

Bab 1

Seleksi Kondisi

LATIHAN 1.1

Tujuan

⊕ Seleksi kondisi dengan perintah *if.....else*.

IF-1.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 main()
05 {
06     int nilai;
07     clrscr();
08     cout << "Masukkan nilai ujian : ";
09     cin >> nilai;
10     if (nilai<=40)
11         cout << "Anda gagal";
12     else if (nilai<=60)
13         cout << "Anda harus mengulang";
14     else
15         cout << "Anda lulus";
16     getch();
17     return 0;
18 }

```

Jelaskan program di atas kemudian lakukan modifikasi dengan seleksi kondisi dimulai dari yang **lulus** (simpan dalam **IF-1M.cpp**)

LATIHAN 1.2

Tujuan

⊕ Seleksi kondisi dengan perintah *if.....else*.

IF-2 .cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 main()
05 {
06     float beli, diskon, bayar;
07     clrscr();
08     cout << "Masukkan nilai beli : ";
09     cin >> beli;
10     if (beli<=100000)

```

```

11     diskon = 0.10 * beli;
12     else if (beli<=300000)
13         diskon = 0.15 * beli;
14     else
15         diskon = 0.20 * beli;
16     bayar = beli - diskon;
17     cout << "Diskon = " << diskon << endl;
18     cout << "Bayar = " << bayar << endl;
19     getch();
20     return ;
21 }

```

Jalankan program di atas kemudian lakukan modifikasi dengan menambahkan pajak sebesar 5 % dari nilai bayar jika nilai pembelian di atas satu juta (simpan dalam IF-2M.cpp). Perhatikan tampilan berikut (yang bergaris bawah merupakan input)!

```

Masukkan nilai beli : 1500000
Diskon    = 300000
Bayar     = 120000
Pajak     = 60000
Total Bayar = 1260000

```

LATIHAN 1.3

Tujuan

⊕ Seleksi kondisi dengan perintah *switch*.

SWITCH-1.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 main()
05 {
06     int gol, status;
07     float gapok, tunjungan, total;
08     clrscr();
09     cout << "Gol : ";
10     cin >> gol;
11     cout << "Status 0:tidak_kawin 1:kawin: ";
12     cin >> status;
13
14     switch (gol)
15     {
16         case 1 : gapok = 300000; break;
17         case 2 : gapok = 500000; break;
18         case 3 : gapok = 800000;
19     }
20     if (status==1)
21         tunjungan = 0.2 * gapok;

```

```

22     else
23         tunjangan = 0;
24         total = gapok + tunjangan;
25
26         cout << "Gaji Pokok = " << gapok << endl;
27         cout << "Tunjangan = " << tunjangan << endl;
28         cout << "Total      = " << total << endl;
29
30         getch();
31         return 0;
32     }

```

Jalankan program di atas kemudian lakukan modifikasi dengan menambahkan tunjangan anak jika punya anak (simpan dalam SWITCH-1M.cpp). Aturan pemberian tunjangan anak:

Jika status =0(tidak kawin)

- ⊕ Anak ke-1: 15 % dari gaji pokok
- ⊕ Anak ke-2: 10 % dari gaji pokok
- ⊕ Anak selanjutnya tidak mendapatkan tunjangan

Jika status =1(kawin)

- ⊕ Anak ke-1: 20 % dari gaji pokok
- ⊕ Anak ke-2: 15 % dari gaji pokok
- ⊕ Anak selanjutnya masing-masing 10 % dari gaji pokok

LATIHAN 1.4

Tujuan

- ⊕ Membuat blok *if*.

IF-KUAD .cpp

```

01 // menghitung akar-akar persamaan kuadrast
02
03 #include <iostream.h>
04
05 main()
06 {
07     double a,b,c, Disk, x1,x2;
08     cout << "akar-akar persamaan kuadrat : "
09         << " Y = aX^2 + bX + c \n";
10     cout << "Input a b c : ";
11     cin >> a >> b >> c;
12     Disk = b*b - 4*a*c;
13     if (Disk<0)
14         cout << "akar-akar imaginer\n";
15     else if (Disk==0)
16     {
17         x1 = -b/(2*a);
18         cout << "akar-akar kembar\n"
19             << "x = " << x1;

```

```

20     }
21     else // Disk > 0
22     {
23         x1 = (-b - Disk) / (2*a);
24         x2 = (-b + Disk) / (2*a);
25         cout << "ada 2 akar \n"
26             << "x1 = " << x1 << "\n"
27             << "x2 = " << x2 << "\n"
28     }
29     return 0;
30 }

```

Jalankan program di atas kemudian lakukan modifikasi dengan menambahkan peringatan "bukan persamaan kuadrat" jika nilai a=0 (simpan dalam **IF-KUADM.cpp**)

TUGAS RUMAH 1

Susun program (**KUADRAN.cpp**) yang meminta masukan koordinat suatu titik dan menentukan pada kuadran mana titik tersebut berada.

Contoh tampilan:

```

Posisi X = -20
Posisi Y = 5
Titik ini berada pada Kuadran 2
Jarak dari titik pusat adalah 20.615

```

```

Posisi X = 20
Posisi Y = 55
Titik ini berada pada Kuadran 1
Jarak dari titik pusat adalah 58.52

```

TUGAS RUMAH 2

Untuk menentukan criteria kegemukan digunakan Indeks Massa Tubuh (IMT), yang bisa dihitung sebagai berikut:

$$IMT = b / t^2$$

b : bobot badan (kg)
t : tinggi badan (m)

Kriteria untuk nilai IMT ditabelkan sebagai berikut:

| Nilai IMT | Kriteria |
|----------------------|-----------|
| $IMT \leq 18.5$ | Kurus |
| $18.5 < IMT \leq 25$ | Normal |
| $25 < IMT \leq 30$ | Gemuk |
| $IMT > 30$ | Kegemukan |

Susun program (**IMT.cpp**) dengan tampilan sebagai berikut (yang bergaris bawah merupakan input):

Bobot badan (kg) : 45
Tinggi badan (m) : 1.72
IMT=15.21, termasuk kurus

Bobot badan (kg) : 85
Tinggi badan (m) : 1.71
IMT=15.21, termasuk gemuk

Bobot badan (kg) : 55
Tinggi badan (m) : 1.63
IMT=15.21, termasuk normal

TUGAS RUMAH 3

Susun program untuk merinci pecahan uang minimum dari sejumlah uang.

Contoh tampilan:

Banyaknya uang = 195800
Pecahan 100000 sebanyak 1 buah
Pecahan 50000 sebanyak 1 buah
Pecahan 20000 sebanyak 2 buah
Pecahan 5000 sebanyak 1 buah
Pecahan 500 sebanyak 1 buah
Pecahan 100 sebanyak 3 buah

Banyaknya uang = 726400
Pecahan 100000 sebanyak 7 buah
Pecahan 20000 sebanyak 1 buah
Pecahan 5000 sebanyak 1 buah
Pecahan 1000 sebanyak 1 buah Pecahan 100 sebanyak 4 buah

Catatan:

- ⊕ Pecahan yang tersedia adalah 100000, 50000, 20000, 10000, 5000, 1000, 500 dan 100 rupiah.
- ⊕ Simpan dalam **RUPIAH.cpp**.

TUGAS RUMAH 4

Susun program untuk pembayaran PDAM dengan aturan sebagai berikut:

Untuk Golongan=1 (rumah tangga)

- ⊕ Untuk 15 m³ pertama tarifnya adalah Rp 500/m³
- ⊕ Untuk 15 m³ kedua tarifnya adalah Rp 1000/m³
- ⊕ Untuk sisanya tarifnya adalah Rp 2000/m³

Untuk Golongan=2 (usaha/bisnis)

- ⊕ Untuk 15 m³ pertama tarifnya adalah Rp 1000/m³
- ⊕ Untuk 15 m³ kedua tarifnya adalah Rp 2000/m³
- ⊕ Untuk sisanya tarifnya adalah Rp 3000/m³

Contoh tampilan (yang bergaris bawah merupakan input)

