



**RPL – Pertemuan 1**

PENGANTAR

# **REKAYASA PERANGKAT LUNAK**

Sub-CPMK: Menjelaskan konsep dasar RPL dan menganalisis proses bisnis & kebutuhan SI

# Tujuan Pembelajaran

- Menjelaskan definisi, ruang lingkup, dan tujuan **Rekayasa Perangkat Lunak (RPL)**.
- Membedakan RPL dengan pemrograman (coding) murni dalam konteks proses dan kualitas.
- Mengidentifikasi proses bisnis dan kebutuhan awal sistem informasi.
- Memahami aktivitas generik proses perangkat lunak menurut Sommerville.



# Apa itu Perangkat Lunak?



Sekumpulan instruksi/program yang memberi tahu komputer cara bekerja.



Tidak berwujud (intangible), berbeda dengan perangkat keras (hardware).



Contoh: Sistem Operasi (Windows, Linux), Aplikasi (Word, WhatsApp), Tools Pemrograman (IDE, Compiler).

# Apa itu Rekayasa Perangkat Lunak?



Disiplin ilmu yang mempelajari **cara membuat perangkat lunak** secara **terstruktur, terukur, dan terkontrol**.



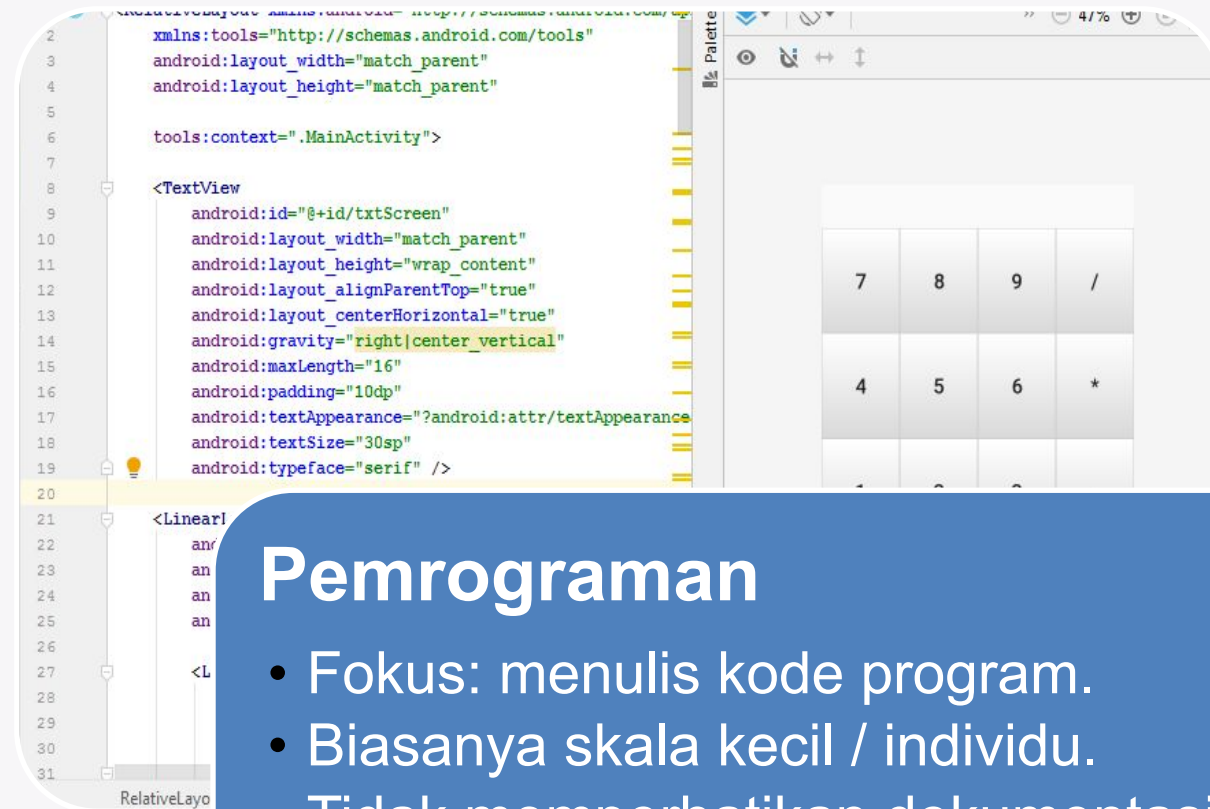
Bukan hanya menulis kode, tapi mencakup **analisis, desain, implementasi, pengujian, dan pemeliharaan**.



