INHERITANCE

Oleh:
Rasim
ILKOM-FPMIPA-UPI

Selayang oop

- Konsep-konsep yang berkaitan erat dengan pemrograman berorientasi objek adalah:
  - objek,
  - kelas,
  - pewarisan (inheritance),
  - polymorphism,
  - dynamic binding

DEFINISI

- pewarisan adalah
  - konsep yang merupakan ciri unik dari model pemrograman berorientasi objek
- Tanpa inheritance, pemanfaatan bahasa C++ hanyalah sekedar abstract data type programming, **bukan object-oriented programming**
- Konsep pewarisan memungkinkan perancang kelas untuk mendenisikan dan mengimplementasikan sebuah kelas berdasarkan kelas-kelas yang sudah ada.

- Jika sebuah kelas A mewarisi kelas lain B, maka A merupakan kelas turunan (derived class/ subclass) dan B merupakan kelas dasar (base class/superclass)
- Seluruh anggota (data & fungsi) di B akan berada juga di kelas A, kecuali constructor, destructor.
- Akibat dari pewarisan, kelas A akan memiliki dua bagian: bagian yang diturunkan dari B dan bagian yang didenisikan sendiri oleh kelas A dan bersifat spesik terhadap A.
- Dalam konteks ini, objek B yang muncul di dalam A dapat dipandang sebagai sub-objek dari kelas B

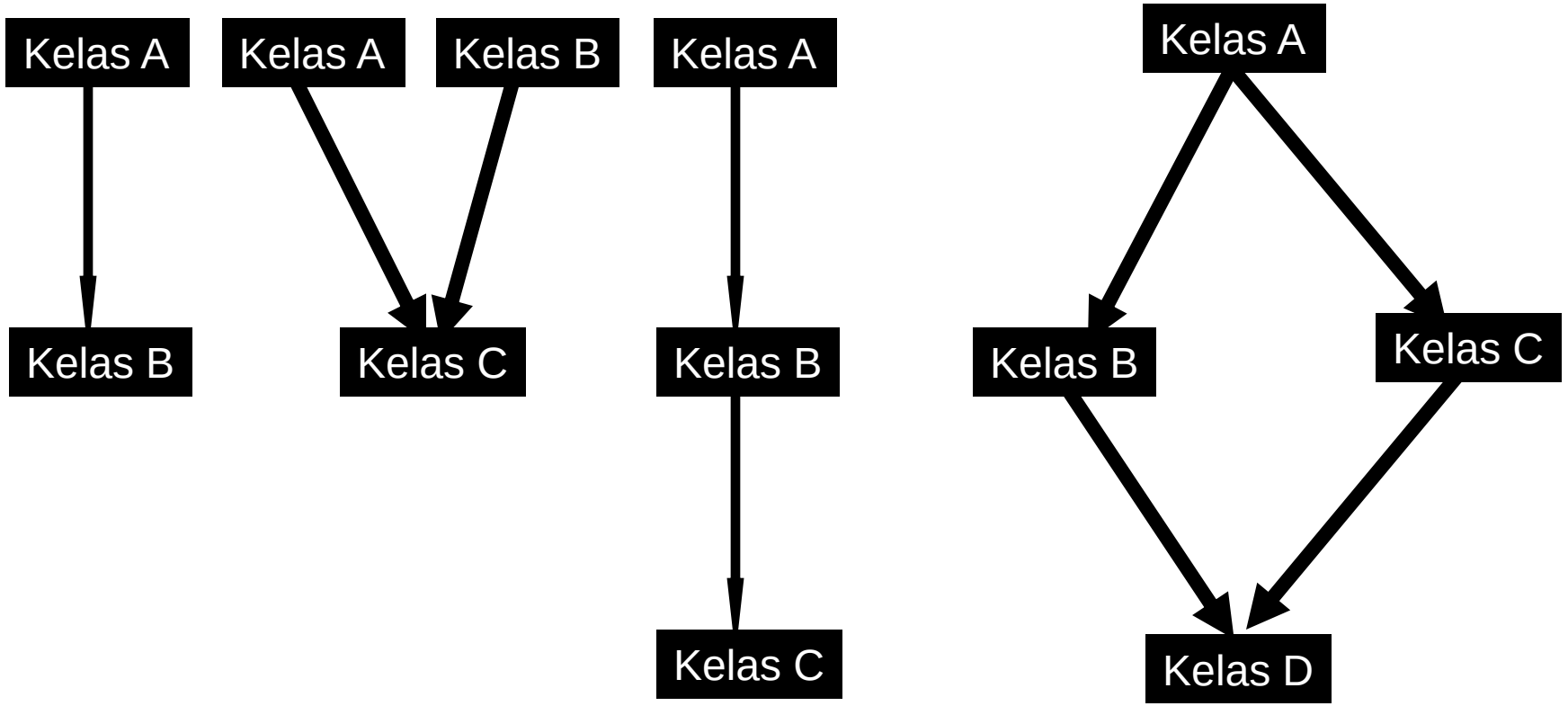
- Penurunan kelas dituliskan dalam C++ sebagai berikut:

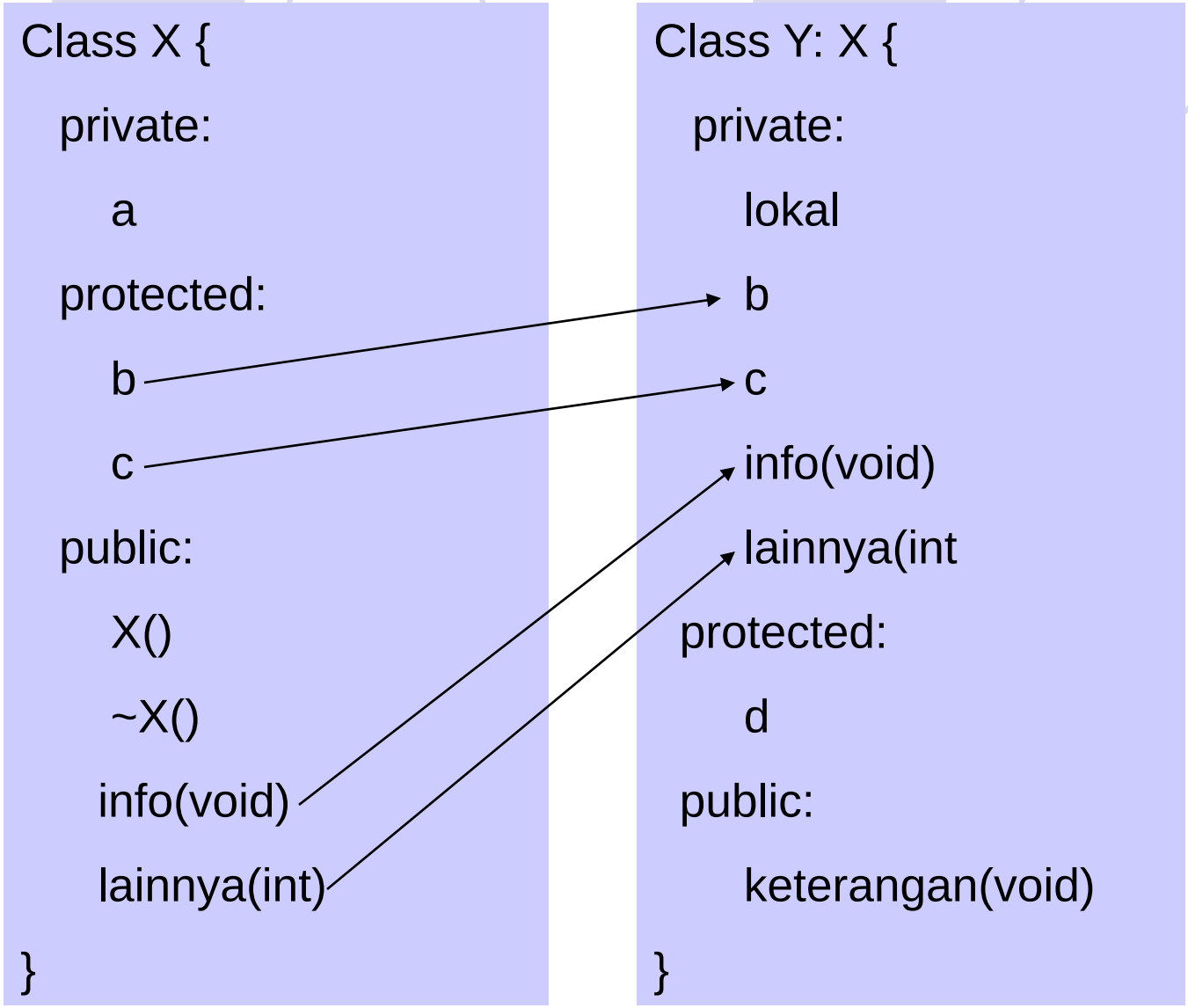
```
class kelas-turunan :mode-pewarisan kelas-dasar {
                    // ...
};
```