```
Golongan = 1
Masukkan pemakaian : 20
Tarif pertama      : Rp  7500
Tarif kedua       : Rp  5000
Tarif sisa        : Rp    0
-----
Subtotal           : Rp 12500
Pajak (10%)       : Rp    0
-----
Total Tagihan     : Rp 12500
```

```
Golongan = 1
Masukkan pemakaian : 60
Tarif pertama      : Rp  7500
Tarif kedua       : Rp 15000
Tarif sisa        : Rp 60000
-----
Subtotal           : Rp 82500
Pajak (10%)       : Rp  8250
-----
Total Tagihan     : Rp 90750
```

```
Golongan = 2
Masukkan pemakaian : 100
Tarif pertama      : Rp 15000
Tarif kedua       : Rp 30000
Tarif sisa        : Rp 210000
-----
Subtotal           : Rp 255000
Pajak (10%)       : Rp    0
-----
Total Tagihan     : Rp 255000
```

Catatan:

- ⊕ Pajak dikenakan hanya jika **subtotal** di atas Rp 50000 untuk gol 1 dan Rp 300000 untuk gol 2
- ⊕ Simpan dalam **PDAM.cpp**

TUGAS RUMAH 5

Susun program (**lembur.cpp**) untuk masalah penggajian

Input : jam kerja/minggu

Output : total upah

Aturan :

Batas jam kerja maksimal 60 jam, dengan upah Rp 5000/jam; bila lebih dari batas ini maka dianggap lembur dengan upah Rp 6000/jam

Batas jam kerja minimal 50 jam; bila kurang dari batas ini maka dikenakan potongan Rp 1000/jam.

Contoh tampilan

Jam Kerja : **55**
Upah = Rp 275000
Lembur = Rp 0
Denda = Rp 0

Total = Rp 275000

Jam Kerja : **70**
Upah = Rp 300000
Lembur = Rp 60000
Denda = Rp 0

Total = Rp 360000

Jam Kerja : **40**
Upah = Rp 200000
Lembur = Rp 0
Denda = Rp 10000

Total = Rp 190000

Bab 2

Perulangan / Iterasi

LATIHAN 2.1

Tujuan

- ⊕ Membuat perulangan dengan perintah *for*.

FOR-1 .cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 main()
05 {
06     int i,n,total;
07     clrscr();
08     cout << "Banyaknya suhu : ";
09     cin >> n;
10
11     total = 0;
12     for (i=1; i<=n; i++)
13         total = total + i;
14     cout << "Total = " << total << endl;
15
16     getch();
17     return 0;
18 }
```

Jalankan program di atas kemudian lakukan modifikasi sehingga menghasilkan tampilan sebagai berikut (simpan dalam **FOR-1M.cpp**) .

```

Banyaknya suhu : 4
1+2+3+4=10
```

LATIHAN 2.2

Tujuan

- ⊕ Membuat perulangan bersarang (*nested loop*)

FOR-2 .cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 main()
05 {
06     int i,j,n;
07     clrscr();
08     cout << "N = ";
```



```

09     cin >> n;
10
11     for (i=1; i<=n; i++)
12     {
13         For (j=1; j<=n; j++)
14             If (j<i) cout << " ";
15             else cout << j;
16         cout << endl;
17     }
18     getch();
19     return 0;
20 }

```

Jalankan program di atas kemudian lakukan modifikasi sehingga menghasilkan tampilan sebagai berikut (simpan dalam **FOR-2M.cpp**) :

```

N = 4
1 * * * * *
  2 2 * * *
    3 3 3 * *
      4 4 4 4

```

LATIHAN 2.3

Tujuan

⊕ Membuat perulangan dengan perintah *do.....while*.

FOR-1 .cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 main()
05 {
06     int n;
07     float x, total, rata;
08     clrscr();
09
10     total = 0;
11     n = 0;
12     do
13     {
14         n++;
15         cout << "Data ke-" << n << " : ";
16         cin >> x;
17         total = total + x;
18     } while (x!=0);
19
20     n--;
21     rata = total / n;

```

```

22     cout << "Ada " << n << " data " << endl;
23     cout << "Total = " << total << endl;
24     cout << "Rata-rata = " << rata;
25
26     getch();
27     return 0;
28 }

```

Jalankan program di atas kemudian lakukan modifikasi untuk mencari maksimum dan minimum dari data yang dimasukkan (simpan dalam **while-1M.cpp**)

TUGAS RUMAH 1

Susun program untuk menghitung angsuran hutang dengan tampilan sebagai berikut (simpan dalam **HUTANG-1.cpp**)

Jumlah Hutang = 1000000
Masa Angsuran (bulan) = 10
% suku Bunga per bulan = 2

| Bula | Sisa Hutang | Bunga | Pokok | Total angsuran |
|------|-------------|-------|--------|----------------|
| 1 | 1000000 | 20000 | 100000 | 120000 |
| 2 | 900000 | 18000 | 100000 | 118000 |
| 3 | 800000 | 16000 | 100000 | 116000 |
| 4 | 700000 | 14000 | 100000 | 114000 |
| 5 | 600000 | 12000 | 100000 | 112000 |
| 6 | 500000 | 10000 | 100000 | 110000 |
| 7 | 400000 | 8000 | 100000 | 108000 |
| 8 | 300000 | 6000 | 100000 | 106000 |
| 9 | 200000 | 4000 | 100000 | 104000 |
| 10 | 100000 | 2000 | 100000 | 102000 |

Catatan

⊕ Bunga dihitung dari sisa hutang

TUGAS RUMAH 2

Modifikasi **HUTANG-1.cpp** menjadi Angsuran Tetap dengan tampilan sebagai berikut (simpan dalam **HUTANG-2.cpp**):

Jumlah Hutang = 1000000
Angsuran Tetap = 80000
% suku Bunga per bulan = 2

| Bula | Sisa Hutang | Bunga | Angsuran | Sisa Hutang |
|------|-------------|-------|----------|-------------|
| 1 | 1000000 | 20000 | 80000 | 940000 |
| 2 | 940000 | 18800 | 80000 | 878800 |
| 3 | 878800 | 17576 | 80000 | 816376 |
| 4 | 816376 | 16328 | 80000 | 752704 |

| | | | | |
|----|--------|-------|-------|--------|
| 5 | 752704 | 15054 | 80000 | 687758 |
| 6 | 687758 | 13755 | 80000 | 621513 |
| 7 | 621513 | 12430 | 80000 | 553943 |
| 8 | 553843 | 11079 | 80000 | 485022 |
| 9 | 485022 | 9700 | 80000 | 414722 |
| 10 | 414722 | 8294 | 80000 | 343017 |
| 11 | 343017 | 6860 | 80000 | 269877 |
| 12 | 269877 | 5398 | 80000 | 195275 |
| 13 | 195275 | 3905 | 80000 | 119180 |
| 14 | 119180 | 2384 | 80000 | 41564 |
| 15 | 41564 | 831 | 42395 | 0 |

Lunas setealah 15 bulan

TUGAS RUMAH 3

Sejenis bakteri akan berkembang biak menjadi dua kali lipat tiap harinya, tetapi setiap 3 hari kumpulan bakteri tersebut musnah 80 persen. Susun program (**BAKTERI.cpp**) dengan input jumlah bakteri sekarang dan waktu perkembangbiakan (perhatikan tampilan program di bawah ini).

