

PRAKTIKKUM 7 PEWARISAN 3

1. Listing Program 1

```
#include <iostream>

class MANUSIA {
    char* nama;
    int tinggi;
    int berat;
public:
    void SetNama(char* N) {
        nama = N;
    }
    void SetTinggi(int T) {
        tinggi = T;
    }
    void SetBerat(int B) {
        berat = B;
    }
    char* GetNama() {
        return nama;
    }
    int GetTinggi() {
        return tinggi;
    }
    int GetBerat() {
        return berat;
    }
    // Membuat fungsi virtual
    virtual void Berjalan() {
        cout<<"Berjalan"<<endl;
    }
    virtual void Berpakaian() {
        cout<<"Berpakaian"<<endl;
    }
};

class MAHASISWA: public MANUSIA {
    char* universitas;
    char* jurusan;
    int semester;
public:
    void SetUniversitas(char* U) {
        universitas = U;
    }
    void SetJurusan(char* J) {
        jurusan = J;
    }
    void SetSemester(int smt) {
        semester = smt;
    }
};
```

```

char* GetUniversitas() {
    return universitas;
}
char* GetJurusan() {
    return jurusan;
}
int GetSemester() {
    return semester;
}
// Melakukan override terhadap fungsi Berjalan
void Berjalan() {
    cout<<"Berjalan dengan cara SANTAI"<<endl;
}
// Melakukan override terhadap fungsi Berpakaian
void Berpakaian() {
    cout<<"Berpakaian dengan baju BEBAS"<<endl;
}
};

class TENTARA: public MANUSIA {
    char* pangkat;
    char* kesatuan;
public:
    void SetPangkat(char* pkt) {
        pangkat = pkt;
    }
    void SetKesatuan(char* kstn) {
        kesatuan = kstn;
    }
    char* GetPangkat() {
        return pangkat;
    }
    char* GetKesatuan() {
        return kesatuan;
    }
    // Melakukan override terhadap fungsi Berjalan
    void Berjalan() {
        cout<<"Berjalan dengan cara TEGAP"<<endl;
    }
    // Melakukan override terhadap fungsi Berpakaian
    void Berpakaian() {
        cout<<"Berpakaian dengan baju SERAGAM"<<endl;
    }
};

// Fungsi utama
int main() {

    // Melakukan instansiasi terhadap kelas MANUSIA
    MANUSIA M;

    // Melakukan instansiasi terhadap kelas MAHASISWA
    MAHASISWA MHS;

    // Melakukan instansiasi terhadap kelas TENTARA
    TENTARA TTR;

```

```

// Memanggil fungsi Berjalan dari masing-masing kelas
M.Berjalan();
MHS.Berjalan();
TTR.Berjalan();
cout<<'\\n';

// Memanggil fungsi Berpakaian dari masing-masing kelas
M.Berpakaian();
MHS. Berpakaian ();
TTR. Berpakaian ();

return 0;
}

```

2. Listing Program 2

```

#include <iostream>

class KARYAWAN {
    char* NIP;
    char* nama;
    double GajiPokok;
public:
    void SetNIP(char* NoInduk) {
        NIP = NoInduk;
    }
    void SetNama(char* nm) {
        nama = nm;
    }
    void SetGajiPokok(double GP) {
        GajiPokok = GP;
    }
    char* GetNIP() {
        return NIP;
    }
    char* GetNama() {
        return nama;
    }
    double GetGajiPokok() {
        return GajiPokok;
    }
    // Membuat fungsi virtual
    virtual void Bekerja() {
        cout<<"Bekerja"<<endl;
    }
};

// Membuat kelas PROGRAMMER yang diturunkan dari kelas KARYAWAN
class PROGRAMMER: public KARYAWAN {
    double bonus;
public:
    // Melakukan overload pada fungsi constructor kelas PROGRAMMER
    PROGRAMMER() {
        bonus = 0;
    }
    PROGRAMMER(double bns) {
        bonus = bns;
    }
}

```

```

double GetBonus() {
    return bonus;
}

// Melakukan override terhadap fungsi Bekerja
void Bekerja() {
    cout<<"Bekerja dengan cara membuat program " \
        "menggunakan komputer";
    cout<<endl;
}
};

class GURU: public KARYAWAN {
    char* MateriAjar;
public:
    void SetMateriAjar(char* MA) {
        MateriAjar = MA;
    }
    char* GetMateriAjar() {
        return MateriAjar;
    }

    // Melakukan override terhadap fungsi Bekerja
    void Bekerja() {
        cout<<"Bekerja dengan cara mengajar di depan kelas";
        cout<<'\n';
    }
};

// Fungsi utama
int main() {

    // Melakukan instansiasi terhadap kelas PROGRAMMER
    // dengan mengeset bonus sebesar 1000000
    PROGRAMMER P(1000000);

    // Melakukan instansiasi terhadap kelas GURU
    GURU G;

    // Memanggil fungsi bekerja untuk kelas PROGRAMMER dan GURU
    P.Bekerja();
    G.Bekerja();

    return 0;
}

```

3. Listing Program 3

```

#include <iostream>

class KARYAWAN {
public:
    virtual void Bekerja() = 0;
};

class PROGRAMMER: public KARYAWAN {

```

```
public:
    void Bekerja() {
        cout<<"Bekerja dengan cara membuat program " \
            "menggunakan komputer";
        cout<<endl;
    }
};

class GURU: public KARYAWAN {
public:
    void Bekerja() {
        cout<<"Bekerja dengan cara mengajar di depan kelas";
        cout<<endl;
    }
};

// Fungsi utama
int main() {

    // Mendeklarasikan pointer yang bertipe kelas KARYAWAN
    KARYAWAN *P;

    P = new PROGRAMMER;
    P->Bekerja();

    delete P;    // Menghapus pointer P
                 // yang menunjuk ke tipe PROGRAMMER

    P = new GURU;
    P->Bekerja();

    return 0;
}
```