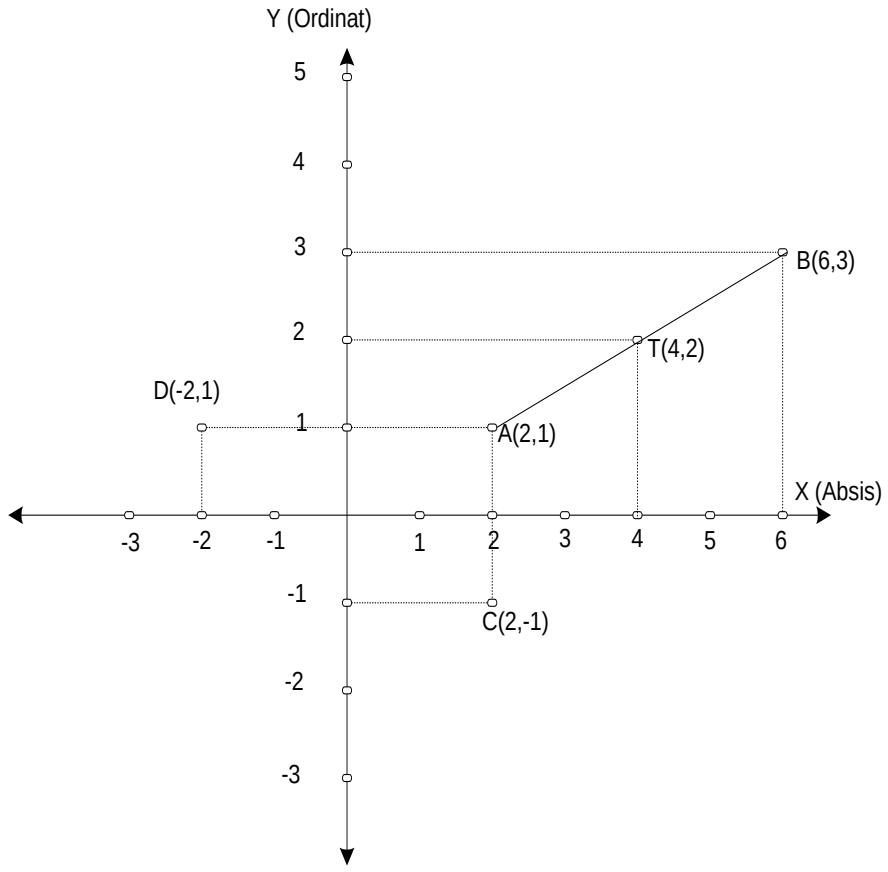
```
struct namaRecord {
       <tipe Atribut 1> Atribut1;
       <tipe Atribut 2> Atribut2;
       . . . . . .
       <tipe Atribut n> Atributn;
};
```

Cara mengakses variabel di dalam record/struct adalah dengan operator dot (.). Misalkan terdapat nama record/struct *mhs*, dan variabel yang akan diakses di dalamnya adalah *npm*, maka cara mengaksesnya adalah " *mhs.npm* "yang artinya kita mengakses *npm* yang merupakan satu atribut dari data *mhs*.

Percobaan 7.1: Koordinat Kartesian

Struktur data yang digunakan adalah tipe record / structure sebagai berikut:

```
struct koordinat {
     float absis;
     float ordinat;
};
```



Gambar 6. Titik dalam Koordinat Kartesian

A, B, C, D, T adalah titik-titik yang ada dalam koordinat kartesian

T merupakan titik tengah dari 2 titik A dan B,

C merupakan titik hasil dari pencerminan titik A terhadap sumbu X

D merupakan titik hasil dari pencerminan titik A terhadap sumbu Y

```
struct koordinat {
     float absis;
     float ordinat;
};

void getPoint( koordinat& ttk){
  cout<<"Masukkan absis   = ";cin >> ttk.absis;
  cout<<"Masukkan ordinat = ";cin >> ttk.ordinat;
}

void printPoint( koordinat ttk){
  cout<< "("<<ttk.absis<<","<<   ttk.ordinat<<")"<<endl;
}
```

```
main(){
 koordinat a,b,c;
 cout << "Input Titik a "<<endl; getPoint(a);
 cout << "Input Titik a "<<endl; getPoint(b);
 cout << "Titik a = ";  printPoint(a);
 cout << "Titik b = ";  printPoint(b);
}
```

Percobaan 7.2: Konversi Waktu

Struktur record waktu adalah sbb:

```
struct waktu {
     int jam;
     int menit;
     int detik;
};
```

```
struct waktu {
     int jam;
     int menit;
     int detik;
};

void getWaktu(waktu& wkt) {
  cout<<"Masukkan jam   = ";cin >> wkt.jam;
  cout<<"Masukkan menit = ";cin >> wkt.menit;
  cout<<"Masukkan detik = ";cin >> wkt.detik;
}

void printWaktu( waktu wkt) {
  cout << wkt.jam << ":"<< wkt.menit << ":" << wkt.detik ;
}

int cariJumlahDetik(waktu wkt){
    return (wkt.jam * 3600 + wkt.menit*60 + wkt.detik);
}

void konversiWaktu(int jDetik, waktu& wkt){
       int sisa;
       wkt.jam = jDetik / 3600;
       sisa = jDetik % 3600;
       wkt.menit = sisa / 60;
       wkt.detik = sisa % 60;
}

main() {
   waktu time1;
   int jDetik;
   cout << "Input Waktu  " << endl; getWaktu(time1);
   cout << "Cetak Waktu  " << endl; printWaktu(time1);

   jDetik = cariJumlahDetik(time1);


   cout << "Jumlah detik adalah : " << jDetik << endl;

   konversiWaktu(jDetik, time1);
   cout << "Cetak Waktu  " << endl; printWaktu(time1);

}
```

Latihan:

- **Ubah** cariJumlahDetik ke bentuk void
- Ubah konversiWaktu ke bentuk fungsi

TUGAS 7

1. Buatlah fungsi untuk:
   a. Mencari titik tengah (T) dari 2 buah koordinat A dan B. Diketahui rumus untuk titik tengah adalah:

      T.absis = (A.absis + B.ordinat) / 2

      T.ordina t= (A.ordinat + B.ordinat) / 2

   b. Mencari koordinat hasil pencerminan terhadap sumbu X dan terhadap sumbu Y

2. Buatlah program untuk mencari jumlah biaya parkir dari selisih 2 waktu (datang dan pulang) dengan aturan 1 jam pertama = Rp. 3000 dan per jam berikutnya = Rp. 3000.

   Contoh tampilan yang diinginkan:

Jam Datang	=	10 : 10 : 10
Jam Pulang	=	11 : 13 : 25
Lama Parkir	=	1 : 3 : 15
Bayar	=	Rp. 6000

BAB 8. OPERASI FILE

Dalam bahasa pemrograman C++ dapat dilakukan operasi file dengan menggunakan metode ifstream, ofstream, dan fstream.

Percobaan 8.1: Operasi File Text

File Teks: file yang pola penyimpanan datanya dalam bentuk karakter. Dipakai untuk menyimpan data seperti karakter atau string.

Operasi dasar pada file pada prinsipnya terbagi menjadi 3 tahap, yaitu:

- membuka atau mengaktifkan file (Open File)
- melaksanakan pemrosesan file
- menutup file (Close File)

Ada beberapa fungsi yang disiapkan yaitu:

```
void bukaTulisIsiFile(char nFile[]);
void tambahIsiFile(char nFile[], char isi[]);
void periksaFile(char nFile[]);
void cetakIsiFilePerKarakter(char nFile[]);
void 8isiFilePerKarakter(char nFile[]);
int cariJumlahHurufA(char nFile[]);
```

Menulis teks ke dalam file

```
void bukaTulisIsiFile(char nFile[]){
       ofstream fileteks;
       fileteks.open(nFile);
       fileteks << "Selamat Datang Mahasiswa Teknik Informatika
                    Unpad "<< endl;
       fileteks << "dalam Mata Kuliah Algoritma dan
                    Pemrograman"<< endl;
       fileteks << "Semester 1 th ajaran baru" << endl;
       fileteks << "Selamat Belajar dan Semoga Sukses " << endl;
       fileteks.close();
}
```

Menambah Data pada File Text

```
void tambahIsiFile(char nFile[], char isi[]){
       ofstream fileteks;
       fileteks.open(nFile, ios::app);
       fileteks << endl;
       fileteks << "Oleh: Akmal"<<  endl;
       fileteks << isi << endl;

       fileteks.close();
}
```

Memeriksa Keberhasilan Operasi File

```
void periksaFile(char nFile[]){
      ifstream fileteks;      // u/ membaca file
      fileteks.open(nFile);
      if (fileteks.fail())
         cout<< "File tak dapat dibuka / tidak ditemukan";
      else
         cout << "File ditemukan " << endl;

      fileteks.close();
}
```

