Pemograman Berorientasi Objek

C#

#1 Konsep Dasar PBO

Indonesia Computer University

2017

OOP Lecturer

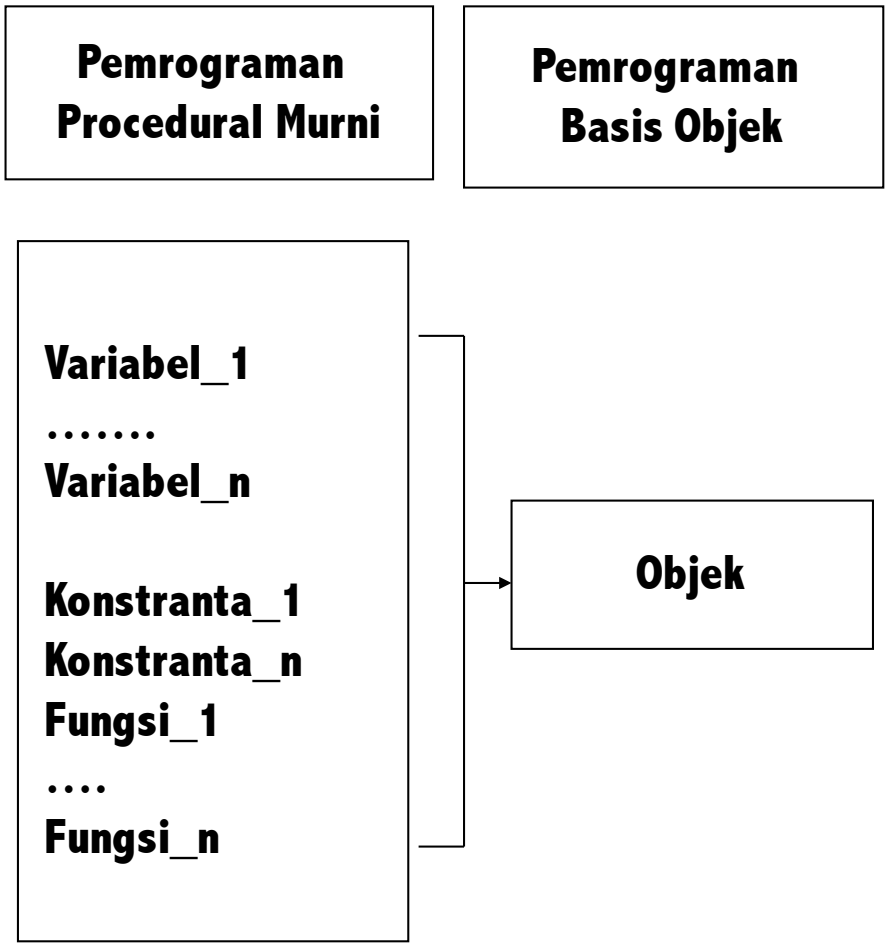
Susmini Indriani L, M.T
Sistem Komputer
susmini.indriani@email.unikom.ac.id

Sri Nurhayati, S.Si, M.T
Teknik Komputer
sri.nurhayati@email.unik

Kompetensi Dasar

- Setelah mengikuti mata kuliah ini diharapkan mahasiswa memiliki kemampuan untuk:
    1. Menguraikan konsep dasar PBO
    2. Mengimplementasikan kelas (class) dan metode (method) untuk mewakili obyek-obyek dalam sistem
    3. Mengimplementasikan abstrak dan interface dalam suatu kelas
    4. Menggunakan initialization dan instance dalam class
    5. Menerapkan konsep orientasi objek : Inheritance dalam sistem
    6. Menerapkan konsep Polimorfisme
    7. Menguraikan dasar UML
    8. Mengindentifikasi Permasalahan menggunakan Use Case Diagram
    9. Mengindentifikasi Permasalahan menggunakan Activity Diagram
    10. Mengindentifikasi Permasalahan menggunakan Sequence Diagram
    11. Menyusun Class Diagram dari permasalahan
    12. Mewujudkan Object Oriented Design (OOD) dan Object Oriented Programing (OOP) menjadi sebuah aplikasi (Studi Kasus)