Banyaknya bakteri sekarang: 20

Waktu (hari) : 7

| Hari ke | Jumlah |
|---------|---------------------------|
| 1 | 20 |
| 2 | 40 |
| 3 | 80 Musnah 80% tinggal 16 |
| 4 | 32 |
| 5 | 64 |
| 6 | 128 Musnah 80% tinggal 25 |
| 7 | 50 |

Banyaknya bakteri setelah hari ke-7 adalah : 50

Banyaknya bakteri sekarang: 12

Waktu (hari) : 9

| Hari ke | Jumlah |
|---------|--------------------------|
| 1 | 12 |
| 2 | 24 |
| 3 | 48 Musnah 80% tinggal 8 |
| 4 | 16 |
| 5 | 32 |
| 6 | 64 Musnah 80% tinggal 12 |

7 24
8 48
9 96 Musnah 80% tinggal 19
Banyaknya bakteri setelah hari ke-9 adalah : 19

TUGAS RUMAH 4

Susun program (**FOR-FOR.cpp**) untuk menampilkan sebagai berikut :

Masukan banyaknya baris = 5

```
1  
2 4  
3 6 9  
4 8 12 16  
5 10 15 20 25
```

Bab
 3

Procedure
LATIHAN 3.1

Tujuan

 ⊕ Membuat program modular menggunakan *procedure*
PLUAS .cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 void LuasBidang();
05 void LuasLingkaran();
06 main()
07 {
08     Char pilih;
09     Do
10     {
11         clrscr();
12         cout << "MENU" << endl;
13         cout << "1. Luas Bidang" << endl;
14         cout << "2. Luas dan Keliling Lingkaran"
15             << endl;
16         cout << "0. Keluar" << endl;
17         cout << "Pilihan Anda : ";
18         cin >> pilih;
19
20         switch (pilih)
21         {
22             Case '1' : LuasBidang();
23                         break;
24             Case '2' : LuasLingkaran();
25                         break;
26         }
27     } while (pilih!='0');
28     return 0;
29 }
30
31 void LuasBidang();
32     return 0;
33 {
34     float panjang, lebar, luas;
35     cout << "\nMenghitung Luas Bidang \n";
36     cout << "Masukkan Panjang = ";
37     cin >> panjang;

```

```

38     cout << "Masukkan Lebar = ";
39     cin >> lebar;
40     luas = panjang * lebar;
41     cout << "Luas bidang = " << luas;
42     getch();
43 }
44
45 void LuasLingkaran()
46 {
47     float r, luas, keliling;
48     cout << "\nMenghitung Luas dan "
49           << "Keliling Lingkaran\n";
50     cout << "Masukkan Jari-Jari = ";
51     cin >> r;
52     luas = 3.14 * r * r;
53     keliling = 2 * 3.14 * r;
54     cout << "Luas lingkaran = " << luas << endl
55           << "Keliling lingkaran = " << keliling;
56     getch();
57 }

```

Jalankan program di atas kemudian lakukan modifikasi sebagai berikut (simpan dalam **PUASM.cpp**) :

⊕ Tambah procedure **Menu** sehingga program utama berubah sebagai berikut

```

...
do
{
    Menu (pilih);
    Switch (pilih)
    {
        case '1' : LuasBidang();
                Break;
        case '2' : LuasLingkaran();
                Break;
        case '3' : VolumeBalok();
                Break;
        case '4' : VolumeSilinder();
                Break;
        case '3' : VolumeBola();
                Break;
    }
} while (pilih!='0');

```

- ⊕ Procedure Menu harus bisa memberikan peringatan jika user memilih pilihan selain 0,1,2,3,4,5.
- ⊕ Tambahkan tiga procedure untuk menghitung volume balok, silinder dan bola.

LATIHAN 3.2

Tujuan

- ⊕ Menyusun procedure yang berparameter.

PLUAS .cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 void InputABC(double &a, double &b, double &c);
05 void HitungAkar(double a, double b, double c);
06
07 main()
08 {
09     double a,b,c;
10     char Lagi='Ya';
11     while ( (lagi=='Ya') || (Lagi=='y') )
12     {
13         clrscr();
14         cout << "Akar-Akar persamaan kuadrat : "
15             << "Y = aX^2 + bX + c \n";
16         InputABC (a,b,c);
17         HitungAkar (a,b,c);
18         cout << "\nHitung Lagi (Y/T) : ";
19         cin >> Lagi;
20     } // end while
21     cout << "\nTerima kasih telah menggunakan program
22 ini";
23 }
24
25 void InputABC (double &a, double &b, double &c)
26 {
27     do
28     {
29         cout << "Input a b c : ";
30         cin >> a >> b >> c;
31     } while (a==0);
32 }
33
34 void HitungAkar(double a, double b, double c)
35 {
36     double Disk, x1,x2;
37     Disk = b*b - 4*a*c;
38     if (Disk<0)
39         cout << "akar akar imaginer\n";
40     else if (Disk==0)
41     {
42         x1 = -b/(2*a);

```

```

43     cout << "akar-akar kembar\n";
44         << "x = " << x1;
45     }
46     else // Disk > 0
47     {
48         x1 = (-b - Disk) / (2*a);
49         x2 = (-b + Disk) / (2*a);
50         cout << "ada 2 akar \n"
51             << "x1 = " << x1 << "\n"
52             << "x2 = " << x2 << "\n";
53     }
54 }

```

Jalankan program di atas kemudian lakukan modifikasi dengan menampilkan akar-akar imajiner (simpan dalam **PKUADM.cpp**)

LATIHAN 3.3

Tujuan

- ⊕ Menyusun procedure yang berparameter.

BOX.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03 #include <dos.h>
04
05 void Box (int x1, int y1, int x2, int y2);
06
07 main()
08 {
09     clrscr();
10     Box (4,2,76,20);
11     getch();
12     return 0;
13 }
14
15 void Box (int x1, int y1, int x2, int y2)
16 {
17     int x,y;
18     for (x=x1; x<=x2; x++)
19     {
20         gotoxy (x,y1);
21         cout << "*";
22         delay(10);
23     }
24     for (y=y1; y<=y2; y++)
25     {
26         gotoxy (x2,y);
27         cout << "*";

```



```

28     delay(10);
29     }
30     for (x=x2; x>=x1; x++)
31     {
32         gotoxy (x,y2);
33         cout << "*";
34         delay(10);
35     }
36     for (y=y2; y>=y1; y++)
37     {
38         gotoxy (x1,y);
39         cout << "*";
40         delay(10);
41     }
42 }

```

Jalankan program di atas kemudian lakukan modifikasi sebagai berikut (simpan dalam **BOX.cpp**):

- ⊕ Tambahkan procedure **GoOut** sehingga procedure **Box** berubah sebagai berikut:

```

void Box (int x1, int y1, int x2, int y2)
{
    int x,y;
    for (x=x1; x<=x2; x++)
        GoOut (x,y1, '*');
    for (y=y1; y<=y2; y++)
        GoOut (x2,y, '*');
    for (x=x2; x>=x1; x++)
        GoOut (x,y2, '*');
    for (y=y2; y>=y1; y++)
        GoOut (x1,y, '*');
}

```

- ⊕ Modifikasi procedure **Box** sehingga arah pecetakan '*' berlawanan jarum jam

LATIHAN 3.4

Tujuan

- ⊕ Menggunakan procedure yang berparameter.

KPK.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 void CariKpk (int a, int b, int &k);
05
06 main()

```

```

07 {
08     clrscr();
09     int bil1,bil2,kpk;
10     cout << "Masukkan bilangan pertama : ";
11     cin >> bil1;
12     cout << "Masukkan bilangan kedua : ";
13     cin >> bil2;
14
15     cariKpk (bil1, bil2, kpk);
16     cout << "KPK dari dua bilangan tersebut adalah ="
17         << kpk;
18     getch();
19     return 0;
20 }
21
22 // mencari kpk dengan syarat a<=b
23 void CariKpk (int a, int b, int &k)
24 {
25     int selesai=0;
26     k = b;
27     while (!selesai)
28     {
29         if ((k%a==0)&&(k%b==0))
30             selesai = 1;
31         else
32             k++;
33     }
34 }

```

Jalankan program di atas kemudian lakukan modifikasi dengan menambahkan procedure untuk mencari faktor persekutuan terbesar/FPB (simpan dalam **KPK.cpp**).

Procedure **CariKpk** dan **CariFpb** bekerja dengan syarat **a<=b**, tambahkan procedure **Swap** untuk menukarkan isi dua variable tersebut jika syarat tidak terpenuhi, sehingga isi program utama akan bertambah sebagai berikut:

```

If (bil1>bil2)
    Swap (bil1,bil2);

```

Bab
4

Function

LATIHAN 4.1

Tujuan

⊕ Membuat program modular menggunakan *function*..**FOR-1.cpp**

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 int Maximum (int a, int b);
05
06 main()
07 {
08     int x,y,max;
09     clrscr();
10     cout << "x = ";
11     cin >> x;
12     cout << "y = ";
13     cin >> y;
14     max = Maximum(x,y);
15     cout << "Yang terbesar adalah " << max;
16     getch();
17 }
18
19 int Maximum (int a, int b)
20 {
21     if (a>b)
22         return a;
23     else
24         return b;
25 }

```

Jalankan program di atas kemudian lakukan modifikasi function **Maximum** untuk mencari nilai terbesar di antara tiga bilangan (simpan dalam **FMAXM.cpp**) . Perhatikan isi program utama sebagai berikut:

```

int x,y,z,max;
cout << "x = "; cin >> x;
cout << "y = "; cin >> y;
cout << "z = "; cin >> z;
max = Maximum (x,y,z);
cout << "Yang terbesar adalah " << max;

```

LATIHAN 4.2

Tujuan

- ⊕ Membuat program modular menggunakan *function*.

FTOTAL.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 int HitungTotal (int n);
05
06 main()
07 {
08     int n, total;
09     clrscr();
10     cout << "Banyaknya suku : ";
11     cin >> n;
12     total = HitungTotal (n);
13     cout << "Total = " << total;
14     getch();
15 }
16
17 int HitungTotal (int n)
18 {
19     int tot = 0;
20     for (int i=1; i<=n; i++)
21         tot = tot + i;
22     return total;
23 }

```

Jalankan program di atas kemudian lakukan modifikasi function **HitungTotal** untuk mencari total n bilangan ganjil pertama (simpan dalam **FTOTALM.cpp**). Perhatikan isi program sebagai berikut:

```

Bayaknya suku : 4
Total = 10

```

LATIHAN 4.3

Modifikasi **PLUASM.cpp** (**Latihan 3.1**) sehingga procedure **menu** berubah menjadi function dan program utamanya sebagai berikut (simpan dalam **FLUAS.cpp**) :

```

...
do
{
    Pilih = Menu ();
    Switch (pilih)
    {
        case '1' : LuasBidang();
                Break;

```

```
        case '2' : LuasLingkaran();
                Break;
        case '3' : VolumeBalok();
                Break;
        case '4' : VolumeSilinder();
                Break;
        case '5' : VolumeBola();
                Break;
    }
} while (pilih='0');
```

LATIHAN 4.4

Modifikasi **KPKM.cpp** (**Latihan 3.4**) sehingga program utamanya sebagai berikut (simpan dalam **FKPK.cpp**) *

```
...
cout << "Masukkan bilangan pertama : ";
cin >> bil1;
cout << "Masukkan bilangan kedua : ";
cin >> bil2;

if (bil1>bil2)
    swap(bil1,bil2)

kpk = cari Kpk (bil1, bil2);
fpb = cari Fpb (bil1, bil2);

cout << "KPK dari dua bilangan tersebut adalah = "
    << kpk;
cout << "FPB dari dua bilangan tersebut adalah = "
    << fpb;
```

LATIHAN 4.5

Modifikasi **SWITCH-1M.cpp** (dalam **Latihan 1.3**) hingga program utamanya berubah sebagai berikut (simpan dalam **Gaji.cpp**) :

```
...
Gapok = HitungGajiPokok (gol);
Tunjangan = HitungTunjangan (Status);
Tunjangan_anak = HitungTunjanganAnak (status, anak);
...
```

TUGAS RUMAH 1

Susun function untuk memeriksa apakah suatu bilangan adalah prima. Isi program utamanya adalah sebagai berikut (simpan dalam **PRIMA.cpp**)

```

int IsPrime (int x);

main()
{
    clrscr();
    int I;
    cout << "Bilangan prima antara 1 sampai 100 \n";
    for (i=2; i<=100; i++)
        if (IsPrime(i))
            cout << I << " ";
    getch();
    return 0;
}

int IsPrime (int x)
{
    ...
    ...
}

```

TUGAS RUMAH 2

Bilangan amisabel adalah bilangan yang jumlah faktor-faktornya sama dengan bilangan tersebut.

Contoh:

$$\begin{aligned}
 1 &= 1 \\
 6 &= 1 + 2 + 3 \\
 28 &= 1 + 2 + 4 + 7 + 14
 \end{aligned}$$

Susun function untuk memeriksa apakah suatu bilangan adalah amisabel, kemudian cetak semua bilangan antara 1 sampai 10000 (simpan dalam **AMSB.cpp**).

TUGAS RUMAH 3

Nilai kombinasi ${}_n C_m$ didefinisikan secara iterative sebagai berikut:

$${}_n C_m = n! / [(n-m)! m!]$$

Secara rekursif bias didefinisikan:

$$\begin{aligned}
 {}_n C_1 &= n \\
 {}_n C_n &= 1 \\
 {}_n C_m &= {}_{n-1} C_{m-1} + {}_{n-1} C_m \text{ untuk } n > m > 1
 \end{aligned}$$

Susun fungsi rekursif untuk menghitung nilai kombinasi di atas dengan header:

```
Long int Comb (int n, int m);
```

Susun program (**COMB.cpp**) dengan tampilan sebagai berikut:

Nilai $n = \underline{5}$
Nilai $m = \underline{2}$
Nilai 5 Kombinasi 2 = 10

Nilai $n = \underline{6}$
Nilai $m = \underline{2}$
Nilai 6 Kombinasi 2 = 15

Bab 5

Array dan String

LATIHAN 5.1

Tujuan

- ⊕ Membuat program menggunakan array.

