



REKAYASA PERANGKAT LUNAK I

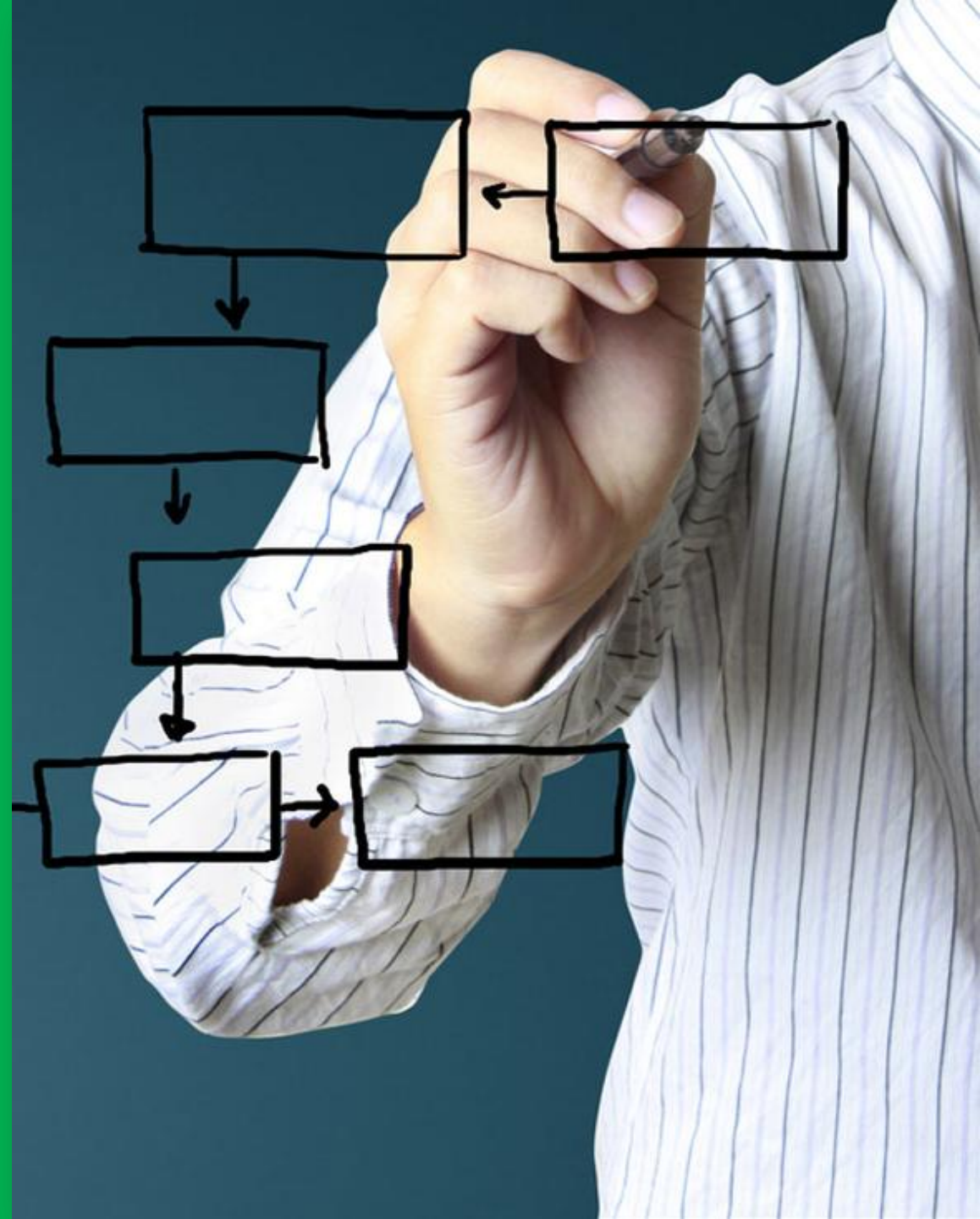
Pendahuluan Rekayasa Perangkat Lunak

Disusun Oleh:

Adam Mukharil Bachtiar

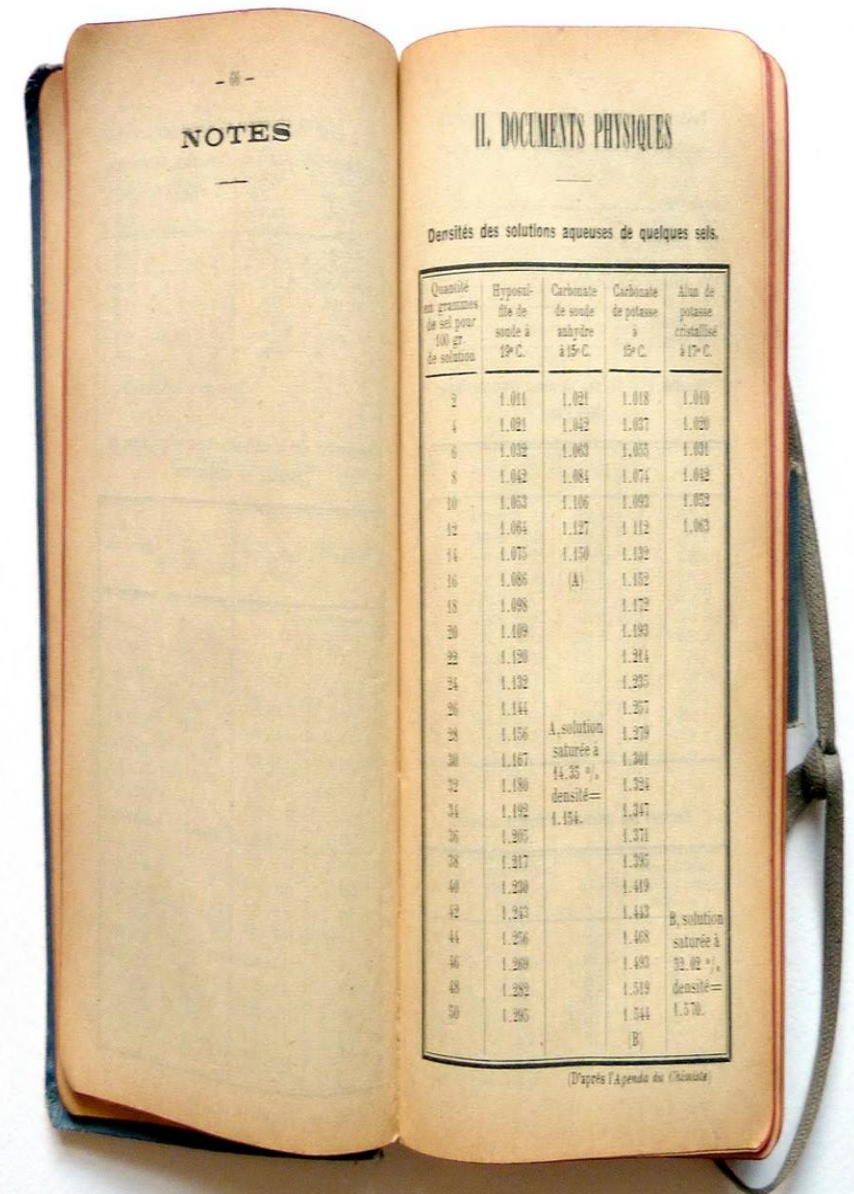
Teknik Informatika UNIKOM

adfbipotter@gmail.com



AGENDA PERKULIAHAN

- ➡ Deskripsi dan silabus mata kuliah
- ➡ Penilaian dan Referensi
- ➡ Pendahuluan Rekayasa Perangkat Lunak



Deskripsi dan Silabus Mata Kuliah

DESKRIPSI MATA KULIAH

➡ Sifat : Wajib

➡ Prasyarat : Algoritma dan Pemrograman, Struktur Data, Basis

Data, IMK

➡ Waktu : 3 SKS

SILABUS MATA KULIAH

- ➡ Pendahuluan Rekayasa Perangkat Lunak
- ➡ Proses Pembangunan Perangkat Lunak
- ➡ Requirement Engineering
- ➡ Analisis Perangkat Lunak
- ➡ Perancangan Perangkat Lunak
- ➡ Pengujian Perangkat Lunak
- ➡ Pemeliharaan Perangkat Lunak



Pendekatan Analisis Terstruktur

Penilaian dan Referensi

PENILAIAN

Indeks	Nilai
A	$80 \leq NA \leq 100$
B	$68 \leq NA \leq 79$
C	$56 \leq NA \leq 67$
D	$45 \leq NA \leq 55$
E	$0 \leq NA \leq 44$

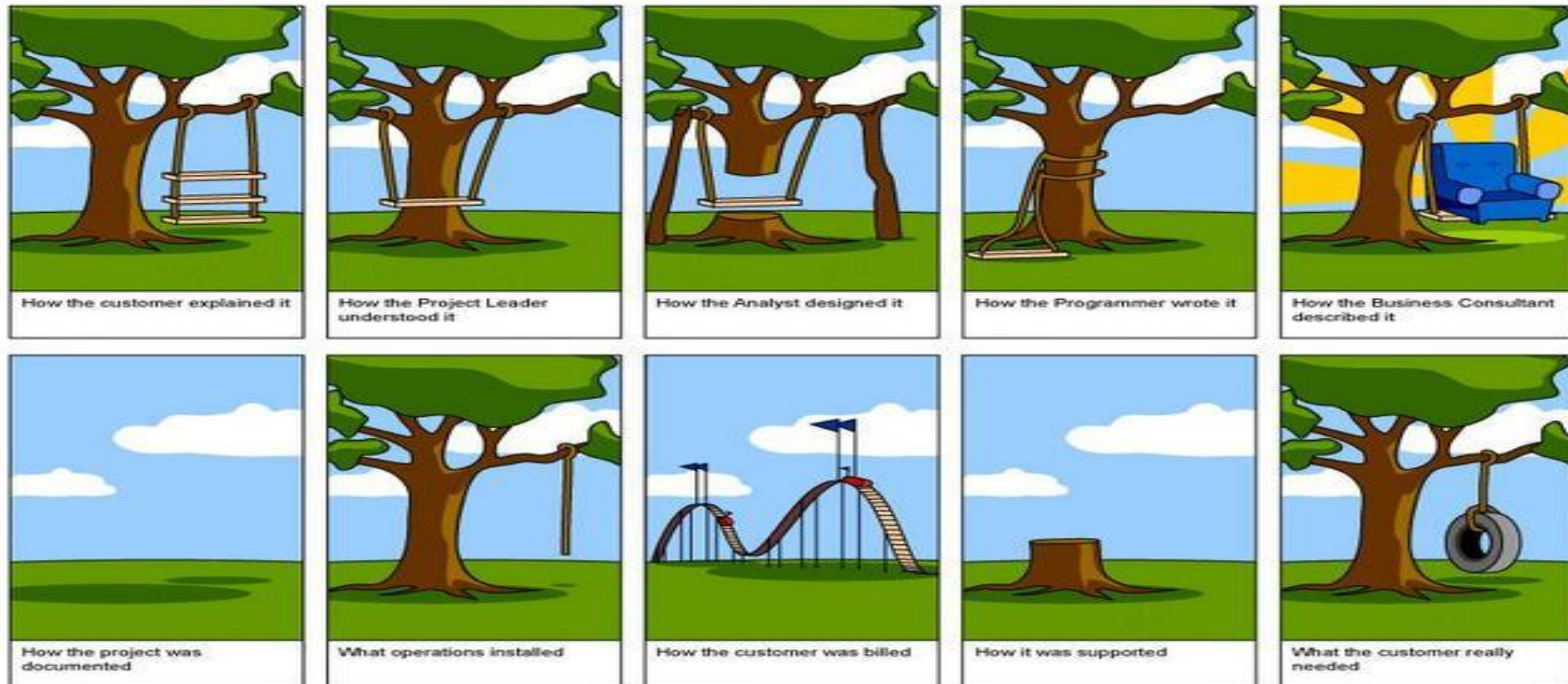
20% TUGAS + **35%** UTS + **45%** UAS

REFERENSI

- ➡ Roger S. Pressman, Software Engineering: A Practitioner's Approach, 7th edition.
- ➡ Ian Sommerville, Software Engineering, 8th edition.
- ➡ Kendall, System Analysis and Design, 8th edition.
- ➡ Slide perkuliahan.

Pendahuluan Rekayasa Perangkat Lunak

KENAPA HARUS MEMPELAJARI RPL



DEFINISI PERANGKAT LUNAK

“Program komputer dan dokumentasi (user manual, dokumen kebutuhan, dokumen perancangan) yang berhubungan dengan program tersebut”

JENIS PERANGKAT LUNAK

❶ Generic/Public Software

“Pengguna umum, kebutuhan luas, lingkungan pengguna tidak terbatas “

❷ Custom/Private Software

“Pengguna khusus, kebutuhan sempit, lingkungan pengguna terbatas”

CONTOH PUBLIC SOFTWARE



DEFINISI REKAYASA PERANGKAT LUNAK

“Disiplin ilmu rekayasa atau teknik yang berkaitan dengan semua aspek dalam membuat perangkat lunak”

PROSES PERANGKAT LUNAK

“Serangkaian aktifitas
yang tujuannya adalah
pembangunan atau evolusi
perangkat lunak”



KRITERIA PERANGKAT LUNAK YANG BAIK

① Maintainability

Software must evolve to meet changing needs

② Dependability

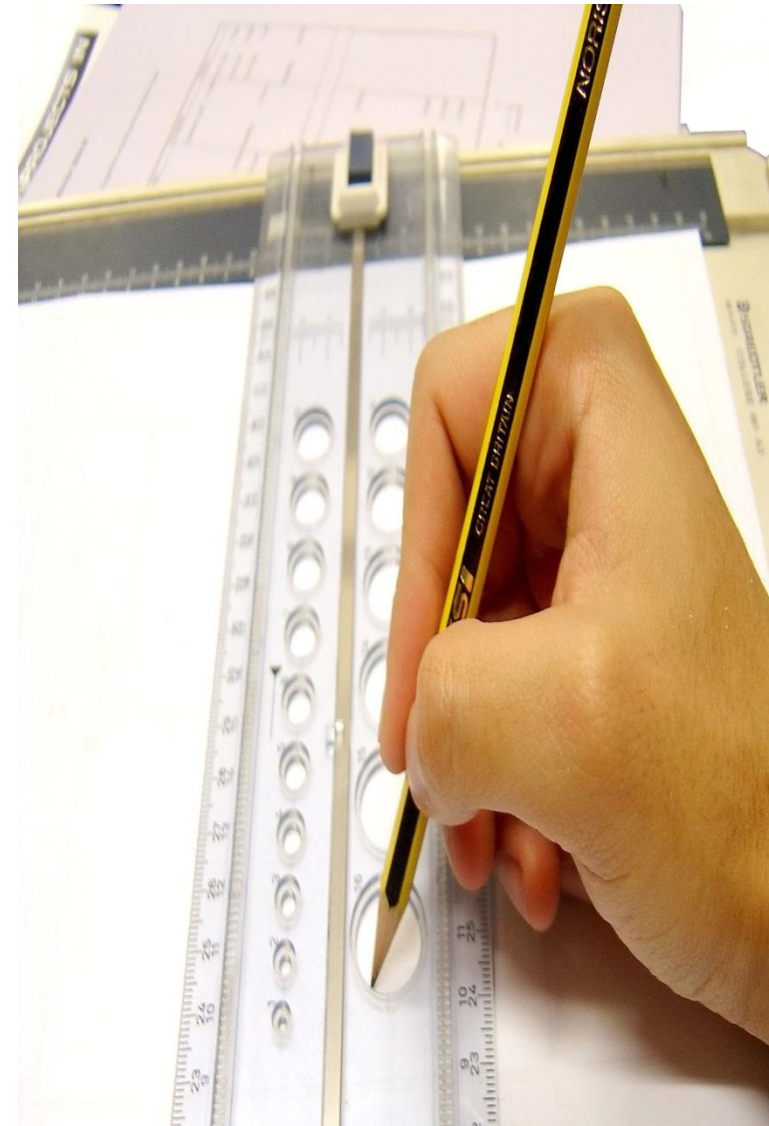
Software must be trustworthy

③ Efficiency

Software should not make wasteful use of system resources

④ Usability

Software must be usable by the users for which it was designed



SOFTWARE TOOLS

① Upper-CASE

Tools untuk mendukung aktifitas proses awal dari requirement dan desain

② Lower-CASE

Tools untuk mendukung aktifitas selanjutnya seperti programming, debugging, dan testing.



Terima Kasih