Operasi Berbasis Karakter

Operasi file dapat dilakukan dalam bentuk karakter. Misalnya proses penyimpanan data ke file dilakukan setiap karakter, atau membaca data file karakter per karakter. Operasi ini didukung oleh function **put() dan get().**

```
void isiFilePerKarakter(char nFile[]){
      ofstream fileteks;
      fileteks.open(nFile);
      fileteks.put('A');
      fileteks.put('B');
      fileteks.put('C');
      fileteks.close();
}
```

Membaca file per karakter sekaligus menampilkan semua isi file

```
void cetakIsiFilePerKarakter(char nFile[]){
      char karakter;
      ifstream fileteks;
      fileteks.open(nFile);
      while(!fileteks.eof())
      {
            fileteks.get(karakter);
            cout << karakter;
      }
      fileteks.close();
}
```

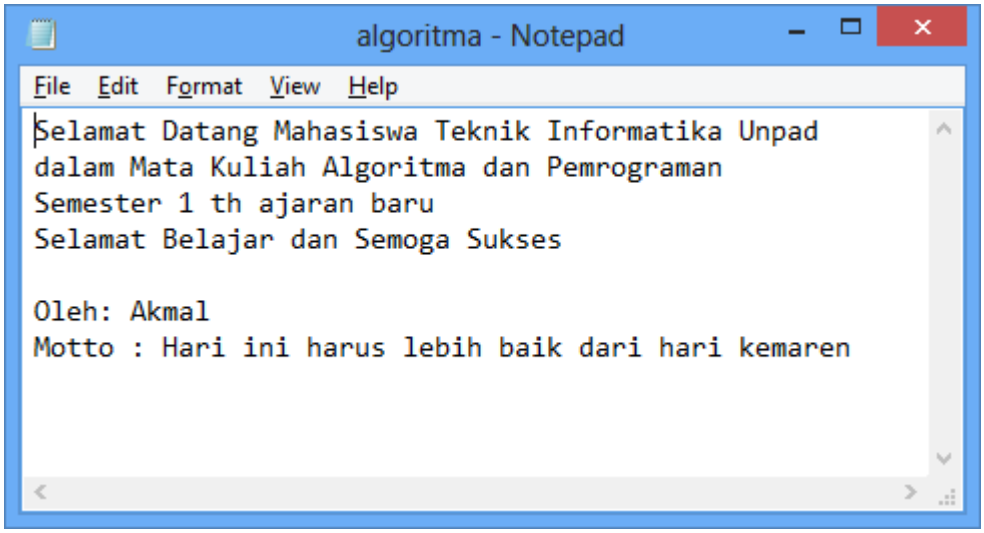
Menghitung banyak huruf 'A' yang terdapat dalam file text.

```
int cariJumlahHurufA(char nFile[]){
      char karakter;
      int jumlahA=0;
      ifstream fileteks;
      fileteks.open(nFile);
      while(!fileteks.eof())  {
            fileteks.get(karakter);
            if (karakter == 'A' || karakter == 'a')
                  jumlahA=jumlahA+1;
      }
      fileteks.close();
      return jumlahA;
}
```

Program Utama

```
int main(){
      char namaFile[]="algoritma.txt";
      char isi[]="Motto : Hari ini harus lebih baik dari
                  hari kemaren ";
      bukaTulisIsiFile(namaFile);
      tambahIsiFile(namaFile,isi);
      periksaFile(namaFile);
//    isiFilePerKarakter(namaFile);
      cetakIsiFilePerKarakter(namaFile);
      cout << "Jumlah huruf A = " <<
               cariJumlahHurufA(namaFile);
}
```

Isi file “algoritma.txt” yang dilihat melalui Notepad adalah sebagai berikut:



TUGAS 8

1. Buatlah program untuk menghitung jumlah karakter vocal dan konsonan dalam suatu file. Inputnya adalah nama file text dan pathnya.
2. Buatlah program C++ untuk menghitung jumlah karakter tertentu, misalnya karakter ‘A’. Input berupa nama file dan karakter yang akan dihitung.
3. Misalkan suatu file teks berisi listing program C++. Buatlah program untuk menghitung pasangan kurung kurawal yang ada pada file teks tersebut.
4. Buatlah program C++ untuk melakukan enkripsi shift chiper suatu file teks (dengan asumsi semua karakter huruf adalah huruf kapital). Inputnya adalah file teks yang akan dienkripsi dan besar pergeseran (integer). Outputnya adalah file teks hasil enkripsi.

   Hint:

   Ide dasar shift chiper adalah mengubah setiap karakter huruf ke karakter huruf lain. Misalkan pergeserannya 2, maka berikut ini karakter hasil enkripsi :

   awal A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

   hasil C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

   Sehingga misal diberikan suatu teks

   C++ IS EASY,

   maka hasil enkripsinya adalah

   E++ KU GCUA