ARRAY1.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03 #include <math.h>
04
05 main()
06 {
07     unsigned int i,n;
08     float *nilai, avg=0, sd=0;
09
10     clrscr();
11     cout << "Masukkan banyaknya data = ";
12     cin >> n;
13
14     // pesan memori
15     nilai = new float [n];
16
17     // input data, indek mulai 0 !
18     for (i=0; i<n; i++)
19     {
20         cout <<"data ke-" << i+1 << " = ";
21         cin >> nilai[i];
22     }
23
24     // hitung rata-rata / average
25     for (i=0; i<n; i++)
26         avg+=nilai[i];
27     avg/=n;
28
29     // hitung Standard Deviasi
30     for (i=0; i<n; i++)
31         sd+=pow(nilai[i]-avg,2);
32     sd = sqrt (sd/(n-1));
33
34     cout.precision (4);
35     cout << "*** Average = " << avg << endl;
36     cout << "*** Standard Deviasi = " << sd;
37

```



```

38     // bebaskan memory
39     delete [] nilai;
40     getch();
41     return 0;
42 }

```

Jalankan program di atas kemudian lakukan modifikasi dengan menambahkan pencarian nilai maksimum dan minimum (simpan dalam **ARRAY1M.cpp**) .

LATIHAN 5.2

Tujuan

⊕ Melewatkan array sebagai parameter fungsi.

ARRAY1.cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03 #include <math.h>
04
05 typedef float *FLOAT;
06 void InputData (FLOAT &nilai, int &n);
07 void GetAvgSd (FLOAT nilai, int n, float &avg,
08              float &sd);
09 float GetMax (FLOAT nilai, int n);
10 float GetMin (FLOAT nilai, int n);
11 int GetNumUpperAvg (FLOAT nilai, int n, float avg);
12
13 main ()
14 {
15     int i,n,nUp,nLo;
16     FLOAT nilai;
17     float avg=0, sd=0;
18
19     clrscr();
20     InputData (nilai, n);
21     GetAvgSd (nilai, n, avg, sd);
22     nUp = GetNumUpperAvg (nilai, n, avg);
23     nLo = n - nUp;
24
25     cout.precision (4);
26     cout << "* Average = " << avg << endl;
27     cout << "* Standard Deviasi = " << sd << endl;
28     cout << "* Maximum = "
29           << GetMax(nilai,n) << endl;
30     cout << "* Minimum = "
31           << GetMin(nilai,n) << endl;
32     cout << "* # nilai di atas average = "
33           << nUp << endl;
34     cout << "* # nilai di bawah average = "

```

```
35         << nLo << endl;
36
37     delete [] nilai;
38     getch();
39     return 0;
40 }
41
42 void InputData (FLOAT Snilai, int &n)
43 {
44     cout << "Masukkan banyaknya data = ";
45     cin >> n;
46     nilai = new float [n] ;
47     for (int i=0; i<n; i++)
48     {
49         cout <<"data ke-" << i+1 << " = ";
50         cin >> nilai [i];
51     }
52 }
53
54 void GetAvgSd (FLOAT nilai, int n, float &avg,
55               float &sd)
56 {
57     int i;
58     avg = 0;
59     for (i=0; i<n; i++)
60         avg+=nilai [i];
61     avg/=n;
62
63     sd = 0;
64     for (i=0; i<n; i++)
65         sd+=pow(nilai[i]-avg,2);
66     sd = sqrt (sd/(n-1));
67 }
68
69 float GetMax (FLOAT nilai, int n)
70 {
71     ...
72 }
73
74 float GetMin (FLOAT nilai, int n)
75 {
76     ...
77 }
78
79 int GetNumUpperAvg (FLOAT nilai, int n, float avg)
80 {
81     int num=0;
```

```

82     for (int i=0; i<n; i++)
83         if (nilai[i]>avg)
84             num++;
85     return num;
86 }

```

Jalankan program di atas dengan melengkapi function **GetMax** dan **GetMin**.

LATIHAN 5.3

Tujuan

- Memanfaatkan string untuk penyandian.

SANDI . cpp

```

01 #include <iostream.h>
02 #include <conio.h>
03
04 char *Encrypt (char *dest, const char *src);
05
06 main()
07 {
08     char *asal;
09     char *sandi="";
10     clrscr();
11     cout << "Masukkan kalimat : ";
12     cin.getline (asal,20);
13     Encrypt (sandi, asal);
14     cout << "Hasil penyandian : " << sandi;
15     getch() ;
16     return 0;
17 }
18 char *Encrypt (char *dest, const char *src)
19 {
20     dest = new char[n]
21     for (int i=0; src[i]!='\0'; i++)
22         dest[i] = src[i] + 1;
23     dest[i] = src[i]; // copy NULL char
24     return dest;
25 }

```

Jalankan program di atas kemudian lakukan modifikasi (simpan dalam **SANDIM.cpp**):

- ⊕ Tambahkan function untuk penguraian / *Decrypt*.
- ⊕ Modifikasi tampilan program dengan sistem menu sebagai berikut:
 - Menu Program
 - 1.Penyandian Kalimat
 - 2.Penguraian Kalimat
 - 0. Keluar
 - Pilihan Anda : _

TUGAS RUMAH 1

Susun program (**STRING1.cpp**) dengan tampilan sebagai berikut:

Masukkan string : UNISMA

```
UNISMA
UNISM
UNIS
UNI
UN
U
```

Masukkan string : TEKNIK ELEKTRO

```
TEKNIK ELEKTRO
TEKNIK ELEKTR
TEKNIK ELEKT
TEKNIK ELEK
TEKNIK ELE
TEKNIK EL
TEKNIK E
TEKNIK
TEKN
TEK
TE
T
```

TUGAS RUMAH 2

Modifikasi tugas di atas dan simpan dalam **STRING2.cpp** dengan tampilan sebagai berikut::

Masukkan string : UNISMA

```
UNISMA
 NISMA
  ISMA
   SMA
    MA
     A
```

TUGAS RUMAH 3

Modifikasi **ARRAY1.cpp** (**Latihan 5.2**) dengan menggunakan sistem menu sebagai berikut (simpan dalam **ARMENU.cpp**):

```
Menu Program
1. Input Data
2. Tampilkan Data
3. Distribusi Frekuensi
4. Statistik
5. Urutkan Data
```

Menu Tampilkan Data adalah menampilkan data dalam bentuk aslinya sebagai berikut:

```
Ada 5 data
Data ke-1:80
```

Data ke-2 : 65
Data ke-3 : 70
Data ke-4 : 90
Data ke-5 : 50

Menu **Distribusi Frekuensi** adalah menampilkan data dalam bentuk sebagai berikut (jika data yang dimasukkan cukup banyak):

Ada 34 data

| Range | Frekuensi |
|--------|-----------|
| 0-20 | 5 |
| 21-40 | 8 |
| 41-60 | 9 |
| 61-80 | 10 |
| 81-100 | 2 |

Menu **Statistik** adalah menampilkan rata-rata, standard deviasi, maksimum dan minimum.

Menu **Urutkan Data** digunakan untuk mengurutkan data dari kecil ke besar (*ascending*).

