

PARADIGMA DAN PRINSIP INTERAKSI

- **Sistem Interaktif** memungkinkan user mencapai suatu tujuan tertentu dalam suatu domain aplikasi. Sistem interaktif harus dapat didayagunakan (usability) untuk meningkatkan keberhasilan suatu sistem aplikasi.

- **Dua pertanyaan (masalah) tentang pendayagunaan (usability) sistem interaktif :**
 - Bagaimana suatu sistem interaktif dibuat/ dibangun supaya mempunyai dayaguna yang tinggi ?
 - Bagaimana mengukur atau mendemontra-sikan dayaguna (usability) suatu sistem interaktif ?

- **Dua pendekatan untuk menjawab pertanyaan/ masalah di atas:**
 - **Paradigma :**
 - ◆ Sistem interaktif yang berhasil /sukses pada umumnya diyakini akan meningkatkan dayaguna (usability) dari sistem tersebut.
 - **Prinsip :**
 - ◆ Interaksi efektif dari berbagai aspek pengetahuan psikologi, komputasi dan sosiologi mengarahkan peningkatan desain dan evolusi suatu produk, yang pada akhirnya akan meningkatkan daya-guna sistem tersebut.

JENIS PARADIGMA

1. Time-Sharing

- ◆ Satu komputer yang mampu mendukung (dapat digunakan oleh) multiple user
- ◆ Meningkatkan keluaran (throughput) dari sistem

2. Video Display Units (VDU)

- ◆ Dapat memvisualisasikan dan memanipulasi informasi yang sama dalam representasi yang berbeda
- ◆ Mampu memvisualisasikan abstraksi data

3. Programming Toolkits (Alat Bantu Pemrograman)

- ◆ Alat Bantu Pemrograman memungkinkan programmer meningkatkan produktivitasnya

4. Komputer Pribadi (Personal Computing)

- ◆ Mesin berukuran kecil yang powerful, yang dirancang untuk user tunggal.

5. Sistem Window dan interface WIMP (Windows, Icons, Menus and Pointers)

- ◆ Sistem window memungkinkan user untuk berdialog / berinteraksi dengan komputer dalam beberapa aktivitas/topik yang berbeda

6. Metapora (Metaphor)

- ◆ Metapora telah cukup sukses digunakan untuk mengajari konsep baru, dimana konsep tersebut telah dipahami sebelumnya.
- ◆ Contoh metapora (dalam domain PC):
 - ◆ Spreadsheet adalah metapora dari Accounting dan Financial Modelling
 - ◆ Keyboard adalah metapora dari Mesin Tik

7. Manipulasi Langsung (Direct Manipulation)

- ◆ **Manipulasi Langsung** memungkinkan user untuk mengubah keadaan internal sistem dengan cepat.
- ◆ **Contoh Direct Manipulation** adalah konsep WYSIWYG (what you see is what you get)

8. Bahasa vs. Aksi (Language versus Action)

- ◆ **Bahasa** digunakan oleh user untuk berkomunikasi dengan interface
- ◆ **Aksi** dilakukan interface untuk melaksanakan perintah user

9. Hypertext

- ◆ **Penyimpanan informasi** dalam format linear tidak banyak mendukung pengaksesan informasi secara random dan browsing asosiatif.
- ◆ **Hypertext** merupakan metode penyimpanan informasi dalam format non-linear yang memungkinkan akses atau browsing secara non-linear atau random.

10. Multi-Modality

- ◆ **Sistem multi-modal interaktif** adalah sistem yang tergantung pada penggunaan beberapa (multiple) saluran (channel) komunikasi pada manusia.
- ◆ **Contoh channel komunikasi** pada manusia : visual (mata), haptic atau peraba (kulit) audio (telinga).

11. Computer-Supported Cooperative Work (CSCW)

- ◆ **Perkembangan jaringan komputer** memungkinkan komunikasi antara beberapa mesin (personal komputer) yang terpisah dalam satu kesatuan grup.
- ◆ **Sistem CSCW** dirancang untuk memungkinkan interaksi antar manusia melalui komputer dan direpresentasikan dalam satu produk.

- ◆ **Contoh CSCW: e-mail (electronic mail)**

PRINSIP YANG Mendukung Pendayagunaan

- ◆ **Learnability** : kemudahan yang memungkinkan user baru berinteraksi secara efektif dan dapat mencapai performance yang maksimal
- ◆ **Flexibility** : menyediakan banyak cara bagi user dan sistem untuk bertukar informasi
- ◆ **Robustness**: tingkat dukungan yang diberikan agar user dapat menentukan keberhasilannya atau tujuan (goal) yang diinginkan.

