

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LABSHEET ALGORITMA DAN STRUKTUR DATA</b>		
	Semester 1	<b>KONSEP DASAR OOP</b>	
	No. LST/TE/EKA5208/09	Revisi : 01	Tgl : 20 Februari 2017

#### A. KOMPETENSI

1. Memahami pengertian pemrograman berorientasi objek
2. Memahami perbedaan pemrograman prosedural dan pemrograman berorientasi objek
3. Memahami karakteristik pemrograman berorientasi objek
4. Memahami pengertian abstraksi, encapsulation, inheritance, dan polimorfisme

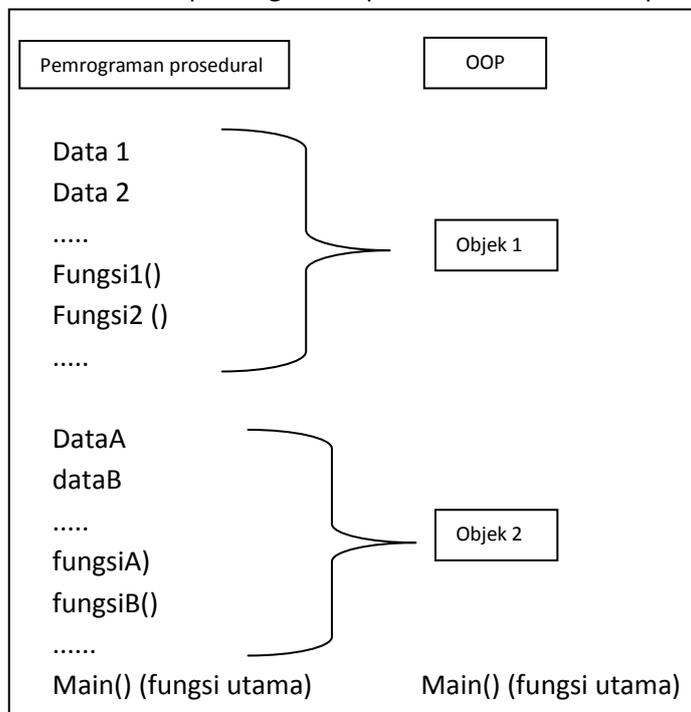
#### B. ALAT DAN BAHAN

1. PC/ Laptop
2. Flowchart application
3. C++ (atau bahasa pemrograman lain yang dikuasai)
4. Labsheet

#### C. KAJIAN TEORI

OOP merupakan bentuk penyederhanaan dari bahasa prosedural sehingga program akan lebih mudah dikembangkan. Pembuatan program dengan metode OOP yaitu menyederhanakan program dengan cara memecahnya ke dalam sub-sub program yang dinamakan kelas, setelah membuat kelas dibuatlah relasi antar kelas.

Perbedaan pemrograman prosedural dan OOP dapat diperhatikan pada gambar berikut:



Berikut ini adalah contoh penulisan kode program dengan menggunakan metode prosedural dan OOP.

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LABSHEET ALGORITMA DAN STRUKTUR DATA</b>		
	Semester 1	<b>KONSEP DASAR OOP</b>	
	No. LST/TE/EKA5208/09	Revisi : 01	Tgl : 20 Februari 2017

**Contoh 1** pembuatan kode program dengan gaya prosedural:

```
#include <stdio.h>

// Mendeklarasikan variabel-variabel global
int A, B, C;

// Membuat prototipe fungsi
void set_A(int AA);
int get_A();

void set_B(int BB);
int get_B();

void set_C(int CC);
int get_C();

// Mendefinisikan fungsi utama
int main(void) {

    // Melakukan pemanggilan fungsi
    set_A(10);
    set_B(20);
    set_C(30);

    // Menampilkan nilai
    printf("Nilai A: %d\n", A);
    printf("Nilai B: %d\n", B);
    printf("Nilai C: %d", C);

    return 0;
}

// Bagian implementasi dari fungsi
// yang terdapat pada bagian prototipe
void set_A(int AA) {
    A = AA;
}

int get_A() {
    return A;
}

void set_B(int BB) {
    B = BB;
}

int get_B() {
    return B;
}

void set_C(int CC) {
    C = CC;
}

int get_C() {
```

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LABSHEET ALGORITMA DAN STRUKTUR DATA</b>		
	Semester 1	<b>KONSEP DASAR OOP</b>	
	No. LST/TE/EKA5208/09	Revisi : 01	Tgl : 20 Februari 2017

```
return C;
}
```

Hasil:

```

Nilai A : 10
Nilai B : 20
Nilai C : 30

```

Contoh 2 pembuatan kode program dengan OOP:

```

#include <iostream>

using namespace std;

// Membuat kelas dengan nama KELAS1
class KELAS1 {
    int A;
public:
    void set_A(int AA);
    int get_A();
    void show_A();
};

// Membuat kelas dengan nama KELAS2
class KELAS2 {
    int B;
public:
    void set_B(int BB);
    int get_B();
    void show_B();
};

// Membuat kelas dengan nama KELAS3
class KELAS3 {
    int C;
public:
    void set_C(int CC);
    int get_C();
    void show_C();
};

// Implementasi dari fungsi
// yang terdapat pada KELAS1
void KELAS1::set_A(int AA) {
    A = AA;
}

int KELAS1::get_A() {
    return A;
}

void KELAS1::show_A() {
    cout<<"Nilai A : "<<A<<endl;
}

// Implementasi dari fungsi
// yang terdapat pada KELAS3
void KELAS2::set_B(int BB) {

