

# ANALISA & PERANCANGAN SISTEM INFORMASI

Sufajar Butsianto, M.Kom

MODUL 4

**DESAIN SISTEM**



Tools :

Star UML, Rational Rose dll

TOOLS

A decorative graphic consisting of several parallel white lines of varying lengths, slanted upwards from left to right, located in the bottom right corner of the slide.

1. Mahasiswa memahami apa yang dilakukan dalam tahap desain sistem
2. Mahasiswa memahami desain sistem dengan pendekatan berorientasi objek
3. Mahasiswa mampu membuat desain sistem dengan pendekatan berorientasi objek

PENCAPAIAN  
PEMBELAJARAN



# 1. DEFINISI DESAIN SISTEM

Desain atau perancangan dalam pembangunan perangkat lunak merupakan upaya untuk mengonstruksi sebuah sistem yang memberikan kepuasan (mungkin informal) akan spesifikasi kebutuhan fungsional, memenuhi target, memenuhi kebutuhan secara implisit atau eksplisit dari segi performansi maupun penggunaan sumber daya, kepuasan batasan pada proses desain dari segi biaya, waktu, dan perangkat. Kualitas perangkat lunak biasanya dinilai dari segi kepuasan pengguna perangkat lunak terhadap perangkat lunak yang digunakan.

## 2. KONSEP DASAR PENDEKATAN BERORIENTASI OBJEK

Pendekatan berorientasi objek merupakan suatu teknik atau cara pendekatan dalam melihat permasalahan dan sistem (sistem perangkat lunak, sistem informasi, atau sistem lainnya). Pendekatan berorientasi objek akan memandang sistem yang akan dikembangkan sebagai suatu kumpulan objek yang berkorespondensi dengan objek-objek dunia nyata. Ada banyak cara untuk mengabstraksikan dan memodelkan objek-objek tersebut, mulai dari abstraksi objek, kelas, hubungan antar kelas sampai abstraksi sistem.

## 2. KONSEP DASAR

### PENDEKATAN BERORIENTASI

**OBJEK COUNT** karakteristik yang dimiliki oleh sistem berorientasi obyek, sbb:

- ❑ Abstraksi
- ❑ Enkapsulasi
- ❑ Pewarisan (*Inheritance*)
- ❑ *Reusability*
- ❑ Generalisasi dan Spesialisasi
- ❑ Komunikasi Antar Obyek
- ❑ *Polymorphism*

# 3. METODOLOGI BERORIENTASI OBJEK

Metodologi berorientasi objek adalah suatu strategi pembangunan perangkat lunak yang mengorganisasikan perangkat lunak sebagai kumpulan objek yang berisi data dan operasi yang diberlakukan terhadapnya. Metodologi berorientasi objek merupakan suatu cara bagaimana sistem perangkat lunak dibangun melalui pendekatan objek secara sistematis. Metode berorientasi objek meliputi rangkaian aktivitas analisis berorientasi objek, perancangan berorientasi objek, pemrograman berorientasi objek, dan pengujian berorientasi objek.



### 3. METODOLOGI BERORIENTASI OBJEK -COUNT

Keuntungan menggunakan metodologi berorientasi objek adalah sebagai berikut:

- ❑ Meningkatkan produktivitas
- ❑ Kecepatan pengembangan
- ❑ Kemudahan pemeliharaan
- ❑ Adanya konsistensi
- ❑ Meningkatkan kualitas perangkat lunak

