



Proses Software

Arna Fariza
PENS-ITS

1

Rekayasa Perangkat Lunak




Proses Software


- ❑ Sekumpulan aktifitas yang saling terkait untuk spesifikasi, desain, implementasi dan testing sistem software

2

Rekayasa Perangkat Lunak




Tujuan




- ☐ Memperkenalkan model proses software
- ☐ Menggambarkan beberapa model proses dan kapan digunakan
- ☐ Menggambarkan outline model proses untuk rekayasa kebutuhan, pengembangan software, testing dan evolusi
- ☐ Mengenalkan teknologi CASE untuk mendukung aktifitas proses software

3

Rekayasa Perangkat Lunak





Topik



- ☐ Model proses software
- ☐ Iterasi proses
- ☐ Spesifikasi software
- ☐ Desain dan implementasi software
- ☐ Validasi software
- ☐ Evolusi software
- ☐ Automated process support

4

Rekayasa Perangkat Lunak

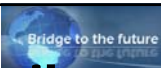



Proses Software

- ❑ Sekumpulan aktifitas terstruktur yang dibutuhkan untuk mengembangkan sistem software
 - o Spesifikasi
 - o Desain
 - o Validasi
 - o Evolusi
- ❑ Model proses software adalah representasi abstrak dari proses. Merupakan gambaran dari proses dari beberapa perspektif tertentu

5

Rekayasa Perangkat Lunak

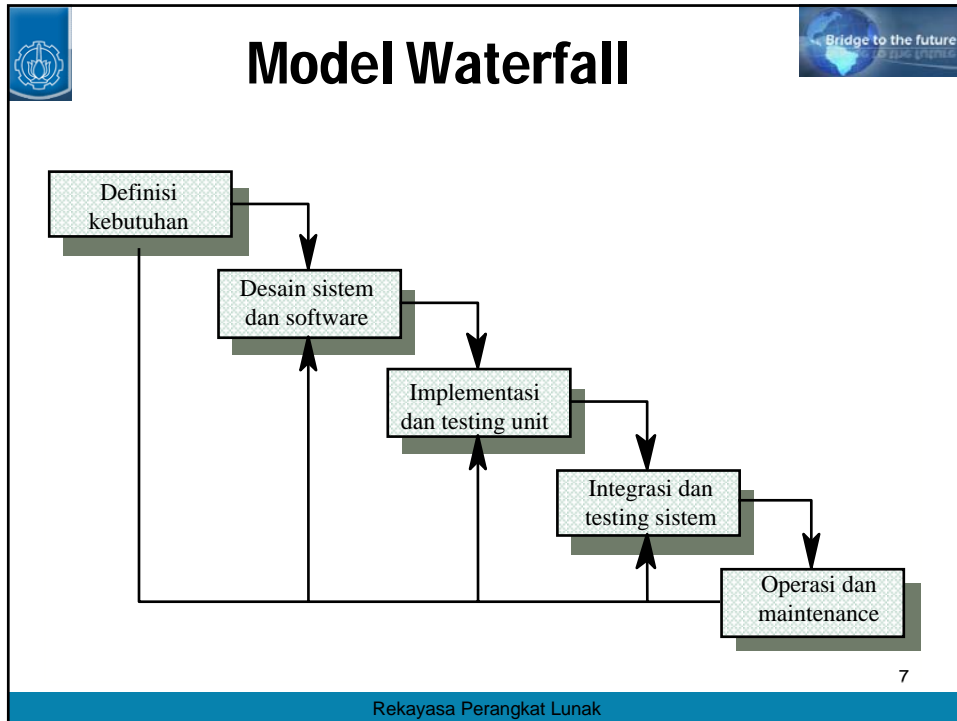


Model Proses Software Generik

- ❑ Model waterfall
 - o Membagi dan membedakan fase spesifikasi dan pengembangan
- ❑ Pengembangan Evolusioner
 - o Spesifikasi dan pengembangan terpisah
- ❑ Pengembangan sistem Formal
 - o Model sistem matematis yang secara formal diterjemahkan ke dalam implementasi
- ❑ Pengembangan Reuse-based
 - o Sistem dibangun dari komponen yang sudah ada

6

Rekayasa Perangkat Lunak



- Fase model Waterfall**
- ☐ Analisa dan definisi kebutuhan
 - ☐ Desain sistem dan software
 - ☐ Implementasi dan unit testing
 - ☐ Integrasi dan testing sistem
 - ☐ Operasi dan maintenance
 - ☐ Kekurangan dari model waterfall adalah kesulitan untuk mengakomodasi perubahan setelah proses berjalan
- 8
- Rekayasa Perangkat Lunak



Permasalahan Model Waterfall



- ☐ Tidak fleksibel dalam pembagian proyek ke dalam tingkat yang berbeda
- ☐ Sulit untuk merespon perubahan kebutuhan konsumen
- ☐ Sehingga model ini hanya cocok jika kebutuhan sudah dimengerti dengan baik