Bab 6

File Sekuensial

Operasi File

Open :

| | | | | |
|------------|---|----------|---|--|
| 1. reset | } | ofstream | → | open file yang sudah ada, jika belum ada langsung create |
| 2. rewrite | | ofstream | | fin(.....) |
| 3. append | | ofstream | | filekeluaran(.....) |
| | | ifstream | | → open file |
| | | ios.app | | → append / menambah |

if(fin.fail()) → gagal dibuka → nilai balik ≠ 0 jika ada salah

if(fin.good()) → bisa → nilai balik = 0 kalau ada salah

if(fin.bad()) → error → nilai balik ≠ 0 kalau ada salah

if(fin.eof()) → akhir → nilai balik = 0 kalau ada salah

LATIHAN 6.1

Tujuan

⊕ Membuat program untuk menampilkan isi file teks.

TEXT.cpp

```

01 #include <fstream.h>
02 #include <conio.h>
03 #include <ctype.h>
04
05 char *UpperString (const char *str);
06
07 main()
08 {
09     char *str;
10     char namaFile[20];
11     clrscr();
12     cout << "Masukkan nama file : ";
13     cin >> getline (namafile,20);
14
15     ifstream fin(namafile);
16     if (!fin)
17     {
18         cout << "File tidak bias dibuka";
19         return 1;
20     }
21
22     int i=0;
23     while (!fin.eof())
24 
```

```

25     {
26         fin.getline (str,80);
27         str = UpperString (str);
28         i++;
29         cout << i << ": " << str << endl;
30     }
31     fin.close();
32     getch();
33     return 0;
34 }
35
36 char *UpperString (const char *str)
37 {
38     char *s="";
39     ...
40     return s;
41 }

```

Lengkapi fungsi **UpperString** yang digunakan untuk mengubah isi string ke huruf besar kemudian jalankan program di atas.

Lakukan modifikasi sehingga bisa menampilkan banyaknya huruf, angka dan karakter lain (sapasi, koma, titik dan lain-lain) dari file yang dibuka/ditampilkan (simpan dalam **TEXTIM.cpp**).

LATIHAN 6.2

Tujuan

⊕ Membuat program untuk membaca data pada file text.

```

TEXT2.cpp
01 #include <iomanip.h>
02 #include <fstream.h>
03 #include <conio.h>
04 #include <ctype.h>
05 #include <stdlib.h>
06
07 main()
08 {
09     char nim[6], nama[11];
10     char c_tugas[4], c_uts[4], cuas[4];
11     int tugas, uts, uas;
12
13     ifstream fin("mhs.dat");
14     if (!fin)
15     {
16         cout << "File tidak bias dibuka";
17         return 1;
18     }
19

```

```

20 clrscr();
21 int i=0;
22 cout << "  nim   nama           tgs uts uas\n";
23 while (!fin.eof())
24 {
25     fin.get (nim,6);
26     fin.get (nama,11);
27     fin.get (c_tugas,4);
28     fin.get (c_uts,4);
29     fin.getline (c_uas,5);
30
31     tugas = atoi (c_tugas);
32     uts = atoi (c_uts);
33     uas = atoi (c_uas);
34     i++;
35     cout << i << ". " << nim << nama
36         << setw(4) << tugas
37         << setw(4) << uts
38         << setw(4) << uas << endl;
39 }
40 fin.close();
41 getch();
42 return 0;
43 }

```

Jalankan program di atas dengan terlebih dahulu membuat file **Mhs.dat** dengan contoh isi sebagai berikut (hati-hati penentuan spasi):

```

1234567890123456789012345 ←posisi kolom (jangan disimpan)
01001 abah           45 65 89
01002 bibi           80 45 70
01003 cucu           60 80 30
01004 dinda          70 80 70

```

Data yang disimpan di atas adalah nim, nama, nilai tugas, nilai uts dan nilai uas.

Modifikasi program di atas dengan menambah nilai akhir dengan tampilan sebagai berikut (simpan dalam **TEXT2M.cpp**) :

| Nim | Nama | Tugas | UTS | UAS | Nilai Akhir | Huruf |
|-------|-------|-------|-----|-----|-------------|-------|
| 01001 | abah | 45 | 65 | 89 | 67 | C+ |
| 01002 | bibi | 80 | 45 | 70 | 73 | B |
| 01003 | cucu | 60 | 80 | 30 | 75 | B+ |
| 01004 | dinda | 70 | 80 | 70 | 78 | B+ |

Bobot nilai untuk perhitungan nilai akhir serta batas nilai untuk konversi ke nilai huruf tentukan sendiri

TUGAS RUMAH

Buat program untuk menyediakan isi file teks dan hasilnya disimpan di file lain (simpan dalam **SANDIF.cpp**).

Contoh tampilan

```
Menu program
1. Penyandian Kalimat
2. Penguraian Kalimat
0. Keluar
Pilihan Anda : 1
```

```
Nama file sumber : surat.txt
Nama file tujuan : surat.snd
```

Proses penyandian file telah dilakukan, tekan ENTER

Catatan: `atoi()` → fungsi untuk mengkonversi string yang ditunjuk `str` menjadi bilangan bertipe `int`. String yang dikonversi memiliki format sbb:

```
[sp][td][ddd]
[sp] = tab/spasi
[td] = tanda +/-
[ddd] = jumlah digit
```

Contoh:

```
atoi("+2.1E+02") → 2
```

```
atoi("+2") → 2
```

```
atoi(" 20.1") → 2
```

```
atoi("CDE") → 2
```

| | |
|-----------------|--------------------|
| Bab 7 | File Random |
|-----------------|--------------------|

LATIHAN 7.1

Tujuan

- ⊕ Menggunakan tipe data *struct* dalam pengolahan file random.

IF-1.cpp

```

01 #include <iomanip.h>
02 #include <fstream.h>
03 #include <conio.h>
04 #include <ctype.h>
05 #include <stdlib.h>
06
07 Struct TRecMhs {
08     char nim[11], nama[11];
09     int tugas, uts, uas;
10 };
11
12 void Menu (char &pilih);
13 void TambahData (const char *NamaFile);
14 void TampilData (char *Namafile);
15
16 main()
17 {
18     char NamaFile[]="mhs-rnd.dat";
19     char pilih;
20     do
21     {
22         Menu (pilih);
23         Switch (pilih)
24         {
25             case '1' : TambahData (NamaFile);
26                 break;
27             case '2' : TampilData (NamaFile);
28                 break;
29         }
30     } while (pilih!='0');
31     return 0;
32 }
33
34 void Menu (char &pilih)
35 {
36     clrscr();
37     cout << "MENU" << endl;

```

```

38     cout << "1. Tambah Data" << endl;
39     cout << "2. Lihat Data" << endl;
40     cout << "0. Keluar" << endl;
41     cout << "Pilihan Anda : ";
42     cin >> pilih;
43 }
44
45 void TambahData (const char *NamaFile)
46 {
47     TRecMhs mhs;
48     cout << "\nTambah Data \n";
49     cout << "Nim    : "; cin >> mhs.nim;
50     cout << "Nama   : "; cin >> mhs.nama;
51     cout << "Nilai" << endl;
52     cout << "Tugas  : "; cin >> mhs.tugas;
53     cout << "UTS    : "; cin >> mhs.uts;
54     cout << "UAS    : "; cin >> mhs.uas;
55     fstream fl (NamaFile, ios::in|ios::out);
56     fl.seekp (0, ios::end);
57     fl.write ((char*)&mhs, sizeof(mhs));
58     fl.close();
59 }
60
61 void TampilData (char *NamaFile)
62 {
63     TRecMhs mhs;
64     ifstream fl (NamaFile, ios::in|ios::out);
65     int i=0;
66     while (!fl.eof())
67     {
68         fl.read ((char*)&mhs, sizeof(mhs));
69         if (!fl.eof())
70         {
71             i++;
72             cout << i
73                 << setw(6) << mhs.nim
74                 << setw(10) << mhs.nama
75                 << setw(4) << mhs.tugas
76                 << setw(4) << mhs.uts
77                 << setw(4) << mhs.uas << endl;
78         }
79     }
80     fl.close();
81     getch();
82 }

```

Jalankan program di atas kemudian lakukan modifikasi pada procedure **TampilData** sehingga tampilannya seperti pada **TEXT2.cpp** (**Latihan 6.2**) dengan tambahan pada bagian bawah sebagai berikut (simpan dalam **RANDOM1M.cpp**) :

```
Nim      Nama      Tugas    UTS      UAS      Nilai Akhir  Huruf
-----
01001   abah      45       65       89       67          C+
01002   bibi      80       45       70       73          B
...
...
```

```
Rata-rata nilai Tugas = 60.43
Rata-rata nilai UTS   = 72.67
Rata-rata nilai UAS   = 65.41
Rata-rata nilai akhir = 61.89
```

Distribusi Frekuensi Nilai Huruf

```
A      : 5 mahasiswa
B+     : 7 mahasiswa
B      : 9 mahasiswa
C+     : 16 mahasiswa
C      : 11 mahasiswa
D+     : 5 mahasiswa
D      : 3 mahasiswa
E      : 2 mahasiswa
```

TUGAS RUMAH

Tambah menu program dari **RANDOM1.cpp** sehingga menjadi sebagai berikut (simpan dalam **MENURND.cpp**) .

```
Menu
1. Tambah Data
2. Lihat Data
3. Edit Data
4. Hapus Data
5. Cari berdasarkan nim
6. Cari berdasarkan nama
0. Keluar
```

Tampilan menu **Edit Data** sebagai berikut:

```
Edit Data
Nomer record yang akan diedit :3

Data Lama
Nim   : 01006
Nama  : Farhan
Nilai
```

Tugas : 80
UTS : 70
UAS : 75

Data Baru

Nim : 01006

Nama : Fulan

Nilai

Tugas : 90

UTS : 70

UAS : 75

Tampilan menu **Hapus Data** sebagai berikut:

Hapus Data

Nomer record yang akan dihapus : 4

Record telah dihapus

Tampilan menu **Cari berdasarkan nim** sebagai berikut:

Cari NIM

Masukkan nim : 01006

Data ditemukan

Nim : 01006

Nama : Fulan

Nilai

Tugas : 90

UTS : 70

UAS : 75

Tampilan menu **Cari berdasarkan nama** setara dengan **Cari berdasarkan nim**.

Cari NAMA

Masukkan nama : Rusli

Data ditemukan

Nim : 010098

Nama : Rusli

Nilai

Tugas : 60

UTS : 40

UAS : 55

| | |
|-----------------|----------------|
| Bab 8 | Pointer |
|-----------------|----------------|

LATIHAN 8.1

Tujuan

⊕ Seleksi kondisi dengan perintah *if.....else*.**PTR1.cpp**

```

01 #include <iomanip.h>
02 #include <fstream.h>
03 #include <conio.h>
04 #include <stdlib.h>
05
06 struct TRecData {
07     char item;
08     TRecData *next ;
09 };
10
11 typedef TRecData *TPtr;
12
13 void Menu (char Spilih)
14 {
15     clrscr() ;
16     cout << "MENU" << endl;
17     cout << "1. Tambah Data" << endl;
18     cout << "2. Lihat Data" << endl; •
19     cout << "0. Keluar" << endl;
20     cout << "Pilihan Anda : ";
21     cin >> pilih;
22 }
23
24 void TambahData (TPtr &p)
25 {
26     char c;
27     cout << "\nTambah Data \n";
28     cout << "Item    : ";
29     cin >> c;
30
31     TPtr temp = new(TRecData);
32     temp->item = c;
33     temp->next = p;      ~
34     p = temp;
35 }
36
37 void TampilData (TPtr p)

```

```
38 {
39     TPtr temp = p;
40     int i=0;
41     cout <<"\nIsi Linked List\n";
42     while (temp)
43     {
44         i++;
45         cout << i << ". " << temp->item << endl;
46         temp = temp->next;
47     }
48     getch ();
49 }
50
51 main()
52 {
53     TPtr head=0;
54     char pilih;
55     do
56     {
57         Menu (pilih);
58         switch (pilih)
59         {
60             case '1' : TambahData (head);
61                 break;
62             case '2' : TampilData (head);
63                 break;
64         }
65     } while (pilih!='0');
66     return 0;
67 }
```

Jalankan program di atas kemudian lakukan modifikasi dengan menambahkan menu pencarian dan penghapusan data (simpan dalam **PTR1M.cpp**):

Bab 9

Studi Kasus

9.1. EVALUATOR FUNGSI POLINOMIAL

Konsep Dasar

Sebuah polinomial berderajat sembarang dapat direpresentasikan sebagai suatu urutan pasangan-pasangan *koefisien-pangkat* yang disusun secara menurun berdasarkan urutan pangkatnya. Sebagai contoh polinomial :

$$x^7 + 5x^4 + 3x^3 - 6x^2 + x + 5$$

dapat direpresentasikan sebagai :

$$(1, 7), (5, 4), (3, 3), (-6, 2), (1, 1), (5, 0)$$

Urutan pasangan-pasangan *koefisien-pangkat* di atas dapat disimpan dalam suatu *list* yang diimplementasikan dengan variabel dinamis (*linked-list*).

Struktur Program (POLI . cpp)

- ⊕ Variabel global A,B,C yang bertipe list variabel dinamis untuk menyimpan data polinomial
- ⊕ Menu program
 1. Input data polinomial melalui keyboard
 - Input A
 - Input B
 2. Load data polinomial dari disk (nama file diinputkan)
 - Load A
 - Load B
 3. Save data polinomial ke disk (nama file diinputkan)
 - Save A
 - Save B
 - Save C
 4. Operasi matematika dan cetak hasil ke layar
 - Operasi Penjumlahan (C = A+B)
 - Operasi Pengurangan (C = A-B)
 - Operasi Perkalian (C = A*B)
 5. Perhitungan nilai fungsi polinomial
 - A(x); x diinputkan
 - B(x); x diinputkan
 - C(x); x diinputkan

9.2. KLASEMEN KOMPETISI SEPAKBOLA

Tujuan Program

Program ini digunakan sebagai alat bantu untuk mengurutkan klasemen tim sepakbola yang berlaga dalam Kompetisi Sepakbola Indonesia 2015

Input Program

Input program diletakkan dalam dua file text:

File 1 : TANDING.DAT

```
6                               { banyaknya pertandingan }
01/10/15  1  2  3  4  { tanggal } { tim 1 } { tim2 } { gol tim 1 } { gol tim 2 }
01/10/15  3  4  1  2
05/10/15  1  5  2  1
05/10/15  2  3  3  2
08/10/15  3  5  3  3
12/10/15  2  4  2  0
```

File 1 : TIM.DAT

```
5                               { banyaknya tim }
AREMA                               { nama tim -> string[15] }
PERSEMA
PSIS
PERSIJA
PSM
```

Output Program

Dari dua file input di atas dihasilkan suatu array yang berisi penilaian hasil pertandingan :

| Nama tim | main | menang | seri | kalah | golM | golK | nilai |
|--------------|------|--------|------|-------|------|------|-------|
| 1.AREMA | 2 | 1 | 0 | 1 | 5 | - 5 | 3 |
| 2.PETROKIMIA | 3 | 3 | 0 | 0 | 9 | - 5 | 9 |
| 3.PSIS | 3 | 0 | 1 | 2 | 6 | - 8 | 1 |
| 4.PERSIJA | 2 | 1 | 0 | 1 | 2 | - 3 | 3 |
| 5.PSM | 2 | 0 | 1 | 1 | 4 | - 5 | 1 |

golM : gol memasukkan

golK : gol kemasukan

Dari data di atas akan didapatkan urutan klasemen sementara. Aturan pengurutan klasemen adalah sebagai berikut:

- ⊕ Tim dengan nilai yang lebih tinggi akan berada pada peringkat yang lebih tinggi (setiap kemenangan diberi nilai 3, seri nilai 1, kalah nilai 0)
- ⊕ Bila ada tim yang nilainya sama, maka yang berada pada peringkat lebih tinggi adalah yang selisih golnya (**memasukkan_gol** - **kemasukan**) lebih baik.

- ⊕ Bila ada tim yang nilai dan selisih golnya sama, maka yang berada pada peringkat yang lebih tinggi adalah tim yang lebih banyak menang.

Dari data di atas dihasilkan output urutan klasemen sementara (di layar dan dalam *text file*) sebagai berikut:

| No | Tim | M | M | S | K | GOL | Nilai |
|----|------------|---|---|---|---|-------|-------|
| 1. | PETROKIMIA | 3 | 3 | 0 | 0 | 9 - 5 | 9 |
| 2. | AREMA | 2 | 1 | 0 | 1 | 5 - 5 | 3 |
| 3. | PERSEMA | 2 | 1 | 0 | 1 | 2 - 3 | 3 |
| 4. | PSM | 2 | 0 | 1 | 1 | 4 - 5 | 1 |
| 5. | PSIS | 3 | 0 | 1 | 2 | 6 - 8 | 1 |

Dari output di atas bisa dilihat:

- ⊕ **PETROKIMIA** menempati puncak klasemen dengan nilai 9
- ⊕ **AREMA** berada di atas **PERSEMA** karena meskipun nilainya sama (3) tetapi selisih gol AREMA (5-5=0) lebih baik daripada PERSEMA (2-3=-1).

9.3. APLIKASI PENJUALAN BARANG

Sebuah perusahaan distributor menyimpan data penjualan barang dalam sebuah file teks (**JUAL.DAT**) dengan struktur sebagai berikut:

- Tanggal, 8 digit dengan format YYYYMMDD (YYYY:tahun, MM:bulan, DD:tanggal)
- Kode salesman, 2 digit
- Kode barang, 4 digit
- Banyak barang, 4 digit.

Perusahaan juga menyimpan daftar harga barang dalam file relatif (**BARANG.DAT**) dengan struktur:

- Kode barang, 4 digit, bernilai antara **0000** sampai **9999** sehingga bisa digunakan sebagai *key* (nomer record) dalam file random
- Nama barang, 10 digit
- Harga barang, 6 digit

Misalkan file **JUAL.DAT** berisi data sebagai berikut:

```
20030101 01 1012 0010
20030101 01 1002 0005
20030101 02 0082 0005
20030101 03 0005 0015
20030102 01 0011 0010
20030102 02 0015 0010
20030102 02 0211 0020
```

File sequensial di atas sudah terurut menurut tanggal dan kode sales. Dengan membaca file yang telah diurutkan di atas, susun program laporan penjualan sebagai berikut:

LAPORAN PENJUALAN

| TANGGAL | SALES | NAMA. BARANG | HARGA | BANYAK | TOTAL |
|------------|-----------|--------------|-------|--------|-------|
| 01-01-2000 | 01 | Pensil ABC | 1000 | 10 | 10000 |
| | | Sabun X | 1500 | 5 | 7000 |
| | 02 | Pen Y | 2000 | 5 | 10000 |
| | 03 | Pen Z | 3000 | 15 | 45000 |
| | Sub total | | | | 72000 |
| 02-01-2000 | 01 | Sabun A | 1500 | 10 | 15000 |
| | 02 | Sampho C | 2500 | 10 | 25000 |
| | | Pasta Gigi A | 500 | 20 | 10000 |
| | Sub total | | | | 50000 |
| Total | | | | 122000 | |

Catatan : **Total** didapatkan dengan mengalikan **banyak barang** dengan harga pada file **BARANG.DAT**.

LAMPIRAN

Lampiran 1. Tipe Data, Ukuran dan Jangkauan

| Data Type | Size (bits) | Range |
|---------------|-------------|--|
| unsigned char | 8 | 0 to 255 |
| char | 8 | -128 to 127 |
| enum | 16 | -32,768 to 32,767 |
| unsigned int | 16 | 0 to 65,535 |
| short int | 16 | -32,768 to 32,767 |
| Int | 16 | -32,768 to 32,767 |
| unsigned long | 32 | 0 to 4,294,967,295 |
| long | 32 | -2,147,483,648 to 2,147,483,647 |
| float | 32 | 3.4×10^{-38} to 3.4×10^{38} |
| double | 64 | 1.7×10^{-308} to 1.7×10^{308} |
| long double | 80 | 3.4×10^{-4932} to 1.1×10^{-4932} |
| near pointer | 16 | - |
| far pointer | 32 | - |

Note:

- . : decimal separator
- , : thousand separator

Lampiran 2. C++ Keywords

Ansi C

| | | | |
|----------|--------|----------|----------|
| Auto | else | register | union . |
| break | enum | return | unsigned |
| case | extern | short | void |
| char | float | signed | volatile |
| const | for | sizeof | while |
| continue | goto | static | |
| default | if | struct | |
| do | int | switch | |
| double | long | typedef | |

Tambahan pada C++

| | |
|----------|-----------|
| asm | private |
| catch | protected |
| class | public |
| delete | template |
| friend | this |
| inline | throw |
| new | try |
| operator | virtual |
| overload | |

Lampiran 3. Operator Aritmatika

| Tipe Integer | | Tipe Floating Point | |
|--------------|-------------|---------------------|-------------|
| + | penjumlahan | + | penjumlahan |
| - | pengurangan | - | pengurangan |
| * | perkalian | * | perkalian |
| / | pembagian | / | pembagian |
| % | sisanya | | |

Lampiran 4. Operator Logika

| Operator | Artinya |
|----------|---------|
| ! | not |
| && | and |
| | or |

Lampiran 5. Operator Relasional

| Operator | Arti |
|--------------------|------------------------------|
| <code>==</code> | Sama dengan |
| <code>!=</code> | Tidak sama dengan |
| <code><</code> | Kurang dari |
| <code><=</code> | Kurang dari atau sama dengan |
| <code>></code> | Lebih dari |
| <code>>=</code> | Lebih dari atau sama dengan |

Lampiran 6. Kombinasi Assignment

| Penyingkatan | Arti |
|---------------------|------------------------|
| <code>x++</code> | <code>x = x + 1</code> |
| <code>x--</code> | <code>x = x - 1</code> |
| <code>x += y</code> | <code>x = x + y</code> |
| <code>x -= y</code> | <code>x = x - y</code> |
| <code>x *= y</code> | <code>x = x * y</code> |
| <code>x /= y</code> | <code>x = x / y</code> |
| <code>x %= y</code> | <code>x = x % y</code> |