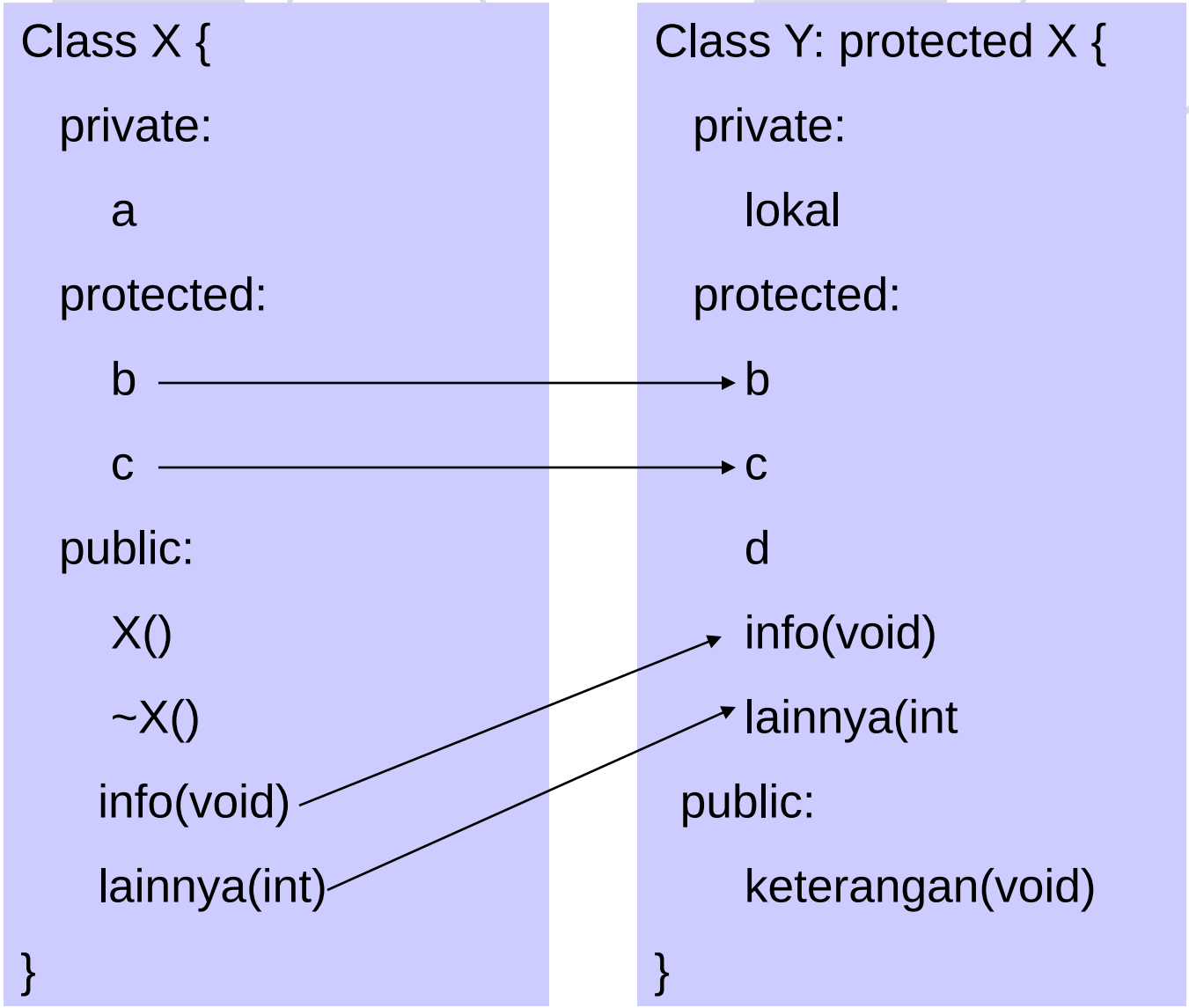
Tingkat-Akses di kelas dasar	Mode-pewarisan		
	`private`	`protected`	`public`
private	private	private	private
protected	private	protected	protected
public	private	protected	public