Tujuannya: menghasilkan perangkat lunak yang **berkualitas, sesuai kebutuhan, tepat waktu, dan efisien biaya**.

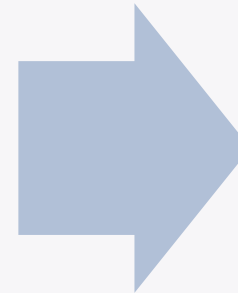


# RPL vs Pemograman



## Pemrograman

- Fokus: menulis kode program.
- Biasanya skala kecil / individu.
- Tidak memperhatikan dokumentasi, manajemen proyek, dan kualitas secara menyeluruh.
- Contoh: membuat aplikasi kalkulator sederhana.

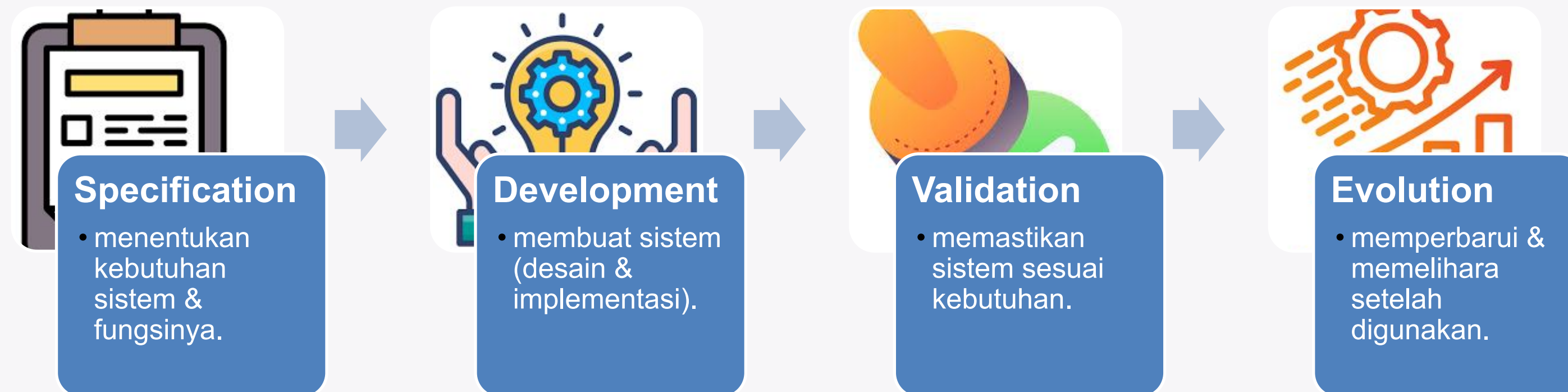


## RPL (Rekayasa Perangkat Lunak)

- Fokus: seluruh proses pengembangan perangkat lunak (analisis → desain → implementasi → uji → pemeliharaan).
- Skala besar, biasanya melibatkan tim.
- Ada manajemen proyek, dokumentasi, standar kualitas.
- Contoh: membangun sistem informasi akademik di kampus.

# Empat Aktivitas Generik Proses Perangkat Lunak (Sommerville)

Menurut Sommerville, ada **4 aktivitas utama** dalam proses perangkat lunak. Aktivitas ini **dapat diulang (iteratif)** sesuai kebutuhan proyek:



# Kualitas Produk VS Kualitas Proses

## Kualitas Produk

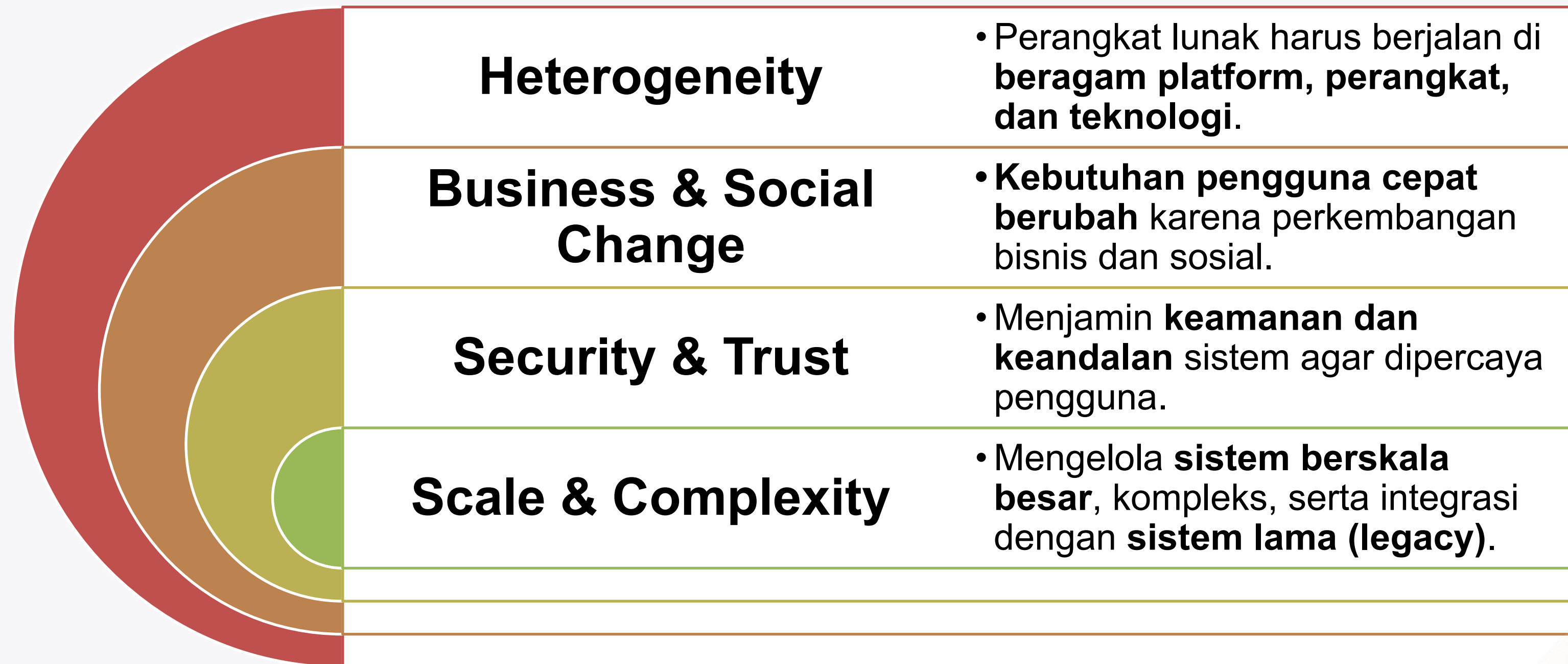
- Fokus pada **hasil akhir perangkat lunak**.
- Ukuran: fungsionalitas, keandalan, kinerja, kegunaan, keamanan.
- Contoh: aplikasi absensi berjalan cepat, tanpa error, mudah digunakan.

## Kualitas Proses

- Fokus pada **cara perangkat lunak dikembangkan**.
- Ukuran: kepatuhan pada standar, dokumentasi, manajemen proyek, metode pengujian.
- Proses yang baik → peluang menghasilkan produk berkualitas lebih besar.

Menurut Sommerville, produk berkualitas biasanya lahir dari proses yang berkualitas, meskipun keduanya bisa diukur secara terpisah.

# Tantangan Utama Rekayasa Perangkat Lunak (Sommerville)





# Etika Profesional & Tanggung Jawab



## Kepentingan Publik

- Menjaga **keselamatan, privasi, dan dampak sosial** dari sistem.



## Kompetensi

- Hanya mengerjakan pekerjaan sesuai **keahlian & kemampuan**.



## Kerahasiaan & Kepatuhan

- Menjaga **rahasia organisasi**, serta taat pada hukum & standar.



## Integritas

- Bersikap **jujur** tentang batasan, risiko, dan status proyek.

# Contoh Pemangku Kepentingan Sistem



# Gambar: Diagram Konteks (Studi Kasus Pendaftaran)



Diagram konteks sederhana untuk menggambarkan batas sistem dan entitas eksternal.

# Persyaratan Sistem (Preview)



## Fungsional:

- Fitur yang harus ada di sistem,
- misalnya:
  - Pendaftaran mahasiswa
  - Verifikasi data
  - Pembayaran biaya kuliah



## Non-fungsional:

- Menjelaskan **kualitas** atau **batasan** sistem.  
Contoh:
- **Keamanan**: data mahasiswa harus terlindungi
- **Performa**: sistem merespons < 2 detik
- **Keandalan**: tersedia 24/7
- **Kegunaan (Usability)**: mudah digunakan mahasiswa dan admin
- **Regulasi**: sesuai aturan akademik kampus



## Artefak awal:

- Dokumen awal yang menjadi dasar pengembangan sistem.  
Berisi:
- **Daftar kebutuhan awal** (fitur dan kualitas)
- **Prioritas** (penting, sedang, rendah)
- **Risiko awal** (contoh: keterlambatan pembayaran, kegagalan server)



# Artefak & Dokumentasi Awal

## Dokumen Visi & Ruang Lingkup

- Menjelaskan tujuan sistem dan batasannya.
- Menetapkan apa yang termasuk dan tidak termasuk dalam sistem.

## Daftar Pemangku Kepentingan

- Identifikasi semua aktor atau pihak yang berinteraksi dengan sistem.
- Contoh: pengguna akhir, admin, sistem eksternal.

## Model Awal

- **Diagram Konteks:** menunjukkan sistem sebagai satu entitas dan interaksi dengan aktor eksternal.
- **Daftar Use Case Tingkat Tinggi:** deskripsi singkat fitur utama dari perspektif pengguna.

## Rencana Kegiatan Awal & Estimasi

- Rencana kegiatan pengembangan secara high-level.
- Estimasi waktu dan sumber daya secara kasar.

# Praktik Baik pada Fase Awal





# Ringkasan

RPL berfokus pada produksi perangkat lunak yang bernilai, berkualitas, dan berkelanjutan.

Aktivitas proses generik: specification, development, validation, evolution.

Perbedaan RPL vs pemrograman: cakupan, dokumentasi, kualitas, peran.

Preview: kebutuhan fungsional/non-fungsional dan pemodelan awal.

# Diskusi & Tugas

## Diskusi :

Diskusi: Identifikasi pemangku kepentingan dan kebutuhan awal untuk sistem pendaftaran kampus.

## Tugas :

Tuliskan 5 kebutuhan fungsional & 5 non-fungsional + diagram konteks sederhana.

## Bacaan :

Sommerville, Software Engineering (10th ed.), Bab 1.



# Sumber (Referensi)

Sommerville, I. (2016). Software Engineering (10th ed., Global Edition).  
Pearson.

**Catatan:** materi pada pertemuan ini merupakan ringkasan dan parafrasa konsep utama Bab 1 + pengantar bab terkait.