```

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LABSHEET ALGORITMA DAN STRUKTUR DATA</b>		
	Semester 1	<b>KONSEP DASAR OOP</b>	
	No. LST/TE/EKA5208/09	Revisi : 01	Tgl : 20 Februari 2017

```

    B = BB;
}

int KELAS2::get_B() {
    return B;
}

void KELAS2::show_B() {
    cout<<"Nilai B : "<<B<<endl;
}

// Implementasi dari fungsi
// yang terdapat pada KELAS3
void KELAS3::set_C(int CC) {
    C = CC;
}

int KELAS3::get_C() {
    return C;
}

void KELAS3::show_C() {
    cout<<"Nilai C : "<<C<<endl;
}

// Mendefinisikan fungsi utama
int main() {

    // Mendeklarasikan objek OB1, OB2, dan OB3
    KELAS1 OB1; // OB1 bertipe KELAS1
    KELAS2 OB2; // OB2 bertipe KELAS2
    KELAS3 OB3; // OB3 bertipe KELAS3

    // Melakukan pengisian nilai
    OB1.set_A(10);
    OB2.set_B(20);
    OB3.set_C(30);

    // Menampilkan nilai
    OB1.show_A();
    OB2.show_B();
    OB3.show_C();

    return 0;
}

```

Hasil:

```

Nilai A : 10
Nilai B : 20
Nilai C : 30

```

Pahamilah kedua kode program di atas dan jelaskan bagaimana perbedaannya.

	<b>FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA</b>		
	<b>LABSHEET ALGORITMA DAN STRUKTUR DATA</b>		
	Semester 1	<b>KONSEP DASAR OOP</b>	
	No. LST/TE/EKA5208/09	Revisi : 01	Tgl : 20 Februari 2017

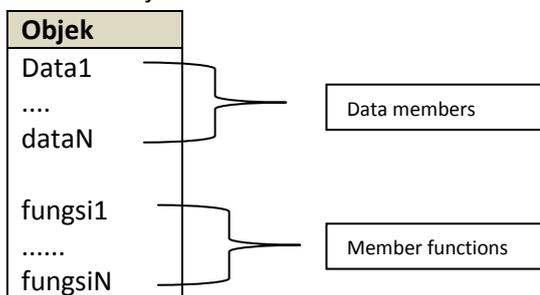
### Karakteristik OOP

#### 1. Abstraction

Merupakan proses pengabstrakan atau menyembunyikan detail program yang sangat rumit sehingga tidak perlu mempermasalahkan bagaimana pembuatannya akan tetapi hanya perlu supaya objek tersebut dapat digunakan sesuai fungsinya.

#### 2. Encapsulation

Yaitu sebuah proses penggabungan antara data dan fungsi-fungsi yang berkaitan ke dalam sebuah objek.



#### 3. Inheritance

Dalam OOP kelas baru dapat diciptakan dengan menurunkan dari kelas lain yang sudah didefinisikan sebelumnya. Kelas baru disebut dengan kelas turunan (*derived class*) sedangkan objek induk disebut dengan kelas dasar (*base class*). Sifat yang terkandung dalam objek turunan adalah sifat hasil pewarisan dari sifat-sifat yang terdapat pada objek induk. Dengan cara ini objek baru dapat dibuat dengan memiliki kemampuan lebih dibandingkan objek induk, karena sifat-sifat baru ditambahkan ke dalam objek tersebut.

#### 4. Polimorfisme

Merupakan proses kebanyakan, yaitu sesuatu hal yang berbedda dapat diimplementasikan melalui satu cara yang sama

### D. LANGKAH KERJA

1. Berdoalah sebelum memulai pekerjaan.
2. Baca dan pahami labsheet yang diberikan.
3. Kerjakan contoh-contoh hasil program di atas .
4. Apabila mengalami kesulitan atau pekerjaan telah selesai, konsultasikan pada dosen pengampu.
5. unggah hasil pekerjaan anda di bestmart pada mata kuliah algoritma dan struktur data.

### E. TUGAS

1. Pelajari dan pahami kode program di atas.
2. Buatlah algoritma dengan bahasa narasi, flowchart, pseudocode, dan program di atas.
3. Buatlah 3 algoritma dan program dengan menggunakan OOP.

	FAKULTAS TEKNIK UNIVERSITAS NEGERI YOGYAKARTA		
	LABSHEET ALGORITMA DAN STRUKTUR DATA		
	Semester 1	<b>KONSEP DASAR OOP</b>	
	No. LST/TE/EKA5208/09	Revisi : 01	Tgl : 20 Februari 2017

#### F. DAFTAR PUSTAKA

Kajian pustaka diambil dari buku:

**Budi Raharjo. 2015. Pemrograman C++ Mudah dan Cepat Menjadi Master C++. Bandung: Penerbit Informatika.**