9

Rekayasa Perangkat Lunak



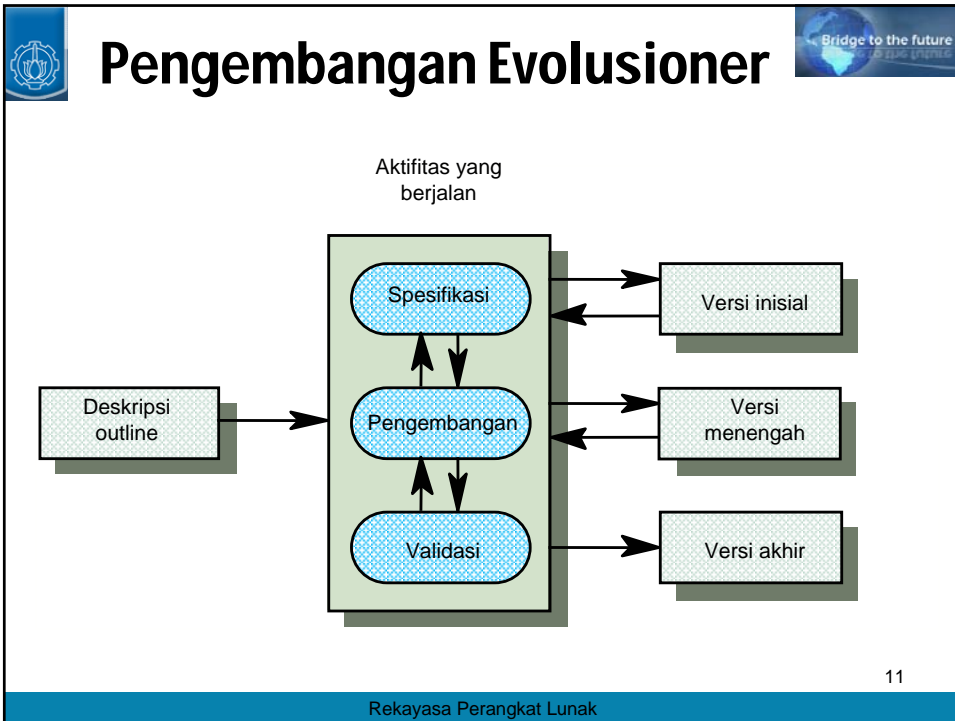
Pengembangan Evolusioner



- ☐ Pengembangan Exploratory
 - o Obyektif : bekerja dengan konsumen dan melibatkan sistem akhir dari spesifikasi skema inisial. Dimulai dengan kebutuhan yang dimengerti dengan baik
- ☐ Throw-away prototyping
 - o Obyektif : mengerti kebutuhan sistem. Dimulai dengan kebutuhan yang tidak dimengerti dengan baik

10

Rekayasa Perangkat Lunak



- Pengembangan Evolusioner**
- Bridge to the future
- ❑ Permasalahan
 - o Tidak ada visibilitas proses
 - o Sistem biasanya tidak terstruktur dengan baik
 - o Kemampuan khusus (misalnya bahasa untuk prototipe cepat) kemungkinan diperlukan
 - ❑ Aplikasi
 - o Untuk sistem interaktif berukuran kecil atau medium
 - o Untuk bagian dari sistem besar (misalnya user interface)
 - o Untuk sistem dengan daur hidup pendek
- 12
- Rekayasa Perangkat Lunak

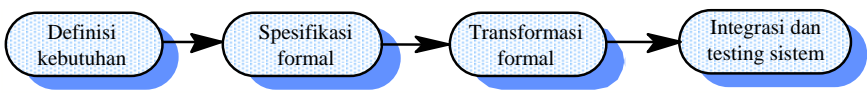
Pengembangan Sistem Formal

- ❑ Berbasis transformasi dari spesifikasi matematis melalui representasi yang berbeda untuk program yg dapat dieksekusi
- ❑ Transformasi adalah 'pemelihara kebenaran' sehingga dapat menunjukkan program sesuai spesifikasinya
- ❑ Merupakan pendekatan 'Cleanroom' untuk pengembangan software

13

Rekayasa Perangkat Lunak

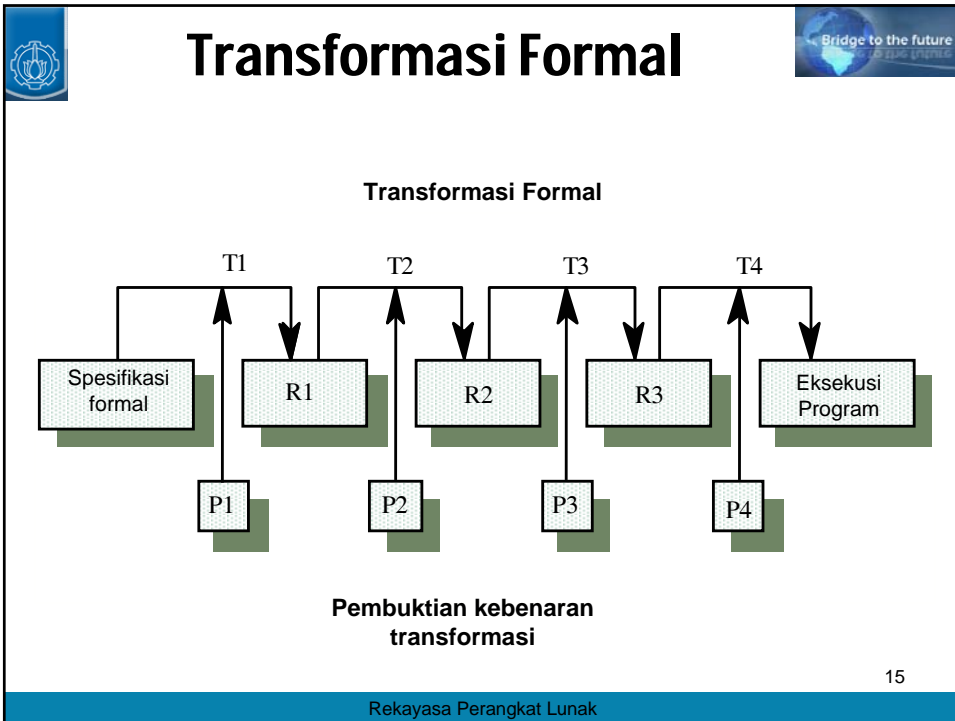
Pengembangan Sistem Formal



```
graph LR; A([Definisi kebutuhan]) --> B([Spesifikasi formal]); B --> C([Transformasi formal]); C --> D([Integrasi dan testing sistem]);
```

14

Rekayasa Perangkat Lunak



Pengembangan Sistem Formal

- ❑ Permasalahan
 - o Perlu kemampuan dan training khusus untuk mengaplikasikan teknik ini
 - o Secara formal sulit untuk menentukan beberapa aspek dari sistem seperti antarmuka user
- ❑ Aplikasi
 - o Sistem kritis terutama dimana keselamatan dan keamanan harus dibuat sebelum sistem beroperasi

16

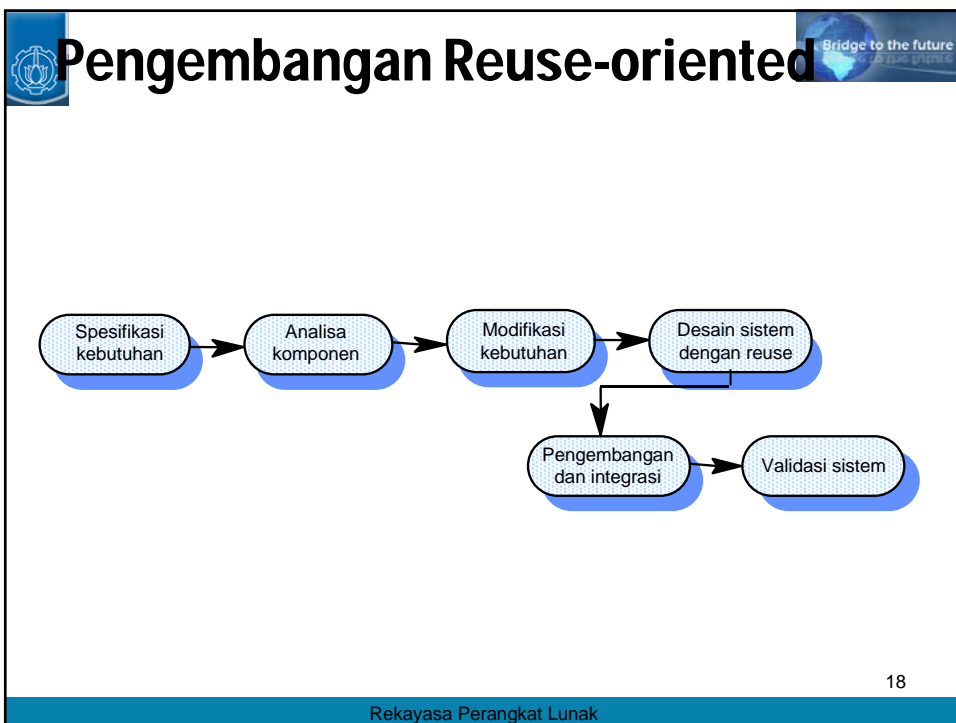
Rekayasa Perangkat Lunak


Pengembangan Reuse-oriented

- ❑ Berbasis systematic reuse dimana sistem diintegrasikan dalam komponen yang sudah ada atau sistem COTS (Commercial-off-the-shelf)
- ❑ Level Proses
 - o Analisa komponen
 - o Modifikasi kebutuhan
 - o Desain sistem dengan reuse
 - o Pengembangan dan integrasi
- ❑ Pendekatan ini menjadi lebih penting tetapi masih terbatas penggunaannya


17

Rekayasa Perangkat Lunak






Iterasi Proses



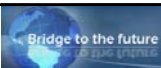
- ❑ Kebutuhan sistem SELALU berkembang selama proyek berlangsung sehingga iterasi proses dimana level sebelumnya dilakukan rework merupakan bagian dari proses untuk sistem yang besar
- ❑ Iterasi dapat diaplikasikan untuk semua model proses generik
- ❑ 2 pendekatan :
 - o Pengembangan incremental
 - o Pengembangan spiral

19

Rekayasa Perangkat Lunak



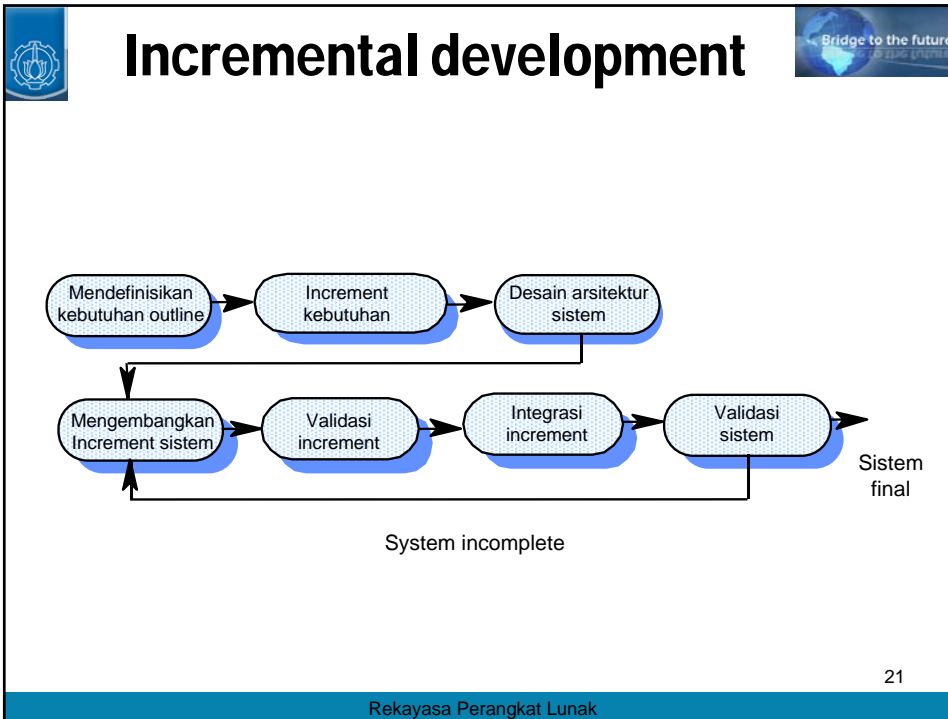
Pengembangan Incremental



- ❑ Pelepasan sistem tidak dalam bentuk pelepasan tunggal, tetapi pengembangan dan pelepasan dibagi ke dalam 'increment' dimana setiap 'increment' melepaskan bagian dari fungsional yang dibutuhkan
- ❑ Kebutuhan user diprioritaskan dan kebutuhan prioritas tertinggi dimasukkan dalam 'increment' awal
- ❑ Bila pengembangan 'increment' dimulai, kebutuhan dibekukan lebih dahulu dan setelah itu kebutuhan untuk 'increment' selanjutnya dapat dilanjutkan

20

Rekayasa Perangkat Lunak



- Keuntungan Pengembangan Incremental**
- ☐ Nilai konsumen dapat diserahkan pada setiap 'increment' sehingga fungsional sistem tersedia lebih dahulu
 - ☐ 'increment' awal berfungsi sebagai prototype untuk membantu memperoleh kebutuhan 'increment' selanjutnya
 - ☐ Resiko lebih rendah dari keseluruhan kegagalan proyek
 - ☐ Layanan sistem prioritas tertinggi cenderung menerima testing terbanyak
- 22
- Rekayasa Perangkat Lunak



Pemrograman Extreme



- ☐ Pendekatan baru untuk pengembangan berbasis pengembangan dan pelepasan fungsional 'increment' yang sangat kecil
- ☐ Mempercayakan perbaikan konstan, keterlibatan user dalam tim pengembang dan pemrograman

23

Rekayasa Perangkat Lunak



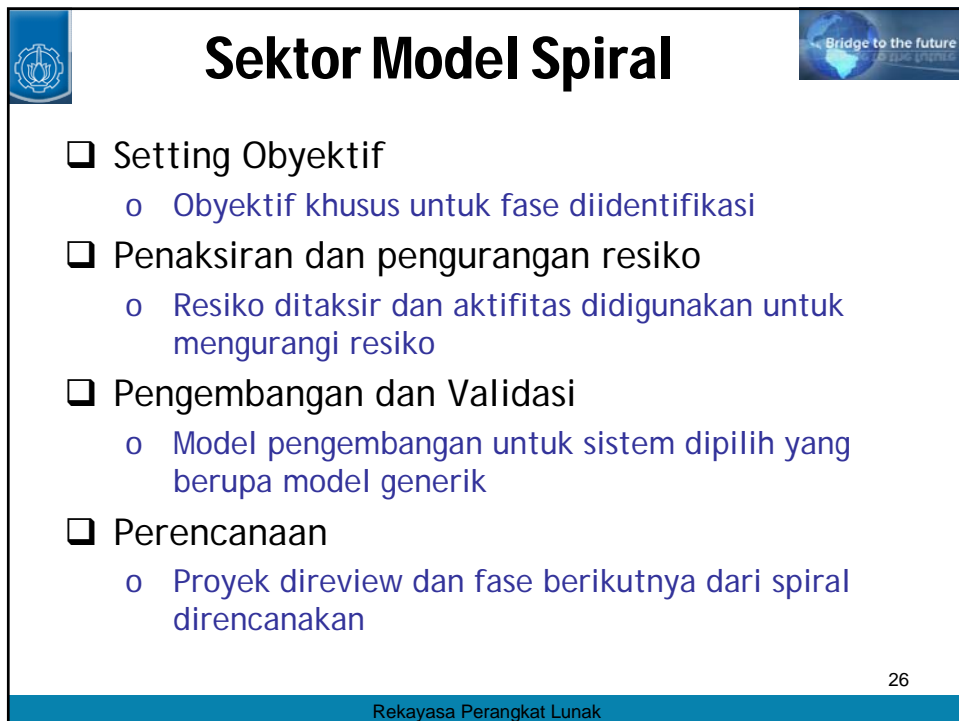
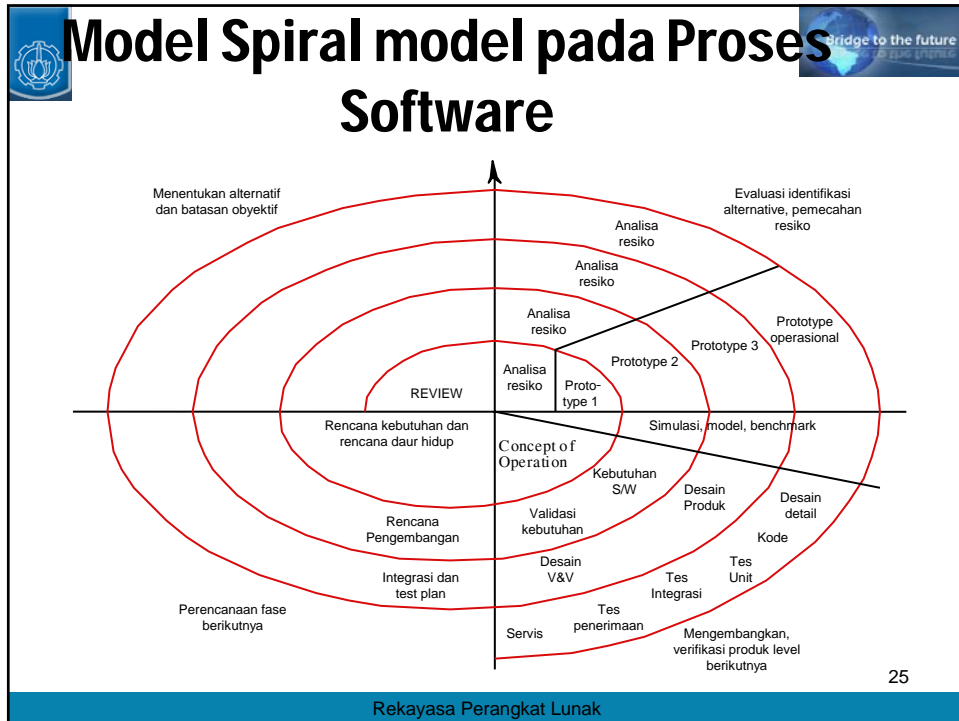
Pengembangan Spiral



- ☐ Proses direpresentasikan sebagai spiral bukan sebagai urutan aktivitas dengan melihat sistem sebelumnya (backtracking)
- ☐ Setiap loop dalam spiral merepresentasikan fase dalam proses
- ☐ Tidak ada fase yang tetap seperti spesifikasi atau desain- loop dalam spiral dipilih tergantung pada apa yang dibutuhkan
- ☐ Resiko ditaksir secara eksplisit dan penyelesaian sepanjang proses

24

Rekayasa Perangkat Lunak





Spesifikasi Software



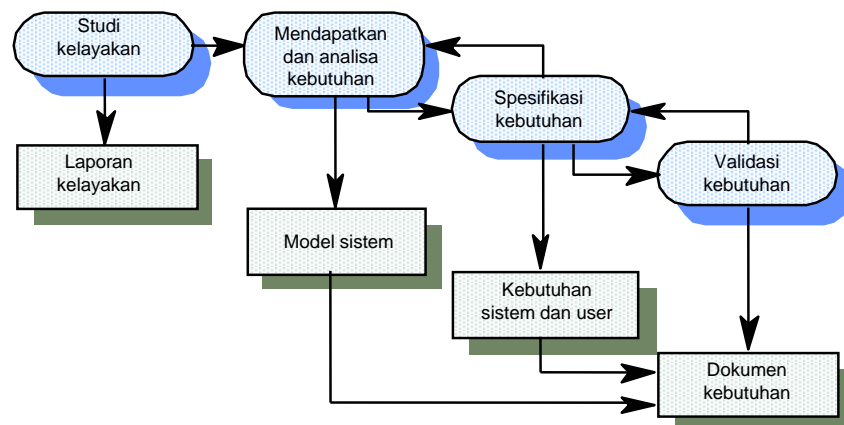
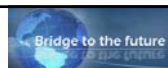
- ❑ Dari proses yang sudah berjalan ditentukan layanan apa yang dibutuhkan dan batasan operasi dan pengembangan sistem
- ❑ Proses rekayasa kebutuhan
 - o Studi kelayakan
 - o Perolehan dan analisa kebutuhan
 - o Spesifikasi kebutuhan
 - o Validasi kebutuhan

27

Rekayasa Perangkat Lunak



Proses Rekayasa Kebutuhan



28

Rekayasa Perangkat Lunak



Implementasi dan Desain Software



- ☐ Proses mengubah spesifikasi sistem menjadi sistem yang dijalankan
- ☐ Desain software
 - o Mendesain struktur software yang didapatkan dari spesifikasi
- ☐ Implementasi
 - o Mengubah struktur software ke dalam program yang dieksekusi
- ☐ Aktivitas desain dan implementasi saling berhubungan dan mungkin terpisah

29

Rekayasa Perangkat Lunak



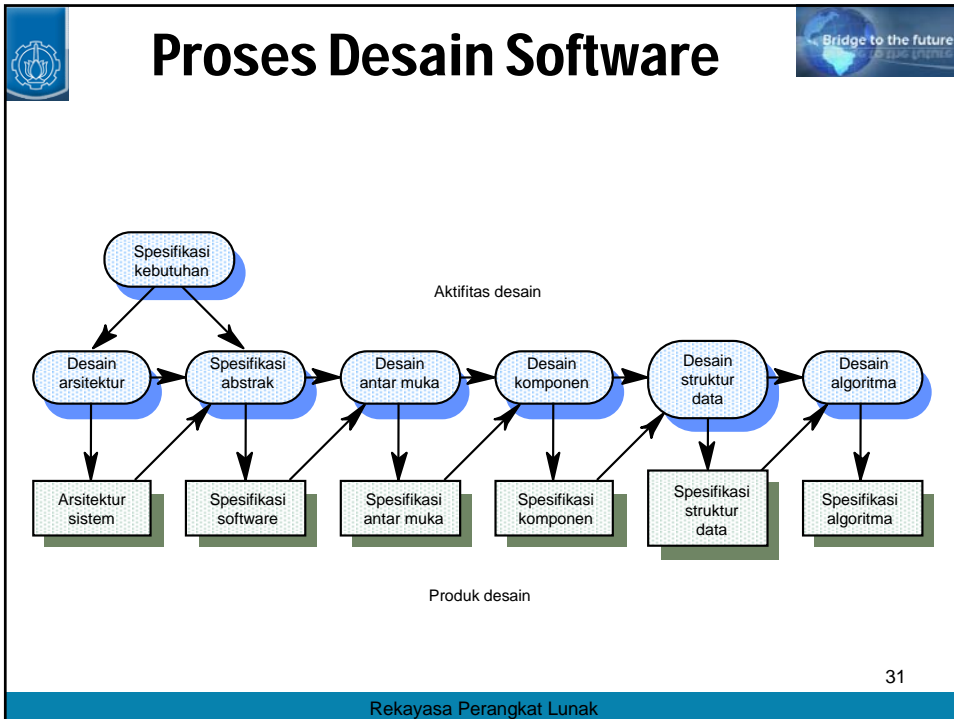
Aktivitas Proses Desain



- ☐ Desain arsitektur
- ☐ Spesifikasi abstrak
- ☐ Desain antar muka
- ☐ Desain komponen
- ☐ Desain struktur data
- ☐ Desain algoritma

30

Rekayasa Perangkat Lunak



- # Metode Desain
- ☐ Pendekatan sistematis untuk pengembangan desain software
 - ☐ Desain biasanya terdokumentasi sebagai kumpulan model grafis
 - ☐ Model yang mungkin
 - o Model data-flow
 - o Model entity-relation-attribute
 - o Model struktural
 - o Model obyek
- The slide number '32' and the text 'Rekayasa Perangkat Lunak' are located at the bottom right.



Pemrograman dan Debugging



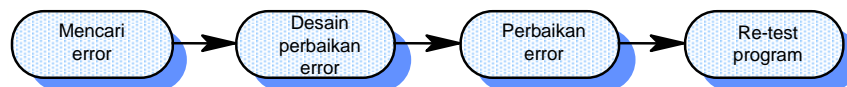
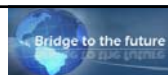
- ❑ Mengubah desain ke dalam program dan menghilangkan error dari program
- ❑ Pemrograman adalah aktifitas personal - tidak ada proses pemrograman generik
- ❑ Programmer membawa beberapa program testing untuk menemukan kegagalan dalam program dan menghilangkan kegagalan dalam proses debugging

33

Rekayasa Perangkat Lunak



Proses Debugging



34

Rekayasa Perangkat Lunak



Validasi Software



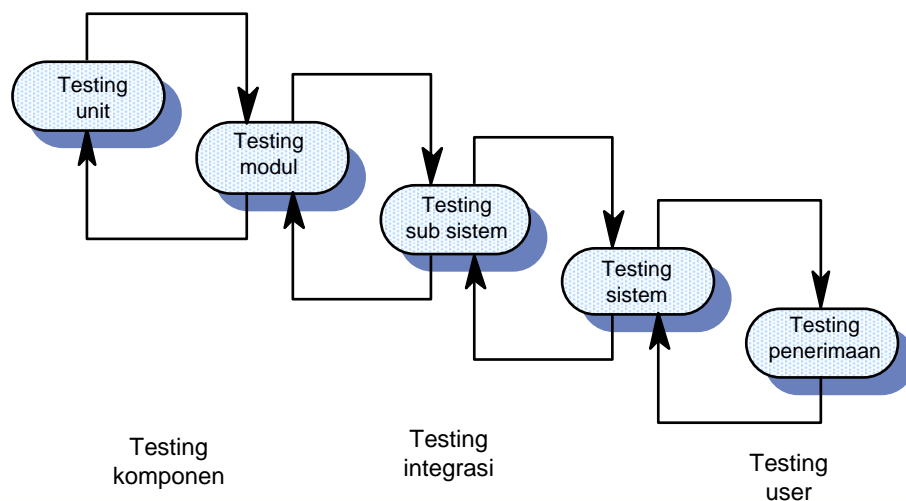
- ❑ Verifikasi dan validasi bertujuan untuk menunjukkan bahwa sistem sesuai spesifikasi dan sesuai kebutuhan konsumen
- ❑ Melibatkan pengecekan dan review proses dan testing sistem
- ❑ Testing sistem melibatkan eksekusi sistem dengan test case yang diambil dari spesifikasi data riil untuk diproses oleh sistem

35

Rekayasa Perangkat Lunak




Proses Testing




36

Rekayasa Perangkat Lunak



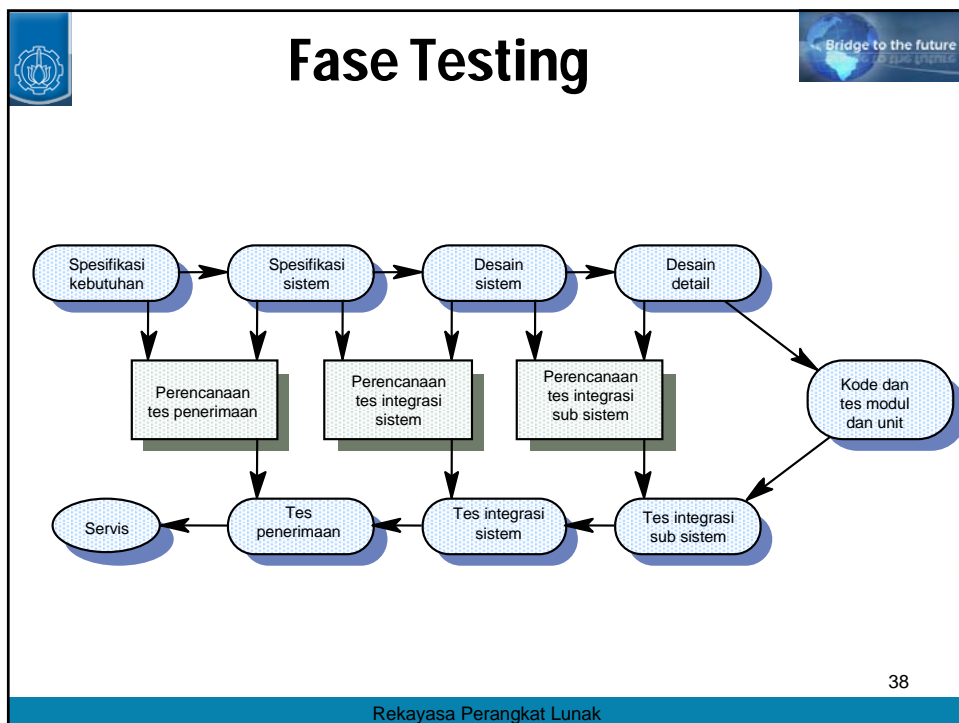
Tingkat Testing




- ❑ Testing Unit
 - o Dilakukan tes pada komponen individu
- ❑ Testing Modul
 - o Dilakukan tes pada kumpulan komponen yang berhubungan
- ❑ Testing sub-system
 - o Modul diintegrasikan ke dalam sub sistem dan dilakukan tes. Fokus pada testing antar muka
- ❑ Testing sistem
 - o Testing pada keseluruhan sistem. Terting terhadap properti penting
- ❑ Testing Penerimaan
 - o Testing dengan data konsumen untuk memeriksa apakah dapat diterima


37

Rekayasa Perangkat Lunak



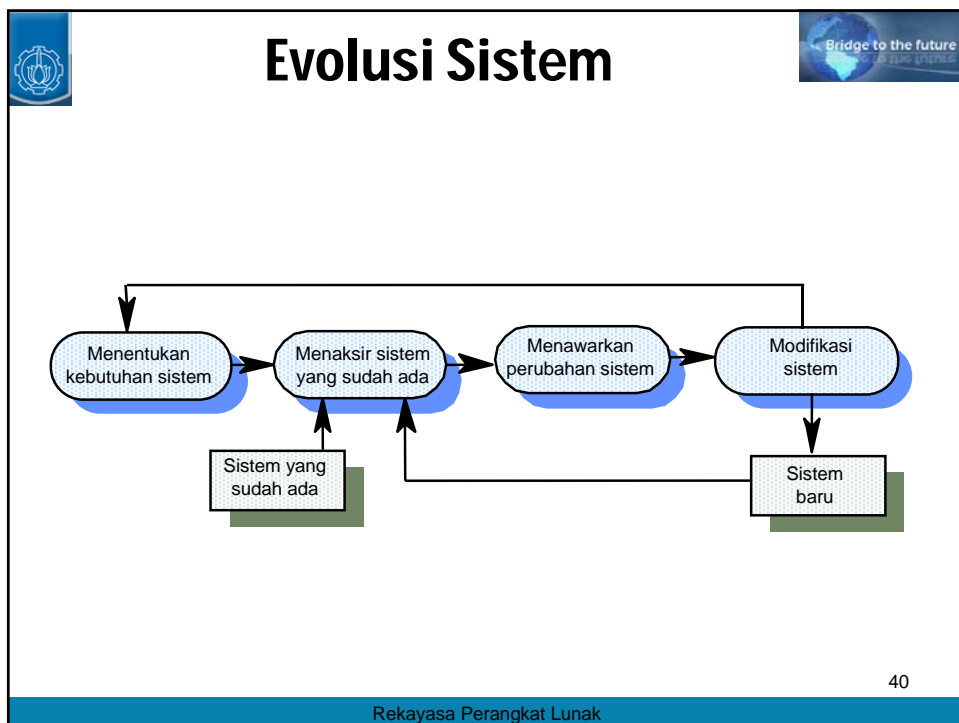


Evolusi Software



- ❑ Software bersifat fleksibel dan dapat berubah
- ❑ Perubahan kebutuhan karena perubahan lingkungan bisnis, software yang mendukung bisnis juga harus terlibat dan berubah
- ❑ Meskipun terdapat batas antara pengembangan dan evolusi (pemeliharaan), peningkatan yang tidak berhubungan menyebabkan sistem sedikit demi sedikit menjadi sistem baru.

Rekayasa Perangkat Lunak
39





Pendukung Proses Otomasi (CASE)



- ❑ Computer-aided software engineering (CASE) adalah software untuk mendukung proses pengembangan dan evolusi
- ❑ Aktivitas otomasi
 - o Editor grafis untuk pengembangan model sistem
 - o Data dictionary untuk mengatur desain entity
 - o Graphical UI builder untuk konstruksi antar muka user
 - o Debugger untuk menemukan kegagalan program
 - o Translator otomatis untuk membangkitkan versi baru dari program

41

Rekayasa Perangkat Lunak




Teknologi CASE




- ❑ Teknologi Case memudahkan perbaikan signifikan dalam proses software merkipun bukan perbaikan penting sesuai perkiraan
 - o Rekayasa software membutuhkan pemikiran kreatif - hal ini tidak dapat diotomasi
 - o Rekayasa software adalah aktivitas tim dan untuk proyek besar, banyak waktu dihabiskan untuk interaksi tim. Teknologi CASE tidak mendukung hal ini

42

Rekayasa Perangkat Lunak



Klasifikasi CASE




- ❑ Klasifikasi membantu mengerti perbedaan tipe tool CASE dan dukungan untuk aktifitas proses
- ❑ Perspektif Fungsional
 - o Tool diklasifikasi berdasarkan fungsi tertentu
- ❑ Perspektif Proses
 - o Tool diklasifikasi berdasarkan aktifitas proses yang didukung
- ❑ Perspektif Integrasi
 - o Tool diklasifikasi berdasarkan organisasi ke dalam unit integrasi

43

Rekayasa Perangkat Lunak



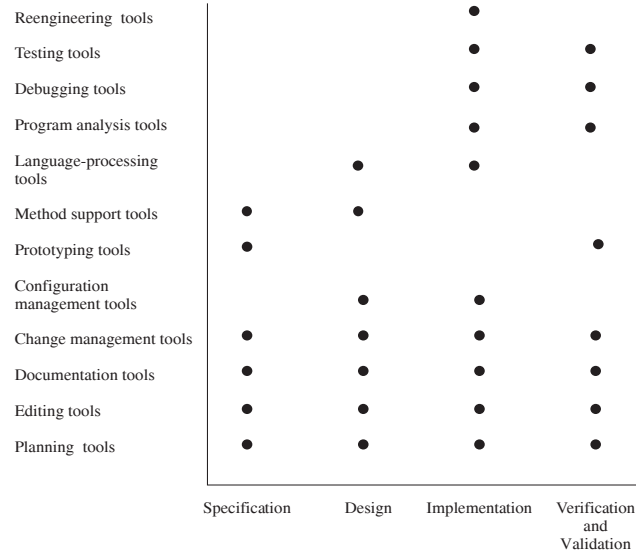
Klasifikasi Functional Tool



Tool type	Examples
Planning tools	PERT tools, estimation tools, spreadsheets
Editing tools	Text editors, diagram editors, word processors
Change management tools	Requirements traceability tools, change control systems
Configuration management tools	Version management systems, system building tools
Prototyping tools	Very high-level languages, user interface generators
Method-support tools	Design editors, data dictionaries, code generators
Language-processing tools	Compilers, interpreters
Program analysis tools	Cross reference generators, static analysers, dynamic analysers
Testing tools	Test data generators, file comparators
Debugging tools	Interactive debugging systems
Documentation tools	Page layout programs, image editors
Re-engineering tools	Cross-reference systems, program re-structuring systems

44

Rekayasa Perangkat Lunak



Klasifikasi berbasis aktifitas

45

Rekayasa Perangkat Lunak



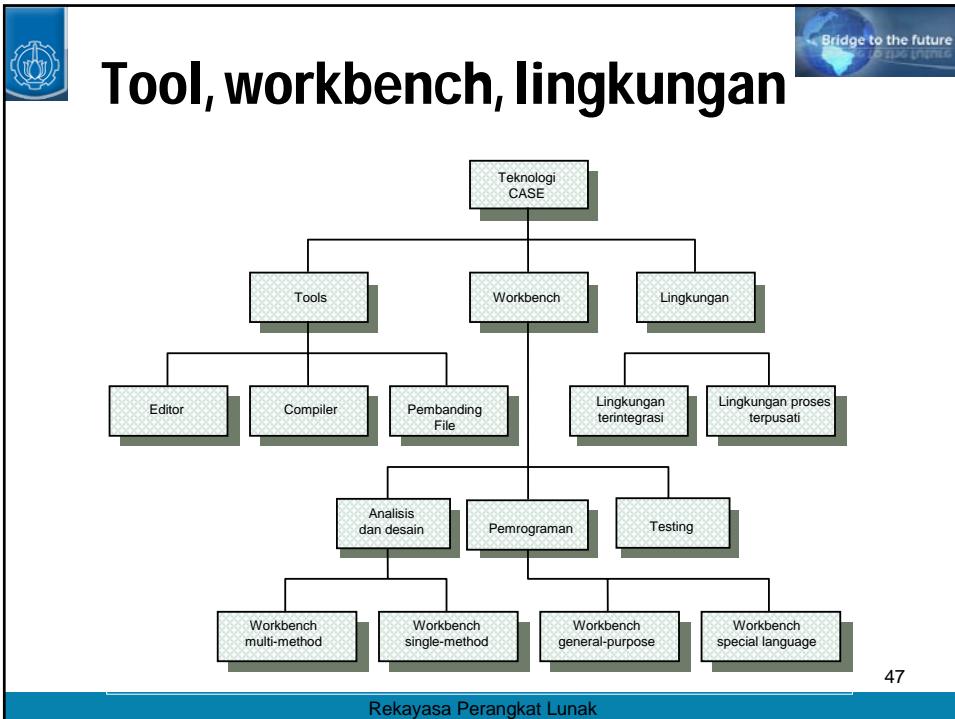
Integrasi CASE



- ☐ Tool
 - o Mendukung task proses individu seperti pemeriksaan konsistensi desain, text editing dll
- ☐ Workbench
 - o Mendukung fase proses seperti spesifikasi atau desain. Biasanya melibatkan sejumlah tool integrasi
- ☐ Lingkungan
 - o Mendukung semua atau bagian substansi dari keseluruhan proses software. Biasanya melibatkan beberapa workbench terintegrasi

46

Rekayasa Perangkat Lunak



- Summary**
- ❑ Proses software adalah aktivitas yang terjadi dalam memproduksi dan menghasilkan sistem software. Direpresentasikan dalam model proses software
 - ❑ Aktivitas umum adalah spesifikasi, desain dan implementasi, validasi dan evolusi
 - ❑ Model proses generik menggambarkan organisasi dari proses software
 - ❑ Model proses iteratif menggambarkan proses software sebagai siklus aktivitas
- The slide number 48 is visible in the bottom right corner.
- Rekayasa Perangkat Lunak



Summary



- ☐ Rekayasa kebutuhan adalah proses mengembangkan spesifikasi software
- ☐ Proses desain dan implementasi mengubah spesifikasi ke program eksekusi
- ☐ Validasi melibatkan pemeriksaan bahwa sistem sesuai dengan spesifikasi dan keperluan user
- ☐ Evolusi menyangkut modifikasi sistem setelah digunakan
- ☐ Teknologi CASE mendukung aktifitas proses software

49