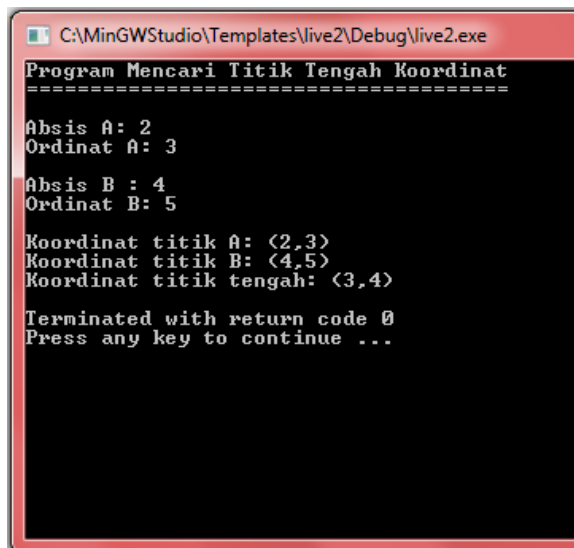


TUGAS ALPROG RECORD & ARRAY OF RECORD

Nama	: Patricia Joanne
NPM	: 140810160065

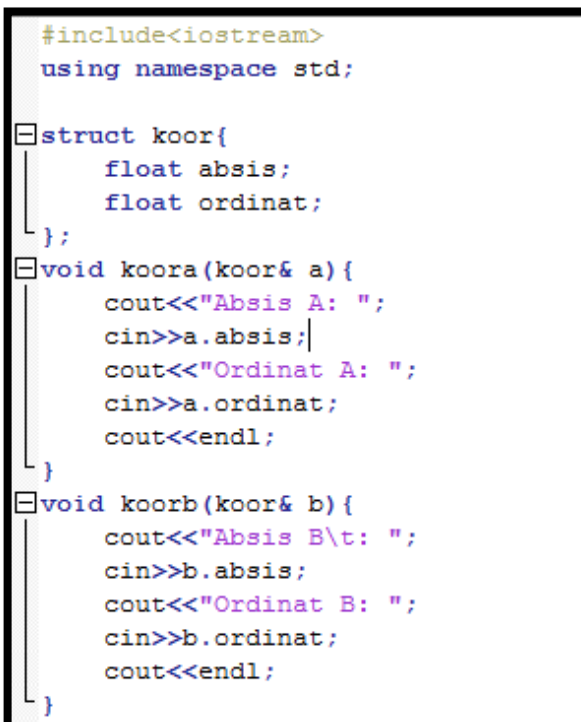
1. Mencari titik tengah koordinat A dan B

// OUTPUT //



```
C:\MinGWStudio\Templates\live2\Debug\live2.exe
Program Mencari Titik Tengah Koordinat
=====
Absis A: 2
Ordinat A: 3
Absis B : 4
Ordinat B: 5
Koordinat titik A: <2,3>
Koordinat titik B: <4,5>
Koordinat titik tengah: <3,4>
Terminated with return code 0
Press any key to continue ...
```

// CODINGAN & PENJELASAN //



```
#include<iostream>
using namespace std;

struct koor{
    float absis;
    float ordinat;
};

void koora(koor& a){
    cout<<"Absis A: ";
    cin>>a.absis;
    cout<<"Ordinat A: ";
    cin>>a.ordinat;
    cout<<endl;
}

void koorb(koor& b){
    cout<<"Absis B\t: ";
    cin>>b.absis;
    cout<<"Ordinat B: ";
    cin>>b.ordinat;
    cout<<endl;
}
```

- Header file iostream
- Struct yang gunanya untuk membuat tipe data bentukan dimana setiap elemennya bisa memiliki tipe data yang berbeda, pada kasus ini struct koor memiliki 2 elemen yaitu absis dan ordinat dalam tipe float
- Prosedur koora untuk menginput absis dan ordinat titik A
- Prosedut koorb untuk menginput absis dan ordinat titik B

```

void titiktengah(koor& a, koor& b, koor& t) {
    t.absis = (a.absis + b.absis) / 2;
    t.ordinat = (a.ordinat + b.ordinat) / 2;
}

void cetak(koor& a, koor& b, koor& t) {
    cout << "Koordinat titik A: (" << a.absis << ", " << a.ordinat << ") " << endl;
    cout << "Koordinat titik B: (" << b.absis << ", " << b.ordinat << ") " << endl;
    cout << "Koordinat titik tengah: (" << t.absis << ", " << t.ordinat << ") ";
}

```

- Prosedut titiktengah untuk melakukan perhitungan titik tengah
- Prosedur cetak untuk menampilkan koodinat titik A, titik B, dan titik tengah

```

int main() {
    koor a, b, t;

    cout << "Program Mencari Titik Tengah Koordinat" << endl;
    cout << "===== " << endl << endl;

    //input koor A
    koora(a);

    //input koor B
    koorb(b);

    //titik tengah
    titiktengah(a, b, t);

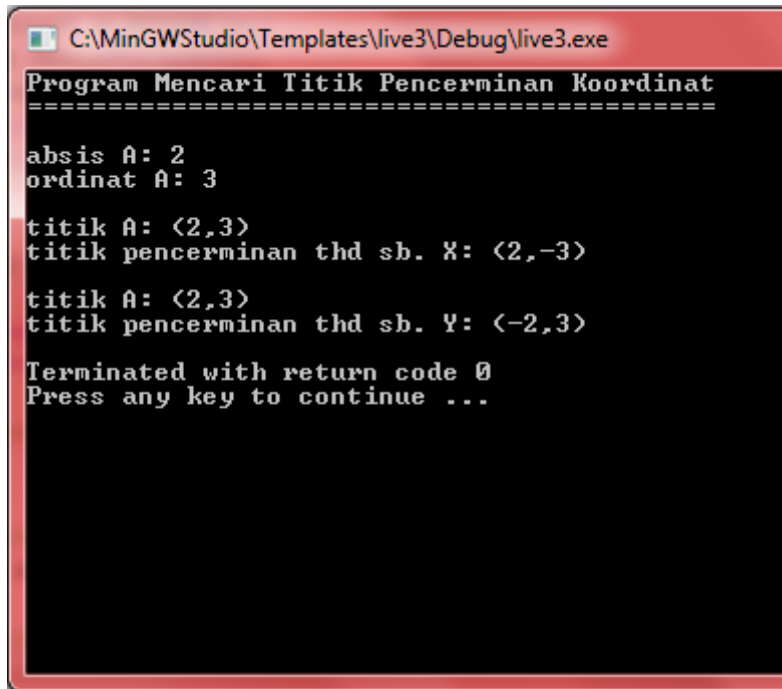
    //cetak koor
    cetak(a, b, t);
}

```

- Fungsi main tempat pemanggilan semua fungsi dan prosedur

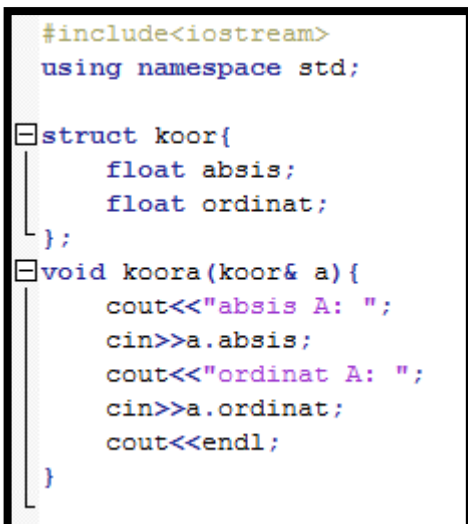
2. Mencari titik pencerminan sebuah koordinat

// OUTPUT //



```
C:\MinGWStudio\Templates\live3\Debug\live3.exe
Program Mencari Titik Pencerminan Koordinat
=====
absis A: 2
ordinat A: 3
titik A: <2,3>
titik pencerminan thd sb. X: <2,-3>
titik A: <2,3>
titik pencerminan thd sb. Y: <-2,3>
Terminated with return code 0
Press any key to continue ...
```

// CODINGAN & PENJELASAN //



```
#include<iostream>
using namespace std;

struct koor{
    float absis;
    float ordinat;
};

void koora(koor& a){
    cout<<"absis A: ";
    cin>>a.absis;
    cout<<"ordinat A: ";
    cin>>a.ordinat;
    cout<<endl;
}
```

- Header file iostream
- Struct yang gunanya untuk membuat tipe data bentukan dimana setiap elemennya bisa memiliki tipe data yang berbeda, pada kasus ini struct koor memiliki 2 elemen yaitu absis dan ordinat dalam tipe float
- Prosedur koora untuk menginput absis dan ordinat titik A

```

void cerminx(koor& a, koor& x) {
    x.abis=a.abis;
    x.ordinat=(a.ordinat*(-1));
    cout<<"titik A: ("<<a.abis<<","<<a.ordinat<<")"<<endl;
    cout<<"titik pencerminan thd sb. X: ("<<x.abis<<","<<x.ordinat<<")"<<endl<<endl;
}

void cerminy(koor& a, koor& y) {
    y.abis=(a.abis*(-1));
    y.ordinat=a.ordinat;
    cout<<"titik A: ("<<a.abis<<","<<a.ordinat<<")"<<endl;
    cout<<"titik pencerminan thd sb. Y: ("<<y.abis<<","<<y.ordinat<<")";
}

```

- Prosedur cerminx untuk melakukan perhitungan pencerminan terhadap sumbu X, yaitu negatif dari ordinatnya lalu ditampilkan
- Prosedur cerminy untuk melakukan perhitungan pencerminan terhadap sumbu Y, yaitu negatif dari absisnya lalu ditampilkan

```

int main() {
    koor a,x,y;

    cout<<"Program Mencari Titik Pencerminan Koordinat"<<endl;
    cout<<"=====|"<<endl<<endl;

    //input koor A
    koora(a);

    //pencerminan sb. X
    cerminx(a,x);

    //pencerminan sb. Y
    cerminy(a,y);
}

```

- Fungsi main tempat pemanggilan semua fungsi dan prosedur

3. Mencari lama parkir dan bayar parkir

// OUTPUT //

```
C:\MinGWStudio\Templates\live1\Debug\live1.exe
Program Penghitung Biaya Parkir di Jatos
=====
Masukkan jam masuk: 2
Masukkan menit masuk: 23
Masukkan detik masuk: 12
Anda masuk pada jam 2 lewat 23 menit 12 detik
=====
Masukkan jam keluar: 3
Masukkan menit keluar: 23
Masukkan detik keluar: 13
Anda keluar pada jam 3 lewat 23 menit 13 detik
=====
Anda telah parkir selama 3601 detik
Atau selama 1 jam 0 menit 1 detik
=====
Biaya parkir yang harus dibayar: 6000
=====
Terminated with return code 0
Press any key to continue ...
```

// CODINGAN & PENJELASAN //

```
#include <iostream>
using namespace std;

struct waktu{
    int jam,men,det;
};

void inputmasuk(waktu& msk){
    cout<<"Masukkan jam masuk: ";
    cin>>msk.jam;
    cout<<"Masukkan menit masuk: ";
    cin>>msk.men;
    cout<<"Masukkan detik masuk: ";
    cin>>msk.det;
    cout<<"Anda masuk pada "<<"jam "<<msk.jam<<" lewat "<<msk.men<<" menit "<<msk.det<<" detik";
    cout<<endl<<endl<<"===== "<<endl<<endl;
}

void inputkeluar(waktu& kel){
    cout<<"Masukkan jam keluar: ";
    cin>>kel.jam;
    cout<<"Masukkan menit keluar: ";
    cin>>kel.men;
    cout<<"Masukkan detik keluar: ";
    cin>>kel.det;
    cout<<"Anda keluar pada "<<"jam "<<kel.jam<<" lewat "<<kel.men<<" menit "<<kel.det<<" detik";
    cout<<endl<<endl<<"===== "<<endl<<endl;
}
```