Tabel 1. Ringkasan Prinsip yang Mempengaruhi Kemampuan Belajar (Learnability)

Prinsip	Definisi	Prinsip yang Terkait
Predictability	Mendukung user untuk menentukan efek dari 'future action' berdasarkan catatan/sejarah interaksi sebelumnya	Operation visibility
Synthesizability	Mendukung user untuk memperkirakan efek dari operasi sebelumnya pada keadaan saat ini	Immediate / Eventual Honesty
Familiarity	Pengetahuan dan pengalaman user dalam domain berbasis komputer atau dunia nyata lainnya dapat diterapkan ketika berinteraksi dengan sistem yang baru	Guessability Affordance
Generalizability	Mendukung user untuk menambah pengetahuan dari interaksi spesifik di dalam dan di luar aplikasi ke situasi lainnya yang mirip	
Consistency	Kemiripan dalam perilaku input / output yang muncul dari situasi atau tugas obyektif yang sama	

Tabel 2. Ringkasan Prinsip yang Mempengaruhi Fleksibilitas

Prinsip	Definisi	Prinsip yang Terkait
Dialogue Initiative	Memungkinkan user terbebas dari kendala-kendala buatan (artificial) pada dialog input yang dipaksakan oleh sistem	System/User preemptive-ness
Multi-Treading	Kemampuan system untuk mendukung interaksi user yang berhubungan dengan lebih dari satu task pada suatu saat (waktu)	Concurrent vs. interleaving, multi-modality
Task Migratability	Kemampuan untuk melewatkan / memberikan kontrol dari eksekusi task yang diberikan sehingga menjadi task internal user atau sistem atau berbagi antara keduanya	
Substitutivity	Memungkinkan nilai-nilai (values) ekuivalen antara input dan output yang masing-masing secara bebas dapat disubstitusi	Representasi perkalian, kesamaan kesempatan (opportunity)
Customizability	Kemampuan user inter-face untuk dimodifikasi oleh user atau system	Adaptivity, Adaptability

Tabel 3. Ringkasan Prinsip yang Mempengaruhi Robustness

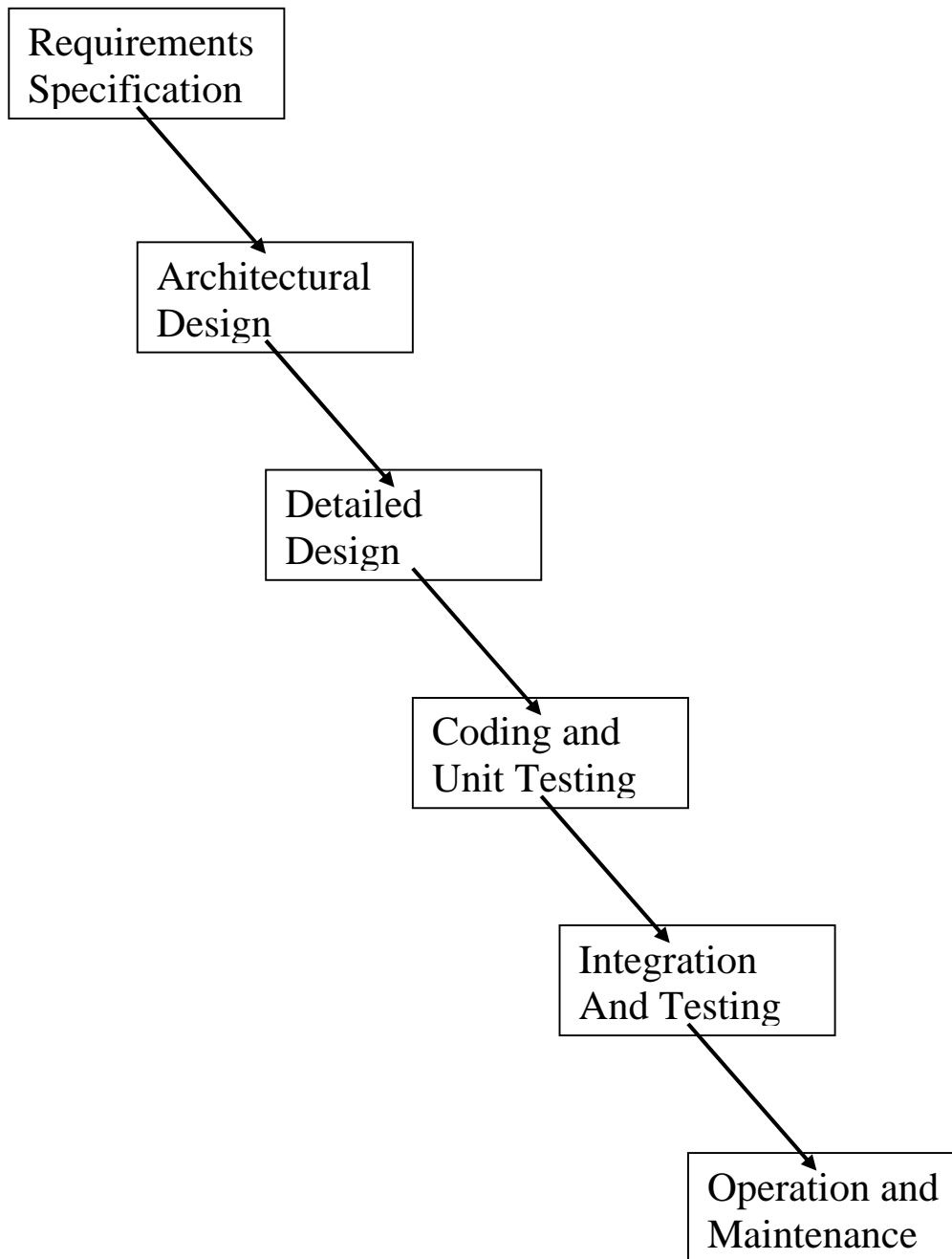
Prinsip	Definisi	Prinsip yang Terkait
Observability	Kemampuan user untuk mengevaluasi keadaan internal system dari representasi yang dapat dimengerti/dirasakan	Browsability, static/dynamic defaults, reachability, persistence, operation visibility
Recoverability	Kemampuan user untuk melakukan koreksi bila sebuah error (kesalahan) telah dikenali	Reachability, forward/backward recovery commensurate effort
Responsiveness	Bagaimana user mengetahui/menyadari laju komunikasi dengan sistem	Stability
Task Conformance	Tingkatan dimana sistem pelayanan mendukung semua tasks yang user ingin lakukan dan dengan cara yang user ketahui	Task completeness, task adequacy

PROSES PERANCANGAN (DESAIN)

Objectives (tujuan umum):

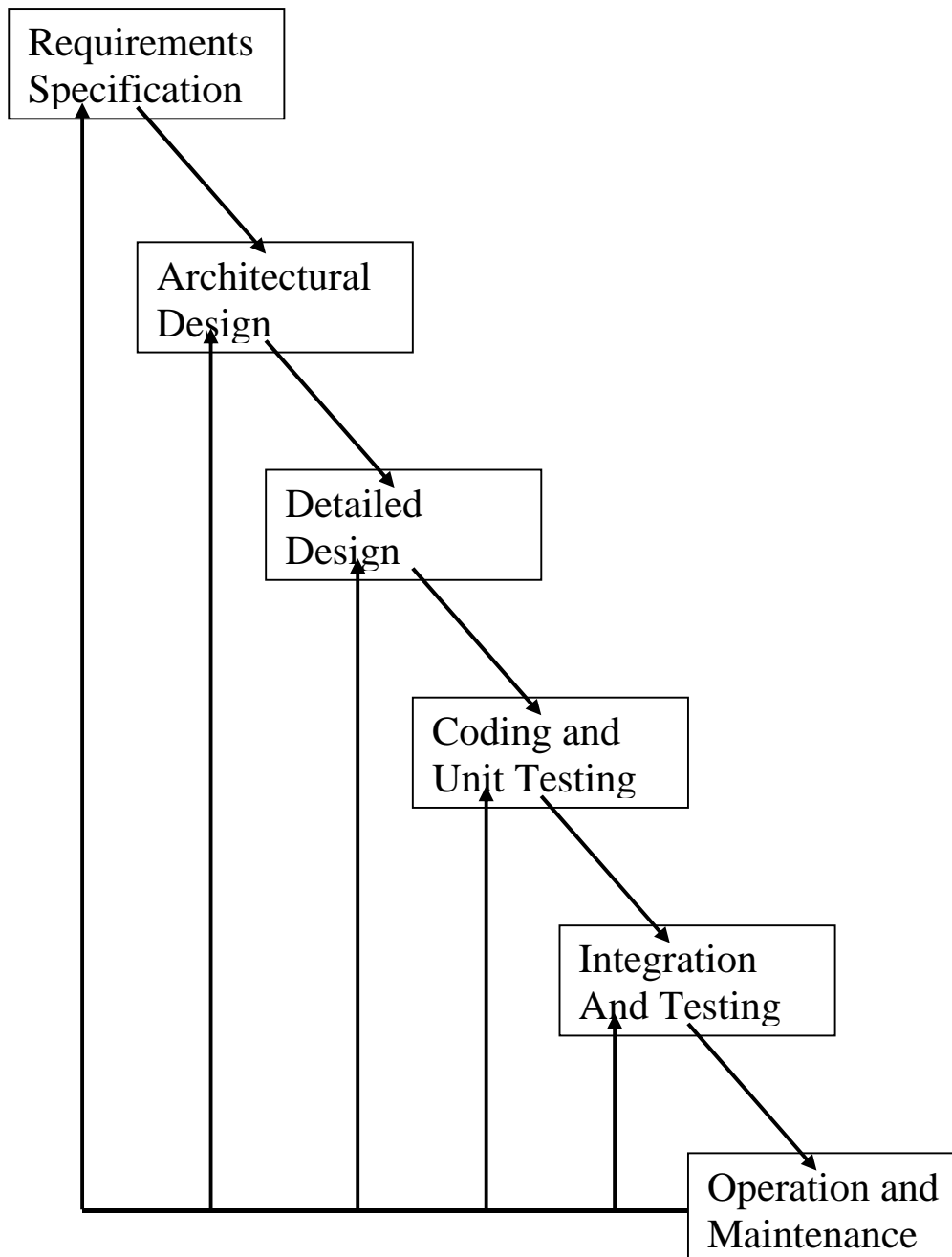
- ◆ **Software engineering memberikan suatu cara untuk memahami struktur proses perancangan (desain), dimana proses tersebut dapat mendukung efektivitas perancangan sistem interaktif.**
- ◆ **Aturan-aturan perancangan (design rules) dalam bentuk standard dan guidelines memberikan arah perancangan, baik dalam bentuk umum maupun dalam bentuk kongkrit, dalam rangka meningkatkan sifat-sifat interaktif dari sistem.**
- ◆ **Usability engineering (rekayasa dayaguna) menawarkan penggunaan kriteria secara eksplisit untuk menilai (judge) keberhasilan suatu produk dalam bentuk dayagunanya.**
- ◆ **Perancangan iterative memungkinkan kerja sama antara customer dengan perancang (designer) untuk mendapatkan feedback (umpan balik) yang berbentuk keputusan yang kritis yang mempengaruhi dayaguna, di awal proses perancangan**
- ◆ **Perancangan melibatkan pengambilan berbagai keputusan diantara sejumlah alternatif.**

Daur-Hidup Pengembangan Software



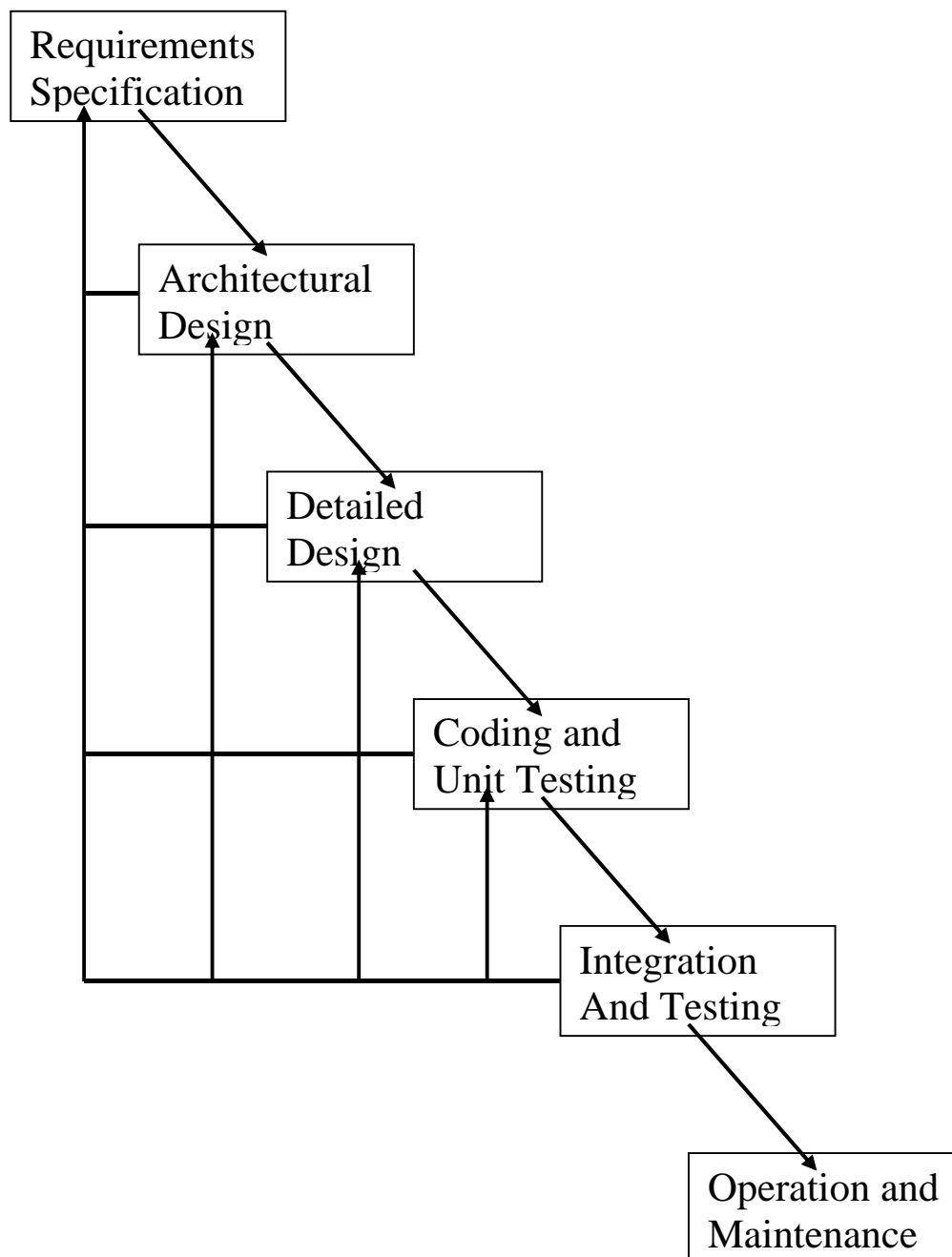
Gambar 2.1. Aktifitas dalam Model Air Terjun dari Daur-Hidup Pengembangan Software

Validasi dan Verifikasi



Gambar 2.2. Feedback dari Aktifitas Maintenance dan Aktifitas Perancangan Lainnya

Sistem Interaktif dan Daur-Hidup Software



Gambar 2.3. Representasi iterasi dalam Model Air Terjun

Penggunaan Aturan Perancangan (Design Rules)

◆ Standard (ISO Standard 9241):

- usability
- effectiveness
- efficiency
- satisfaction

◆ Guidelines :

- data entry
- data display
- sequence control
- user guidance
- data transmission
- data protection

Usability Metrics

Tabel 4. Kriteria untuk Metode Pengukuran Usability Engineering

1	Time to complete a task
2	Percent of task completed
3	Percent of task completed per unit time
4	Ratio of successes to failures
5	Time spent in errors
6	Percent of number of errors
7	Percent of number of competitors better than it
8	Number of commands used
9	Frequency of help and documentation use
10	Percent of favourable/unfavourable user comments
11	Number of repetition of failed commands
12	Number of runs of successes and of failures
13	Number of times interface misleads the user
14	Number of good and bad features recalled by users
15	Number of available commands not invoked
16	Number of regressive behaviours
17	Number of users preferring your system
18	Number of times users need to work around a problem
19	Number of times the user is disrupted from a work task
20	Number of times user loses control of the system
21	Number of times user expresses frustration or satisfaction

Desain Iteratif dan Prototyping

Tiga pendekatan utama prototyping:

- **Throw-away** : prototype dibuat dan ditest. Pengalaman yang diperoleh dari pembuatan prototype tersebut digunakan untuk membuat produk akhir (final), sementara prototype tersebut dibuang (tak dipakai)
- **Incremental** : produk akhirnya dibuat sebagai komponen-komponen yang terpisah. Desain produk akhirnya secara keseluruhan hanya ada satu, tetapi dibagi-bagi dalam komponen-komponen lebih kecil yang terpisah (independent)
- **Evolutionary** : Dalam metode ini, prototypenya tidak dibuang tetapi digunakan untuk iterasi desain berikutnya. Dalam hal ini, sistem atau produk yang sebenarnya dipandang sebagai evolusi dari versi awal yang sangat terbatas menuju produk final atau produk akhir.

Tabel 5. Contoh Usability Metrics dari ISO 9241

Usability objectives	Effectiveness measures	Efficiency measures	Satisfaction measures
Suitability for the task	Percentage of goals achieved	Time to complete a task	Rating scale for satisfaction
Appropriate for trained users	Number of power features used	Relative efficiency compared with an expert user	Rating scale for satisfaction with power features
Learnability	Percentage of functions learned	Time to learn criterion	Rating scale for ease learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling

