



Kuliah  
*Rekayasa Perangkat Lunak*  
*(Software Engineering)*  
Bagian 2

*Software Engineering: A Practitioner's Approach, 6/e*

**Bab 5**  
**Praktek : Sebuah Pandangan Umum**

# Praktek Pemodelan

- Kita membuat model untuk mendapatkan pemahaman yang lebih baik terhadap entitas aktual yang akan dibangun
- *Model Analisis* menampilkan kebutuhan konsumen dengan melukiskan PL dalam tiga domain yang berbeda : domain informasi, domain fungsi, dan domain perilaku.
- *Model Desain* menampilkan karakteristik PL yang membantu praktisi untuk mengkonstruksinya secara efektif : arsitektur, antarmuka, detail level komponen.

# Praktek Pemodelan Analisis

- Prinsip-prinsip pemodelan analisis
  - Menampilkan domain informasi
  - Menampilkan fungsi PL
  - Menampilkan perilaku PL
  - Partisi dari tiga representasi ini
  - Bergerak dari esensi menuju implementasi
- Elemen-elemen model analisis (Bab 8)
  - Data model
  - Flow model
  - Class model
  - Behavior model

# Praktek Pemodelan Desain

- Prinsip-prinsip :
  - Desain harus dapat dilacak dari model analisis
  - Senantiasa memahami arsitektur
  - Fokus pada desain data
  - Antarmuka (pengguna maupun internal) harus didesain
  - Komponen harus menunjukkan independensi fungsional
  - Komponen-komponen harus “loosely coupled”
  - Representasi desain harus mudah dipahami
  - Model desain harus dikembangkan secara iteratif
- Elemen-elemen model desain
  - Data design
  - Architectural design
  - Component design
  - Interface design



# Praktek Konstruksi

- Prinsip Persiapan : Sebelum anda menulis satu baris kode, pastikan bahwa anda :
  - Memahami permasalahan yang anda coba selesaikan (lihat komunikasi dan pemodelan)
  - Memahami prinsip dan konsep desain dasar.
  - Mengambil bahasa pemrograman yang memenuhi kebutuhan PL untuk dibangun dan lingkungan dimana dia beroperasi.
  - Pilih lingkungan pemrograman yang menyediakan tool untuk memudahkan pekerjaan anda.
  - Buat sejumlah tes unit yang akan dilakukan ketika kode komponen sudah lengkap.



# Praktek Konstruksi

- Prinsip-prinsip coding: *ketika mulai menulis program, pastikan anda :*
  - Batasi algoritma anda dengan mengikuti ketentuan pemrograman terstruktur.
  - Pilih struktur data yang memenuhi kebutuhan desain.
  - Pahami arsitektur PL dan buat antarmuka yang konsisten dengannya.
  - Jaga logika kondisional sesederhana mungkin.
  - Buat perulangan bersarang dg cara yang membuatnya mudah untuk diuji.
  - Pilih nama-nama variabel yang bermakna, dan ikuti standar lokal yang lain.
  - Tulislah kode yang self-documenting.
  - Buatlah layout visual (indent, baris kosong) yang mempengaruhi pemahaman.



# Praktek Konstruksi

- Prinsip-prinsip validasi : *Setelah anda melengkapai kode pertama anda, pastikan anda :*
  - Melakukan pelacakan kode ketika dimungkinkan.
  - Melakukan tes unit dan memperbaiki kesalahan yang anda temukan.
  - Refactor kode program.



# Praktek Konstruksi

- Prinsip-prinsip Pengujian
  - Semua tes harus bisa dilacak dari requirement
  - Pengujian harus bisa direncanakan
  - Menggunakan prinsip pareto
  - Pengujian mulai dari “kecil” dan bergerak ke “besar”
  - Pengujian yang melelahkan tidak mungkin

# Praktek Deployment

- Prinsip-prinsip :
  - Kelola harapan pengguna pada setiap tahap
  - Paket penyajian lengkap harus disusun terpadu dan teruji
  - Tim pendukung harus disediakan
  - Materi pelatihan harus disediakan pada pengguna akhir
  - PL yang buggy, diperbaiki dulu, baru disajikan