- Header file iostream
- Struct yang gunanya untuk membuat tipe data bentukan dimana setiap elemennya bisa memiliki tipe data yang berbeda, pada kasus ini struct koor memiliki 3 elemen yaitu jam, menit, dan detik dalam tipe int
- Prosedur inputmasuk untuk menginput jam masuk, menit masuk, dan detik masuk lalu menampilkannya
- Prosedur inputkeluar untuk menginput jam keluar, menit keluar, dan detik keluar lalu menampilkannya

```

void lamadet(waktu& msk,waktu& kel,int& tot){
    int out=kel.jam*3600+kel.men*60+kel.det;
    int in=msk.jam*3600+msk.men*60+msk.det;
    tot=out-in;
    cout<<"Anda telah parkir selama ";
    cout<<tot<<" detik";
}

void lamaparkir(waktu& wkt,int& tot){
    wkt.jam=tot/3600;
    int sisj=tot%3600;
    wkt.men=sisj/60;
    wkt.det=sisj%60;

    cout<<endl<<"Atau selama "<<wkt.jam<<" jam "<<wkt.men<<" menit "<<wkt.det<<" detik";
    cout<<endl<<endl<<"===== "<<endl<<endl;
}

void bayar(waktu& wkt,int& byr){
    if (wkt.men>=1||wkt.det>=1) byr=3000*(wkt.jam+1);
    else byr=3000*wkt.jam;

    cout<<"Biaya parkir yang harus dibayar: "<<byr;
    cout<<endl<<endl<<"===== ";
}

```

- Prosedur lamadet untuk menghitung lama parkir dalam detik dan menampilkannya
- Prosedur lamaparkir untuk menghitung lama parkir dalam waktu jam, menit, dan detik dan menampilkannya
- Prosedur untuk menghitung biaya parkir

```

int main() {
    waktu msk, kel, wkt;
    int tot, byr;
    cout<<"Program Penghitung Biaya Parkir di Jatots" << endl;
    cout<<endl<<"===== " << endl<< endl;

    //input masuk
    inputmasuk(msk);

    //input keluar
    inputkeluar(kel);

    //lama parkir dalam detik
    lamadet(msk, kel, tot);

    //lama parkir dalam jam menit detik
    lamaparkir(wkt, tot);

    //bayar parkir
    bayar(wkt, byr);
}

```

- Fungsi main tempat pemanggilan semua fungsi dan prosedur

4. Gaji pegawai

// OUTPUT //

```

C:\MinGWStudio\Templates\tmm\Debug\tmm.exe
DAFTAR GAJI PEGAWAI PT. INGIN SUKSES TAHUN 2016
=====
Gol  Gaji
=====
1    2jt
2    3jt
3    5jt
4    8jt

Banyak pegawai: 2
Data pegawai ke-1
NIP   : 2000
Nama  : Patris
Gol   : 2

Data pegawai ke-2
NIP   : 4000
Nama  : Joanne
Gol   : 1

NIP    Nama      Gol    Gaji
=====
2000   Patris     2      Rp3000000
4000   Joanne     1      Rp2000000

Rata-rata: 2500000
Gaji terbesar: 3000000
Gaji terkecil: 2000000

Terminated with return code 0

```

// CODINGAN & PENJELASAN //

```
#include<iostream>
#include<iomanip>
using namespace std;

struct pegawai{
    int nip;
    char nama[20]; //nama pegawai maks 20 karakter
    int gaji;
    int gol;
};

typedef pegawai array[20]; //banyak pegawai maks 20
```

- Header file iostream
- Struct yang gunanya untuk membuat tipe data bentukan dimana setiap elemennya bisa memiliki tipe data yang berbeda, pada kasus ini struct koor memiliki 4 elemen yaitu nip, gaji, dan gol dalam tipe int dan nama dalam tipe char.
- Typedef yang gunanya untuk membuat alias larik pegawai menjadi array[20]

```
void carigaji(int& n,array& peg){
    for(int i=0;i<n;i++){
        switch(peg[i].gol){
            case 1: peg[i].gaji = 2000000;
                    break;
            case 2: peg[i].gaji = 3000000;
                    break;
            case 3: peg[i].gaji = 5000000;
                    break;
            case 4: peg[i].gaji = 8000000;
                    break;
        }
    }
}
```

- Prosedur carigaji untuk mencari gaji berdasarkan golongannya


```

void input(int& n,array& peg){
    cout<<"Banyak pegawai: ";
    cin>>n;
    cout<<endl;

    for(int i=0;i<n;i++){
        cout<<"Data pegawai ke-"<<i+1<<endl;
        cout<<"NIP\t: ";
        cin>>peg[i].nip;
        cout<<"Nama\t: ";
        cin.ignore();
        cin.getline(peg[i].nama, 20);
        cout<<"Gol\t: ";
        cin>>peg[i].gol;
        cout<<endl;
    }
}

```

- Prosedur input untuk menginput banyak pegawai dan data pegawai yang terdiri dari NIP, nama, dan golongannya

```

void daftar(){
    cout<<" GOL " <<setw(5)<<" GAJI " <<endl;
    cout<<"===== " <<endl;
    cout<<" 1 " <<setw(5)<<" 2jt " <<endl;
    cout<<" 2 " <<setw(5)<<" 3jt " <<endl;
    cout<<" 3 " <<setw(5)<<" 5jt " <<endl;
    cout<<" 4 " <<setw(5)<<" 8jt " <<endl<<endl;
}

```

- Prosedur daftar untuk menampilkan daftar gaji berdasarkan golongannya

```

void sorting(int& n,array& peg){
    carigaji(n,peg);
    for(int i=n-1;i>0;i--){
        for(int j=0;j<i;j++){
            if(peg[j].nip>peg[j+1].nip){
                swap(peg[j],peg[j+1]);
            }
        }
    }
    cout<<"NIP\tNama\t\tGol\tGaji" <<endl;
    cout<<"===== " <<endl;
    for(int j=0;j<n;j++){
        cout<<peg[j].nip<<"\t"<<peg[j].nama<<"\t\t"<<peg[j].gol<<"\t"<<"Rp"<<peg[j].gaji<<endl;
    }
}

```

- Prosedur sorting untuk mengurutkan data pegawai berdasarkan NIPnya

```

void ratagaji(int& n,array& peg){
    int sum=0;
    int rata;
    for(int i=0;i<n;i++){
        sum=sum+peg[i].gaji;
    }
    rata=sum/n;
    cout<<endl<<"Rata-rata: "<<rata;
}

void max(int& n,array& peg){
    int max=0;
    for (int i=0;i<n;i++){
        if(max<peg[i].gaji) max=peg[i].gaji;
    }
    cout<<endl<<"Gaji terbesar: "<<max;
}

void min(int& n,array& peg){
    int min;
    for(int i=0;i<n;i++){
        if(min>peg[i].gaji) min=peg[i].gaji;
    }
    cout<<endl<<"Gaji terkecil: "<<min;
}

```

- Prosedur ratagaji untuk mencari rata-rata dimulai dari mencari dulu total semua gaji lalu dirata-ratakan sejumlah pegawai dan menampilkannya
- Prosedur max untuk mencari gaji terbesar dan menampilkannya
- Prosedur min untuk mencari gaji terkecil dan menampilkannya

```

int main(){
    int n;
    array peg;

    cout<<"DAFTAR GAJI PEGAWAI PT. INGIN SUKSES TAHUN 2016"<<endl;
    cout<<"===== "<<endl<<endl;

    //cetakdaftar
    daftar();

    //input pegawai
    input(n,peg);

    //sorting
    sorting(n,peg);

    //ratagaji
    ratagaji(n,peg);

    //gajitertinggi
    max(n,peg);

    //gajiterendah
    min(n,peg);
}

```

- Fungsi main tempat pemanggilan semua fungsi dan prosedur