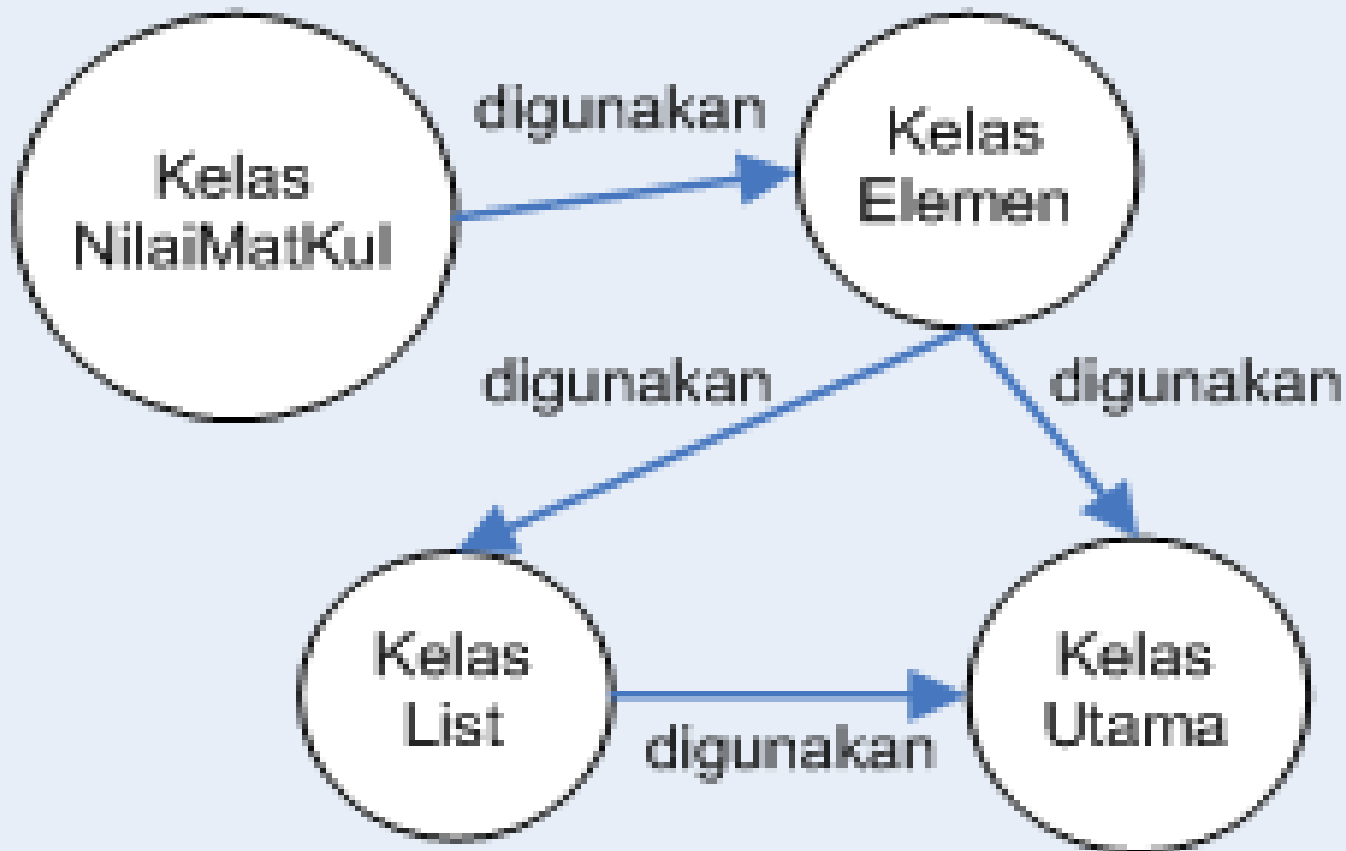
## 4. OBYEK DAN KELAS

Kelas adalah kumpulan dari objek-objek dengan karakteristik yang sama. Kelas merupakan definisi statik dan himpunan objek yang sama yang mungkin lahir atau diciptakan dan kelas tersebut. Sebuah kelas akan mempunyai sifat (atribut), kelakuan (operasi/metode), hubungan (relationship) dan arti. Suatu kelas dapat diturunkan dan kelas yang lain, dimana atribut dan kelas semula dapat diwariskan ke kelas yang baru.

Secara teknis, kelas adalah sebuah struktur tertentu dalam pembuatan perangkat lunak. Kelas merupakan bentuk struktur pada kode program yang menggunakan metodologi berorientasi objek. Ilustrasi dari sebuah kelas dapat dilihat pada gambar berikut.

# ILUSTRASI KELAS

dalam satu direktori

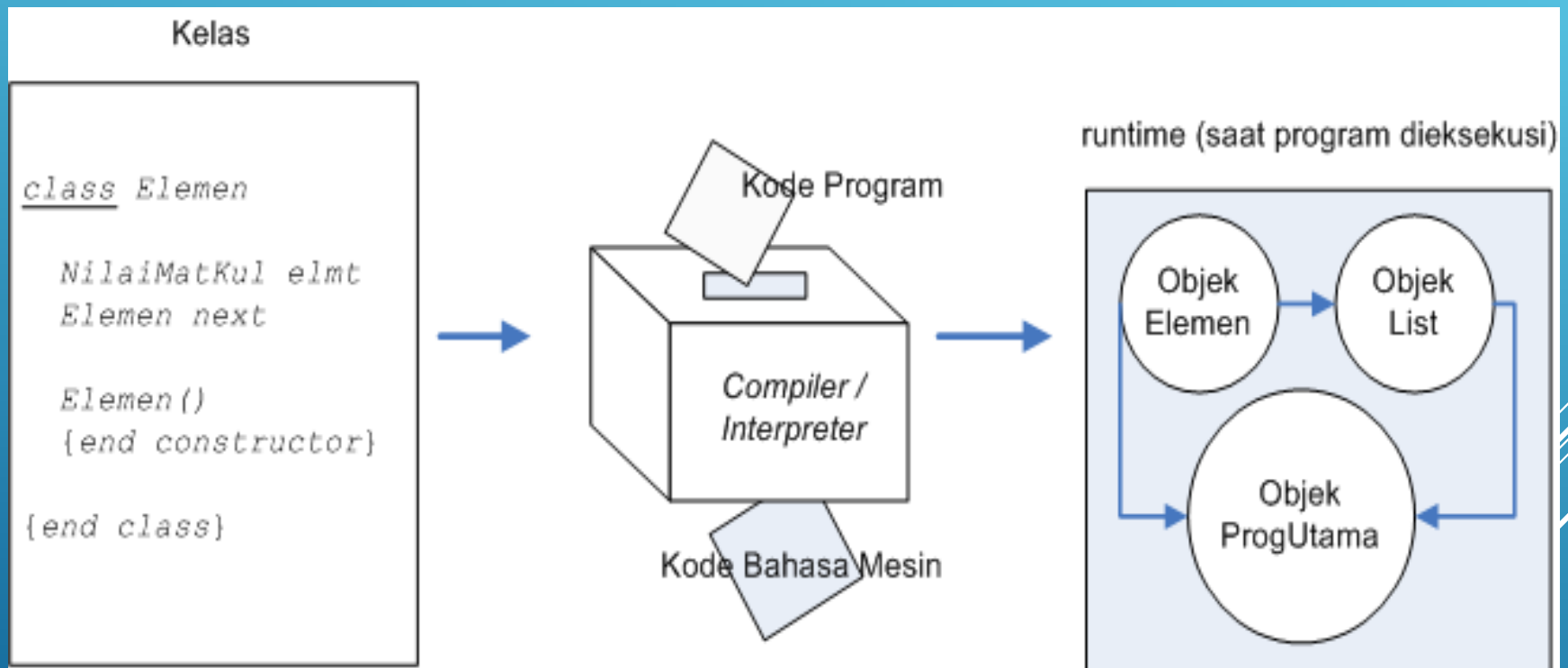


# OBJEK

Objek adalah abstraksi dan sesuatu yang mewakili dunia nyata seperti benda, manusia, satuan organisasi, tempat, kejadian, struktur, status, atau hal-hal lain yang bersifat abstrak. Objek merupakan suatu entitas yang mampu menyimpan informasi (status) dan mempunyai operasi (kelakuan) yang dapat diterapkan atau dapat berpengaruh pada status objeknya. Objek mempunyai siklus hidup yaitu diciptakan, dimanipulasi, dan dihancurkan.

Secara teknis, sebuah kelas saat program dieksekusi maka akan dibuat sebuah objek.

# ILUSTRASI KELAS DAN OBYEK



## 5. Enkapsulasi

Enkapsulasi dapat dianggap sebagai sebuah bungkus. Enkapsulasi inilah yang diimplementasikan dalam sebuah kelas dimana di dalam sebuah kelas terdiri dari atribut dan metode yang dibungkus dalam suatu kelas. Enkapsulasi pada sebuah kelas bertujuan untuk melindungi atribut dan metode-metode yang ada di dalam kelas agar tidak sembarangan diakses oleh kelas lain.

## 6. Atribut

Atribut dari sebuah kelas adalah variabel global yang dimiliki sebuah kelas. Atribut dapat berupa nilai atau elemen-elemen data yang dimiliki oleh objek dalam kelas objek. Atribut dipunyai secara individual oleh sebuah objek, misalnya kode, nama, berat, jenis dan sebagainya.

Operasi atau metode atau *method* pada sebuah kelas atau *class* merupakan dengan fungsi atau prosedur pada metodologi struktural. Sebuah kelas boleh memiliki lebih dari satu metode atau operasi. Operasi atau metode merupakan fungsi atau transformasi yang dapat dilakukan terhadap objek atau dilakukan oleh objek. Metode atau operasi dapat berasal dari

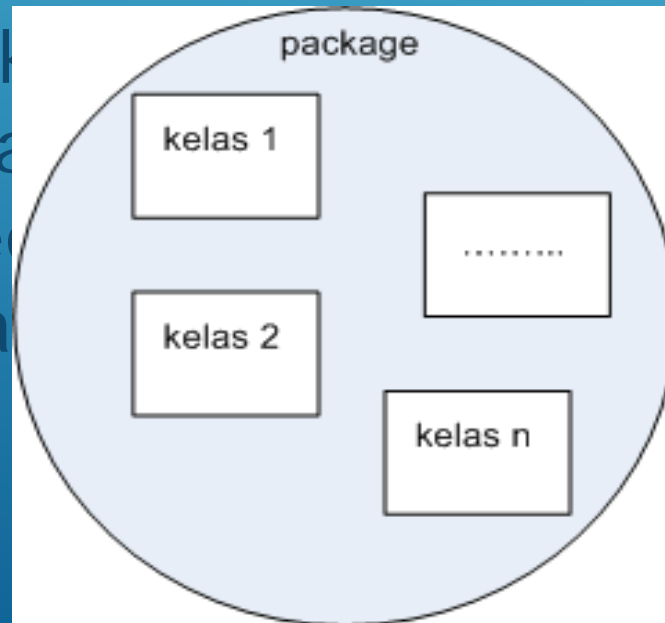
- ▶ event
- ▶ aktivitas atau aksi keadaan
- ▶ fungsi
- ▶ kelakuan dunia nyata

Contoh metode atau operasi misalnya Read



## 8. Package

*Package* adalah sebuah kontainer atau kemasan yang dapat digunakan untuk mengelompokkan kelas-kelas sehingga memungkinkan kelas yang bernama sama dalam *package* yang berbeda-beda dapat dilihat. Berikut ini adalah diagram yang menunjukkan sebuah *package* yang mengandung beberapa kelas.



Antarmuka atau *interface* sangat mirip dengan kelas, tapi tanpa atribut kelas dan memiliki metode yang dideklarasikan tanpa isi. Deklarasi metode pada sebuah interface dapat diimplementasikan oleh kelas lain. Sebuah kelas dapat mengimplementasikan lebih dari satu antarmuka dimana kelas ini akan mendeklarasikan metode pada antarmuka yang dibutuhkan oleh kelas itu sekaligus mendefinisikan isinya pada kode program kelas itu. Metode pada antarmuka yang diimplementasikan pada suatu kelas harus sama persis dengan yang ada pada antarmuka, misalnya pada antarmuka terdapat deklarasi metode `printAnimal()` maka pada kelas yang

Pendekatan berorientasi objek dengan pendekatan terstruktur dapat dilihat pada gambar berikut:

