

# Proses Pengembangan Perangkat Lunak (1)

Team Teaching Mata Kuliah Rekayasa Perangkat Lunak  
Jurusan Teknologi Informasi  
Politeknik Negeri Malang

# Outline

- Proses Pengembangan Perangkat Lunak
- Model Proses Pengembangan Perangkat Lunak
- Contoh Model Proses Pengembangan Perangkat Lunak

# Tujuan

- Memahami konsep proses pengembangan perangkat lunak dan permodelannya
- Memahami proses pengembangan perangkat lunak beserta penerapannya
- Memahami dasar aktivitas dalam proses pengembangan perangkat lunak

# Proses Pengembangan Perangkat Lunak

- Definisi:
  - Rangkaian aktivitas dalam rangka memproduksi produk perangkat lunak
- Istilah lain: **Software Development Life Cycle (SDLC)**

# Proses Pengembangan Perangkat Lunak



# Proses Pengembangan Perangkat Lunak

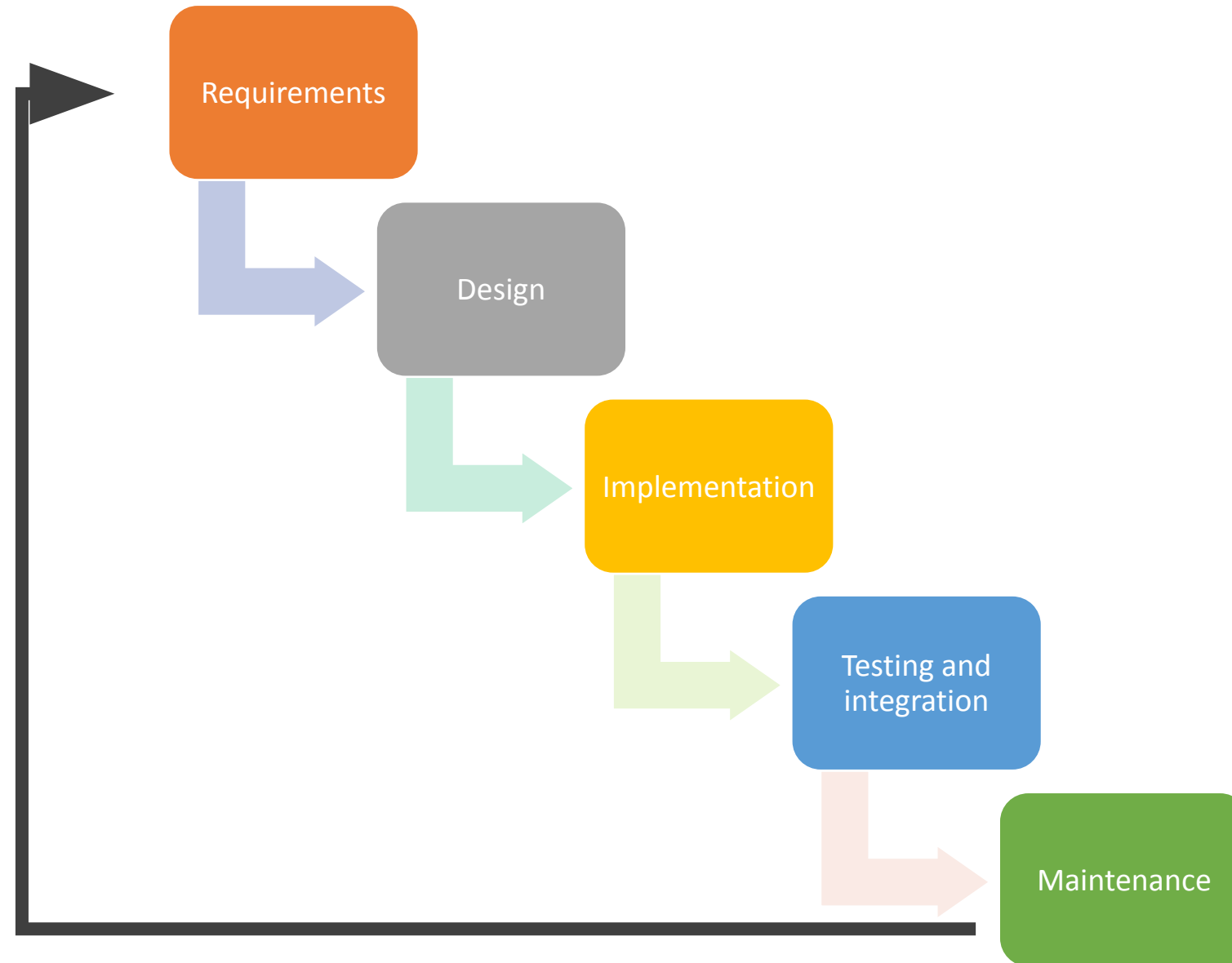
- *Plan-driven*
  - Semua aktivitas pengembangan terstruktur dan terukur sesuai alur
- *Agile*
  - Lebih adaptif terhadap perubahan kebutuhan pengguna

# Model Proses Pengembangan Perangkat Lunak

- Permodelan aktivitas pengembangan perangkat lunak
- Setiap model proses merepresentasikan proses secara **umum**
- Dapat digunakan sebagai *framework* yang dikembangkan menjadi proses pengembangan perangkat lunak yang spesifik
- Model proses:
  - *Waterfall*
  - *Incremental development*
  - *Reuse-oriented software engineering*

# Waterfal

- Dibuat tahun 1970 oleh Winston W. Royce
- Setiap proses harus dilakukan sebelum menuju proses selanjutnya

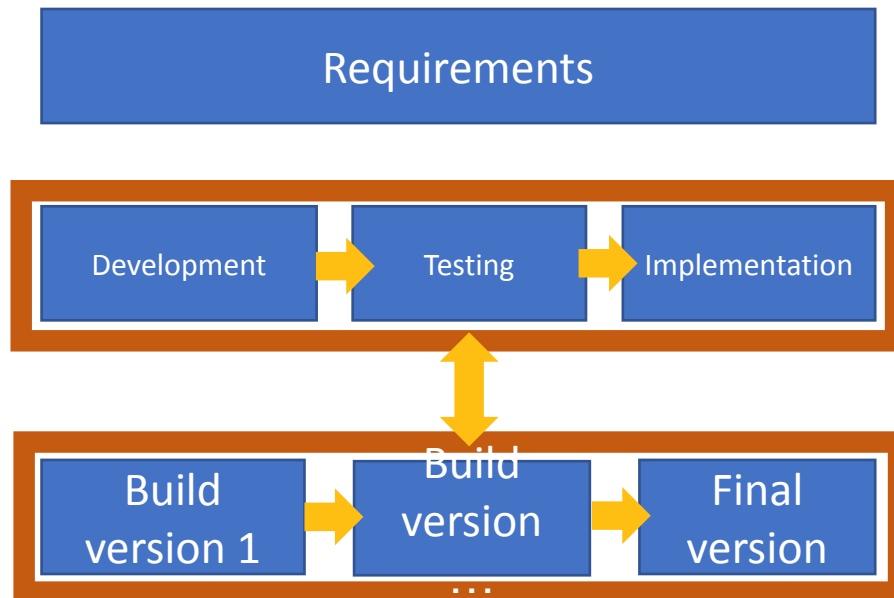




# *Waterfall*

- Kelebihan:
  - Setiap fase terdapat dokumentasi
  - Mudah untuk dimonitor oleh seorang project manager
- Kekurangan:
  - Kurang fleksibel, spesifikasi kebutuhan di awal harus 'sudah pasti'

# *Incremental development*



- Proses pengembangan perangkat lunak yang dibagi menjadi beberapa versi hingga perangkat lunak menjadi satu produk yang utuh
- Menjadi dasar pendekatan Agile

# *Incremental development*

- Kelebihan:
  - Dapat menekan biaya atas perubahan kebutuhan perangkat lunak
  - User dapat memberikan feedback pada setiap versi perangkat lunak
  - Perangkat lunak dapat dikembangkan secara cepat
- Kekurangan:
  - Lebih sulit untuk memonitor proses karena tidak ada dokumen yang *rigid* pada setiap prosesnya
  - Memungkinkan perubahan yang masif pada versi perangkat lunak selanjutnya, sehingga waktu dan biaya pengembangan dapat meningkat

# *Reuse-oriented Software Engineering*

- Menggunakan pendekatan aspek *reusable* pada perangkat lunak
- Istilah lain: **Component Based Software Engineering**
- Komponen pada perangkat lunak yang telah ada, digunakan kembali sesuai kebutuhan



# *Reuse-oriented Software Engineering*

- Kelebihan:
  - Dapat mengurangi biaya pembuatan modul perangkat lunak dengan model yang sama
  - Waktu pengembangan bisa jadi lebih cepat
- Kekurangan:
  - Harus dilakukan kompromi antara kebutuhan pengguna dengan produk yang ada, jika tidak, memungkinkan ketidaksesuaian produk yang eksisting dengan kebutuhan pengguna



Any questions?