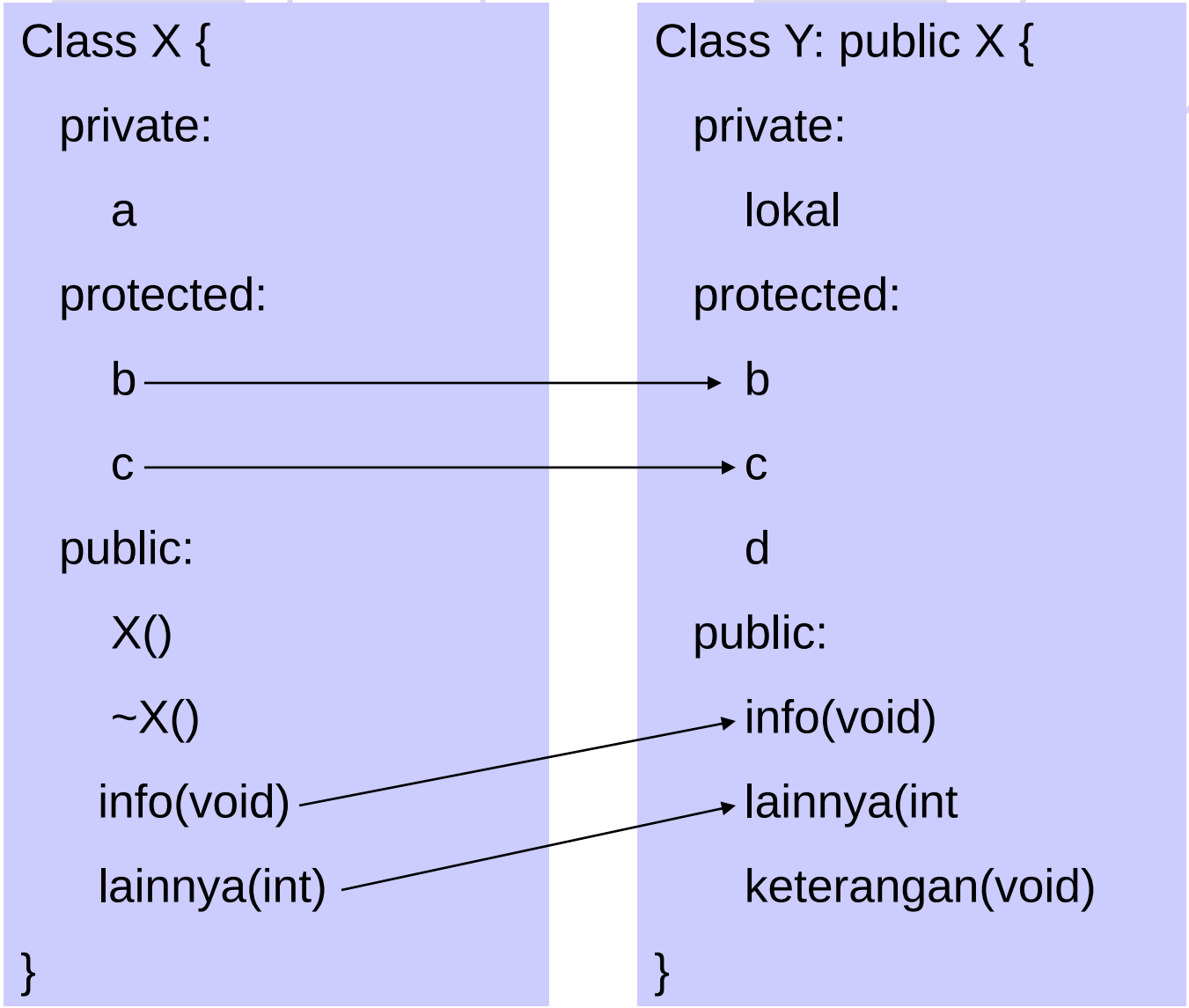
Jenis Pewarisan

Single, Multiple, Repeat, Multiple& Repeated




Class X {
private:
a
protected:
b
c
public:
X()
~X()
info(void)
lainnya(int)
}
Class Y: X {
private:
lokal
b
c
info(void)
lainnya(int
protected:
d
public:
keterangan(void)
}


Class X {
private:
a
protected:
b
c
public:
X()
~X()
info(void)
lainnya(int)
}
Class Y: protected X {
private:
lokal
protected:
b
c
d
info(void)
lainnya(int
public:
keterangan(void)
}


Class X {
private:
a
protected:
b
c
public:
X()
~X()
info(void)
lainnya(int)
}
Class Y: public X {
private:
lokal
protected:
b
c
d
public:
info(void)
lainnya(int
keterangan(void)
}

```
Class Basis{
    int alpha;
    int bravo;

  public:
     void info_basis(){
          cout << " info_basis() dijalankan " << endl;
     };
}

Class Turunan: private Basis{
     public:
          void info_turunan(){
               cout<< "info_Turunan() dijalankan" << endl;
          }
};

Int main(){
   Turunan anak;
    anak.info_basis();
    anak.info_turunan();
   return(0);
}
```

```
Class Basis{
    int alpha;
    int bravo;

  public:
     void info_basis(){
          cout << " info_basis() dijalankan " << endl;
     };
}

Class Turunan: public Basis{
     public:
          void info_turunan(){
               cout<< "info_Turunan() dijalankan" << endl;
          }
};

Int main(){
   Turunan anak;
    anak.info_basis();
    anak.info_turunan();
   return(0);
}
```

```
Class Basis{
  protected:
       int alpha;
       int bravo;

   public:
      void info_basis(){
            cout << " info_basis() dijalankan " << endl;
      };
}

Class Turunan: public Basis{
      public:
        void inisialisasi(int a, int b){
            alpha=a;
            bravo=b;
         }
         void info_turunan(){
                  cout<< "alpha = " << alpha
                        << "bravo = " << bravo <<endl;
          }
};
```

```
Int main(){
    Turunan anak;
     anak.inisialisasi(3, 4);
     anak.info_turunan();
    return(0);
}
```

```
Class Kendaraan{
  protected:
       char nama[15];

   public:
      Kendaraan(char *nama_Kendaraan="xxx"){
            strcpy(nama, nama_Kendaraan);
            cout<<"Hidupkan mesin kendaraan"<<endl;
      };
      ~Kendaraan(){
          cout<<"destruktor kendaraan dijalankan"<<endl;
      }
      void info(){
            cout<<nama<< "sedang berjalan"<<endl;
     }
}
```

```
Class Truk: public Kendaraan{
      public:
        Truk(char *nama_truk){
            strcpy(nama,nama_truk);
            cout<<“Hidupkan Mesin Truk” << nama << endl;
         }
         ~Truk(){
                  cout<< “Matikan mesin truk ….”  <<endl;
         }
};

Void main(){
      Truk Fuso(“Truk Fuso”);
      Fuso.info_kendaraan();
      cout<<“Akhir program”<<endl;
}
```

```
Class Kendaraan{
    char nama[15];

  public:
    Kendaraan(char *nama_Kendaraan="xxx"){
          strcpy(nama, nama_Kendaraan);
          cout<<"Hidupkan mesin kendaraan"<<endl;
    };
    ~Kendaraan(){
        cout<<"destruktor kendaraan dijalankan"<<endl;
    }
    void info(){
          cout<<nama<< "sedang berjalan"<<endl;
   }
}
```

```
Class Truk: public Kendaraan{
      public:
        Truk(char *nama_truk) : Kendaraan(nama_truk){
            cout<<“Hidupkan Mesin Truk” << nama << endl;
        }
       ~Truk(){
                cout<< “Matikan mesin truk ….”  <<endl;
       }
};

Void main(){
      Truk Fuso(“Truk Fuso”);
      Fuso.info_kendaraan();
      cout<<“Akhir program”<<endl;
}
```

```
Class Orang{
     char nama[50];
     int usia;
   public:
      Orang(char *nama, int usia);
      void info_orang();
}

Class Pegawai: public Orang{
       char bagian[25];
       int nip;
   public:
       pegawai(char *nama, int usia, char *bagian, int nip);
       void info_pegawai();
}

Class Manager: public Pegawai{
     char mobil[30];
   public:
     Manager(char *nama, int usia,char *bagian, int nip, char *mobil);
     void info_manager();
}
```

```
Void main(){
   Manager Kabag_EDP(“Udin”, 35, “EDP”, 1185, “Sedan Larantuka”);
   Kabag_EDP.info_manager();
}
Orang::Orang(char *nama, int usia){
     strcpy(Orang::nama, nama);
     Orang::usia=usia;
}
Void Orang::info(){
   cout<<“nama    = “ << nama<<endl;
   cout<<“Usia     = “ << usia << endl;
}
Pegawai::Pegawai(char *nama, int usia, char *bagian, int nip):Orang(nama, usia){
     strcpy(Pegawai::bagian, bagian);
     Pegawai::nip=nip;
}
Void Pegawai::info(){
   info_orang();
   cout<<“Bagian   = “ << bagian<<endl;
   cout<<“NIP       = “ << nip << endl;
}
```

```
Pegawai::Pegawai(char *nama, int usia, char *bagian, int nip, char *mobil):
Pegawai(nama, usia, bagian, nip){
    strcpy(Manager::mobil, mobil);
}

Void Manager::info(){
  info_pegawai();
  cout<<“Mobil   = “ <<mobil<<endl;
}
```

```
Class Buku{
     char judul[35];
     char pengarang[20];
     int harga;
   public:
      Buku(char *judul, char *pengarang, int harga);
      void info_buku();
}

Class CD{
       char ukuran[15];
       int harga;
   public:
       CD(char *ukuran, int harga);
       void info_CD();
}

Class PaketBukuCD:public Buku, public CD {
     int harga;
   public:
     PaketB_CD(char *judul,char *pengarang,int h_buku,char *ukuran,int h_CD);
     void info_paket();
}
```

```
Void main(){
    PaketBukuCD CPP(“C++”, “Ala kadarnya”,20000, “700 Mb”, 2500);
    CPP.info_paket();
}

Buku::Buku(char *judul, char *pengarang, int harga){
    strcpy(Buku::judul,judul);
    strcpy(Buku::pengarang,pengarang);
    Buku::harga=harga;
}
void Buku::info_buku(){
   cout<<“Judul     : “ << judul << endl;
   cout<<“Pengarang : “ << pengarang << endl;
   cout<<“Harga       :  “ << harga << endl;
}

CD::CD(char *ukuran, int harga){
  strcpy(CD::ukuran, ukuran);
  CD::harga = harga;
}
```

```
void CD::info_CD(){
   cout<<“Ukuran     : “ << ukuran << endl;
   cout<<“Harga CD : “ << harga << endl;

}

PaketB_CD::PaketB_CD(char *judul, char *pengarang, int h_buku, char *ukuran,
      int h_CD){
     harga=h_buku+h_CD;
}

void PaketB_CD::info_paket(){
    info_buku();
    info_CD;
    cout<<“Harga Total = “ << harga << endl;
}
```