Konsep PBO


Pemrograman Procedural Murni
Pemrograman Basis Objek
Variabel_1
.....
Variabel_n
Konstranta_1
Konstranta_n
Fungsi_1
....
Fungsi_n
Objek

Konsep Object Oriented

- Ide dasarnya adalah menggabungkan data dan fungsi menjadi satu kesatuan unit yang dikenal sebagai **obyek (*object*)**
- Di dunia nyata, tiap obyek memiliki **ciri (atribut)** dan **tingkah laku (behavior).**
- Misalnya :
  - Obyek: mahasiswa
  - Ciri bisa dilihat dari warna kulit, suara, jenis kelamin.
  - Tingkah laku : berlari, menulis, makan.

Konsep Object Oriented

- Contoh:
  - Objek sebuah mobil mempunyai atribut tipe transmisi, warna dan manufaktur. Mempunyai tingkah laku berbelok, mengerem dan berakselerasi. Dengan cara yang sama pula kita dapat mendefinisikan perbedaan sifat dan tingkah laku dari singa.

Obyek	*Atribut*	*Tingkah Laku*
Mobil	Tipe dari transmisi manufaktur Warna	Berbelok Mengerem Mempercepat
Singa	Berat Warna Lapar atau tidak lapar Jinak atau liar	roaring Tidur Berburu

Yang harus diperhatikan adalah

- Object
- Class
- Attribute
- Method
- Encapsulation/data hiding
- Inheritance
- Polymorphism

Object vs Class

- Pada dunia perangkat lunak, sebuah obyek adalah sebuah komponen perangkat lunak yang strukturnya mirip dengan obyek pada dunia nyata. Setiap obyek dibangun dari sekumpulan data (atribut) yang disebut variabel untuk menjabarkan karakteristik khusus dari obyek, dan juga terdiri dari sekumpulan method yang menjabarkan tingkah laku dari obyek. Bisa dikatakan bahwa obyek adalah sebuah perangkat lunak yang berisi sekumpulan variabel dan method yg berhubungan.
- Class adalah struktur dasar dari OOP. Class terdiri dari dua tipe dari anggota dimana disebut dengan field (attribut/properti) dan method. Field merupakan tipe data yang didefinisikan oleh class, sementara method merupakan operasi. Sebuah obyek adalah sebuah instance (keturunan) dari class.

Contoh Class vs Object

- Kita memiliki sebuah class mobil dimana dapat digunakan untuk mendefinisikan beberapa obyek mobil. Pada tabel dibawah, mobil A dan mobil B adalah obyek dari class mobil. Class memiliki field nomor, plat, warna, manufaktur dan kecepatan yang diisi dengan nilai pada obyek mobil A dan mobil B. Mobil juga dapat berakselerasi, berbelok dan melakukan rem.

Class mobil		Obyek mobil A	Obyek Mobil B
Variabel Intsance	Nomor Plat	ABC 111	XYZ 123
	Warna	Biru	Merah
	Manufaktur	Mitsubishi	Toyota
	Kecepatan	50 km/h	100 km/h
Method *Instance*	Method Akselerasi		
	Method Belok		
	Method Rem		

Object-oriented Analysis (OOA)

- OOA adalah sebuah pendekatan yang digunakan untuk :
  1. Menggunakan obyek yang sudah ada untuk digunakan kembali (reuse) atau diadaptasi untuk penggunaan baru
  2. Mendefinisikan obyek baru atau obyek yang dimodifikasi dan digabungkan dengan obyek yang sudah ada untuk membangun suatu aplikasi bisnis.

- **Object Modeling (Pemodelan Obyek)** – Teknik untuk mengidentifikasi obyek yang ada dalam sistem dan relasi diantara obyek-obyek tersebut.

Mengapa harus Object Oriented?

- **Beberapa Alasan:**
  1. Pengembangan perangkat lunak itu sulit karena “perangkat lunak mudah diimpikan”
  2. Kompleksitas pengembangan perangkat lunak yang terus bertumbuh
  3. membutuhkan dukungan konsep yang lebih handal, guna ulang (reusable) dan natural
  4. OO menawarkan tipe data abstrak, modularitas, pemodelan informasi, proses software untuk mengatasinya
  5. Walaupun demikian, OO bukan jaminan sukses pengembangan perangkat lunak

Kapan dan Dimana OO?

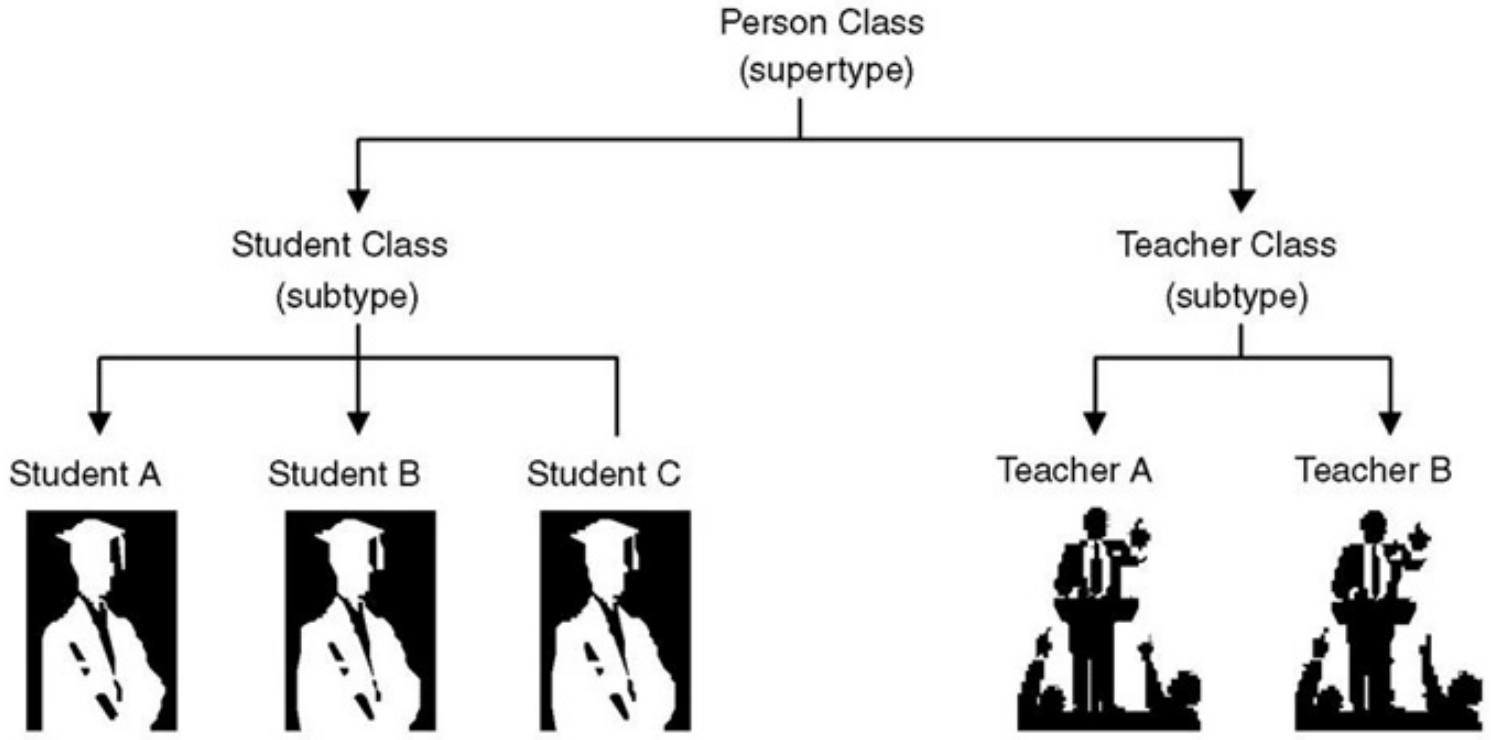
- **Beberapa Situasi Umum:**
  1. Jika perangkat lunak (PL) yang dibangun cukup kompleks
  2. Jika PL yang dibangun diperkirakan akan tumbuh makin kompleks di masa mendatang
  3. Jika kita ingin membangun PL yang:
     - Berdasar pada komponen yang telah pernah ada sebelumnya (daur ulang)
     - Dapat dipergunakan kembali di masa mendatang (reusable)
  4. Dan mungkin, kapanpun dan di manapun

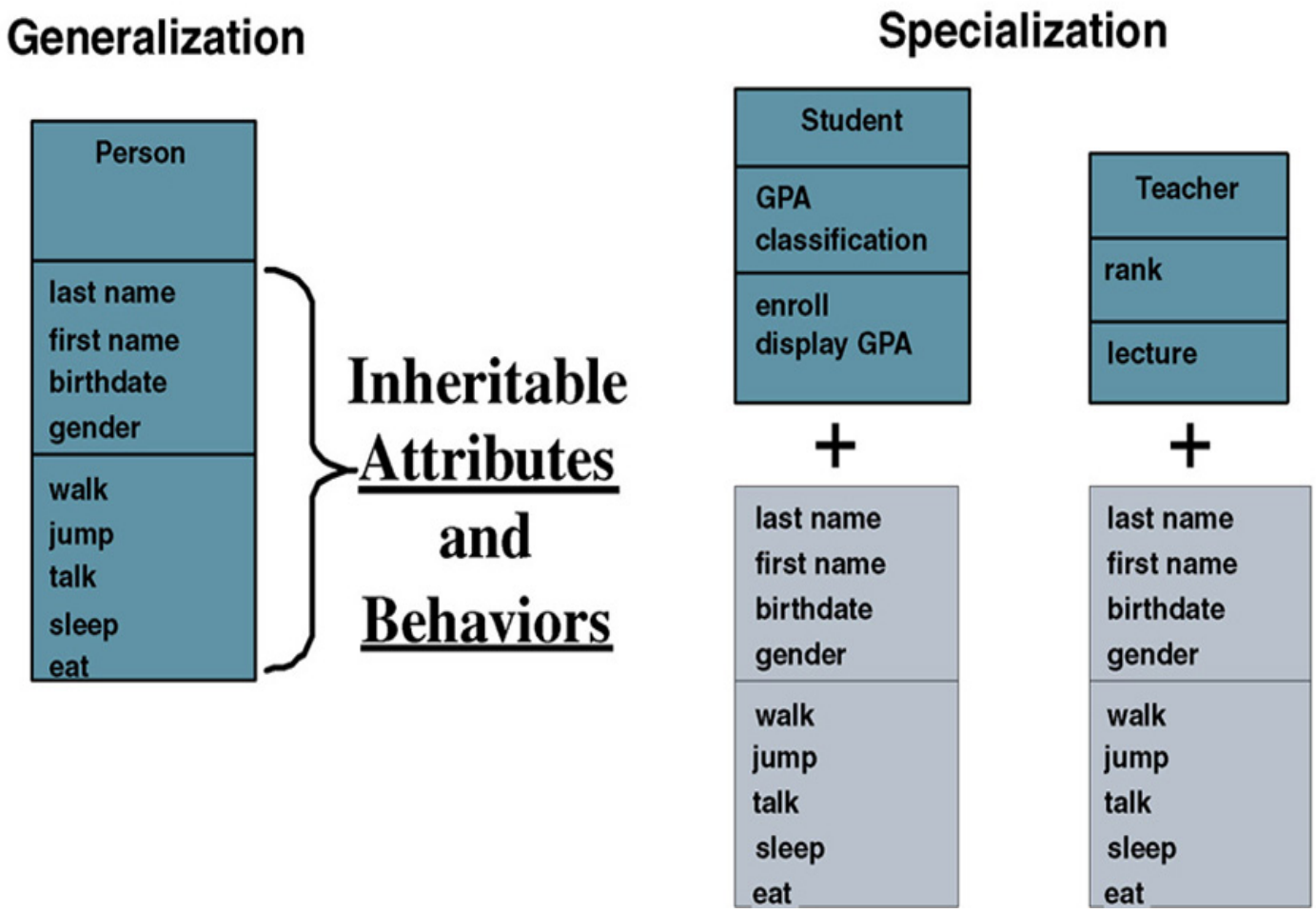
Pembungkusan (Encapsulation)

- Dalam sistem berorientasi objek kita menggabungkan potongan-potongan informasi dan perilaku-perilaku spesifik yang bekerja pada informasi tersebut, kemudian mengemasnya menjadi apa yang disebut dengan objek. Inilah yang disebut dengan pembungkusan (encapsulation).

Pewarisan (Inheritance)

- Pewarisan (Inheritance) – konsep yang menyatakan bahwa metode atau atribut dalam kelas dapat diturunkan atau digunakan kembali oleh kelas lain.



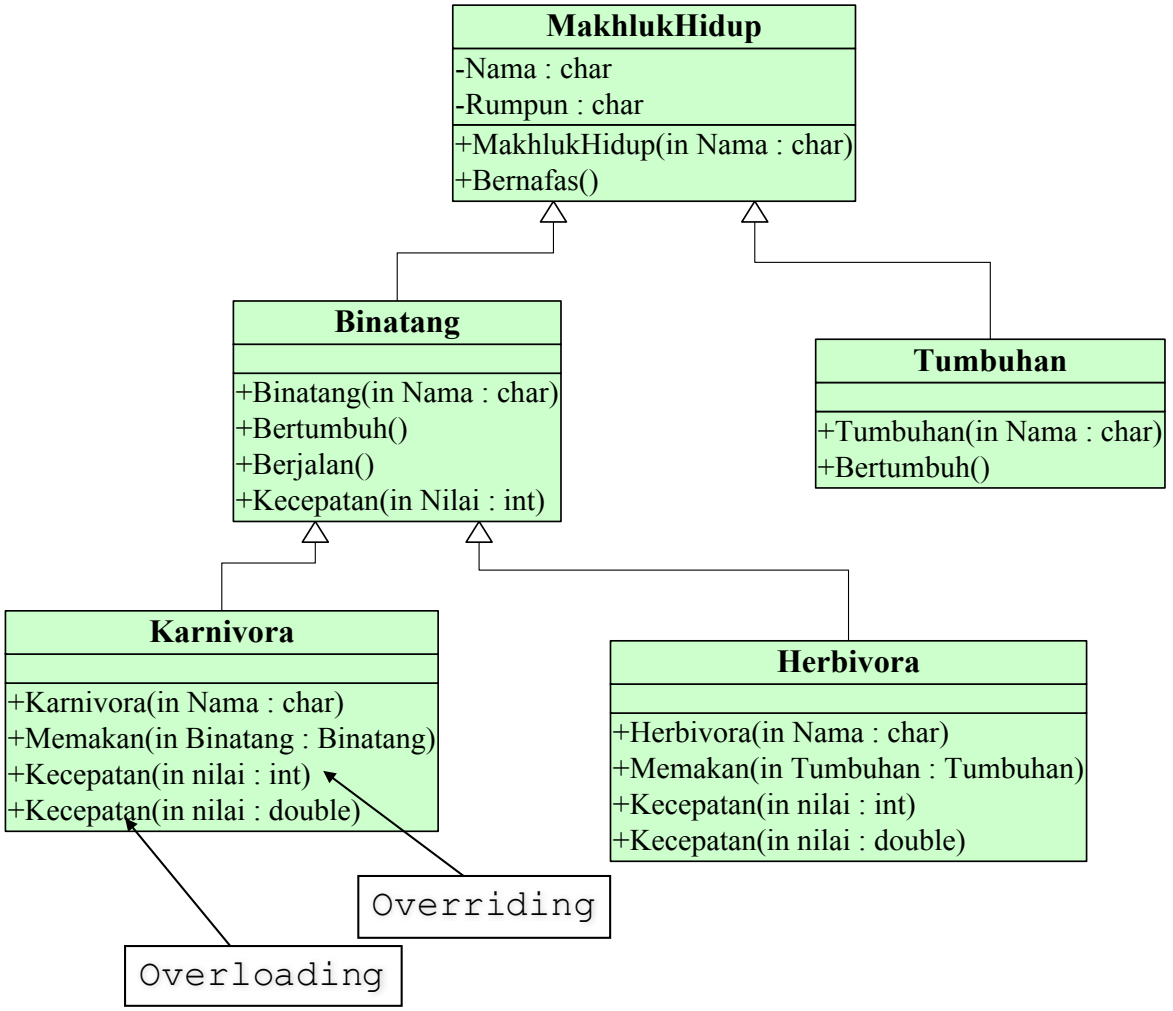

Generalization
Specialization
Person
last name
first name
birthdate
gender
walk
jump
talk
sleep
eat
Inheritable
Attributes
and
Behaviors
Student
GPA
classification
enroll
display GPA
+
last name
first name
birthdate
gender
walk
jump
talk
sleep
eat
Teacher
rank
lecture
+
last name
first name
birthdate
gender
walk
jump
talk
sleep
eat

Polimorfisme (Polymorfism)

- Polimorfisme diturunkan dari bahasa latin yaitu poly yang berarti banyak dan morph yang berarti bentuk
- Polimorfisme sendiri berarti sesuatu yang memiliki banyak bentuk
- Polimorfisme mengijinkan kelas induk untuk mendefiniskan sebuah method general (bersifat umum) untuk semua kelas turunannya, dan selanjutnya kelas-kelas turunan dapat memperbaharui implementasi dari method tersebut secara lebih spesifik sesuai dengan karakteristiknya masing-masing.

Overriding & Overloading

- **Overriding** : terjadi ketika deklarasi method subclass sama (termasuk parameter) dengan method pada superclass.
- **Overloading** : yaitu penggunaan satu nama untuk beberapa method yang berbeda (berbeda parameter)


MakhlukHidup
-Nama : char
-Rumpun : char
+MakhlukHidup(in Nama : char)
+Bernafas()
Binatang
+Binatang(in Nama : char)
+Bertumbuh()
+Berjalan()
+Kecepatan(in Nilai : int)
Tumbuhan
+Tumbuhan(in Nama : char)
+Bertumbuh()
Karnivora
+Karnivora(in Nama : char)
+Memakan(in Binatang : Binatang)
+Kecepatan(in nilai : int)
+Kecepatan(in nilai : double)
Herbivora
+Herbivora(in Nama : char)
+Memakan(in Tumbuhan : Tumbuhan)
+Kecepatan(in nilai : int)
+Kecepatan(in nilai : double)
Overriding
Overloading

Keuntungan Menggunakan Pemodelan OO

- Kemampuan untuk menangani tipe-tipe data dan masalah-masalah yang lebih kompleks dan lebih sulit.
- Memperbaiki komunikasi antara pengguna, analisis, perancangan dan pemrogram.
- Meningkatkan derajad konsistensi antara tahap analisis, perancangan, serta kegiatan pemrograman karena sama untuk setiap tahap itu.
- Ketangguhan dan ketegaran sistem (robustness).
- Kemampuan untuk menggunakan ulang hasil-hasil analisis, perancangan serta pemrograman (reusable component) pada suatu proyek ke proyek lainnya.
- Meningkatkan konsistensi antara model-model yang dikembangkan selama analisis, perancangan, serta pemrograman berorientasi objek.

Unified Modelling Language (UML)

- Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak.
- UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan piranti lunak dalam bahasa bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk modeling aplikasi prosedural dalam VB atau C.

UML Diagram:

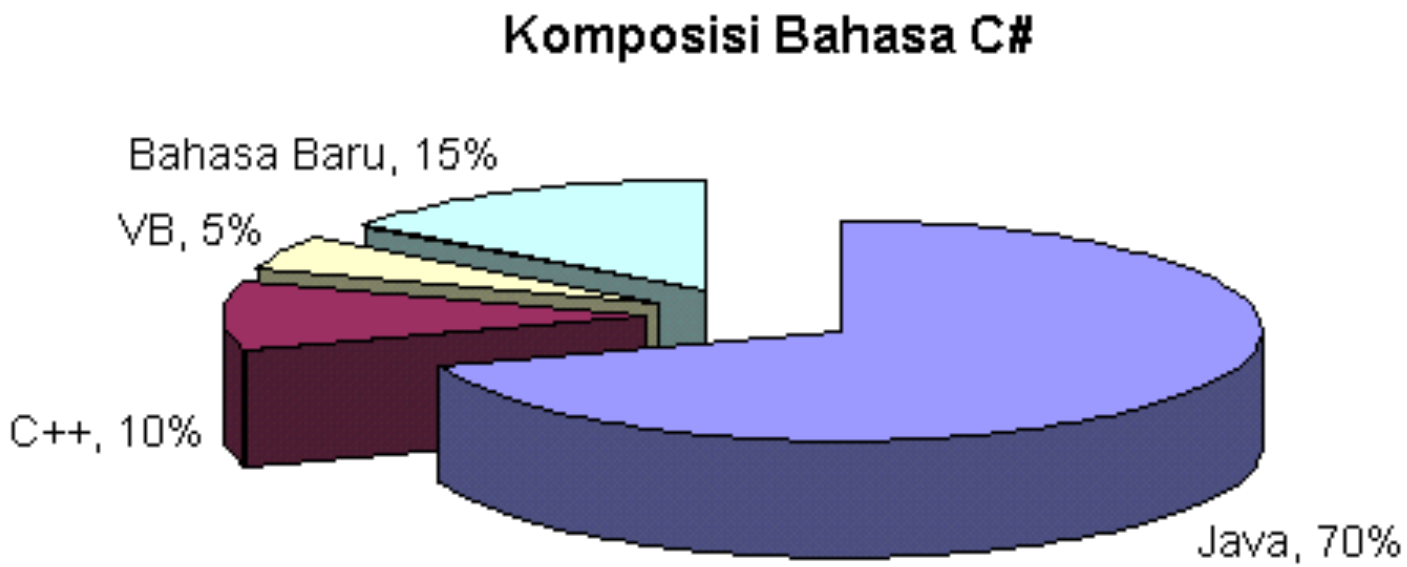
- use case diagram
- conceptual Diagram
- class diagram
- object diagram
- state chart diagram
- activity diagram
- sequence diagram
- collaboration diagram
- component diagram
- deployment diagram

Pengenalan C#

- C# (C sharp) adalah sebuah bahasa pemograman berorientasi objek yang didukung oleh microsoft.NET Framework.
- Diciptakan → Microsoft (Anders Hejlsberg)
- Aplikasi yang bisa dibuat dengan C#:
  - Aplikasi Console
  - Aplikasi Windows (Dekstop)
  - Aplikasi Web
  - Aplikasi Web Services

Mengapa menggunakan C#?

- Sederhana (simple)



- Modern exception handling, garbage collection, extensible data types, dan code security.
- Object-Oriented Language ➔ encapsulation, inheritance, dan polymorphism.
- Powerfull dan fleksibel ➔ C# bisa digunakan untuk membuat berbagai macam aplikasi, seperti aplikasi pengolah kata, grafik, spreadsheets, atau bahkan membuat kompiler untuk sebuah bahasa pemrograman
- Efisien ➔ jumlah kata-kata (keywords) yang tidak terlalu banyak.
- Modular ➔ terdapat class yang terdiri dari method. Memiliki sifat reusable code.
- C# akan menjadi populer

Editor C#

- Notepad
- Visual Studio 6
- Visual Studio .NET
- Editor-editor Lainnya:
- Visual SlickEdit dari MicroEdge,
- WebMatrikx untuk aplikasi C# berbasis web,
- editor text seperti UltraEdit,
- Macromedia Homesite, dll

Struktur Penulisan Program C#

```
using System;
using System.Collections.Generic;
using System.Text;

namespace Latihan1
{
    public class HelloWorld
    {
        // Bagian utama program C#
        public void Main()
        {
            Console.WriteLine("Hello, World");
        }
    }
}
```

- **Using ➔** Library
- **Namespace ➔** menyatakan aplikasi kita atau lebih dikenal dengan paket aplikasi (application package).
- **Output ➔** berupa tulisan “Hello, World” yang akan tampil pada mode console (mode dos prompt),
- **Komentar**

  menggunakan //komentar,

  /* komentar */

Tugas

- Tulislah beberapa contoh class, object dan method dari kehidupan anda sehari-hari! minimal 5 jenis!