

## **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 mendemonstrasikan dayaguna (usability) suatu sistem interaktif ?**
- **Dua pendekatan untuk menjawab pertanyaan/ masalah di atas:**
  - **Paradigma :**
    - ◆ **Sistem interaktif yang berhasil /sukses pada umumnya diyakini akan mening-katkan 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):**
    - ◆ **Spreadsheets** 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)**

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

**Tabel 2. Ringkasan Prinsip yang Mempengaruhi Fleksibilitas**

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

**Tabel 3. Ringkasan Prinsip yang Mempengaruhi Robustness**

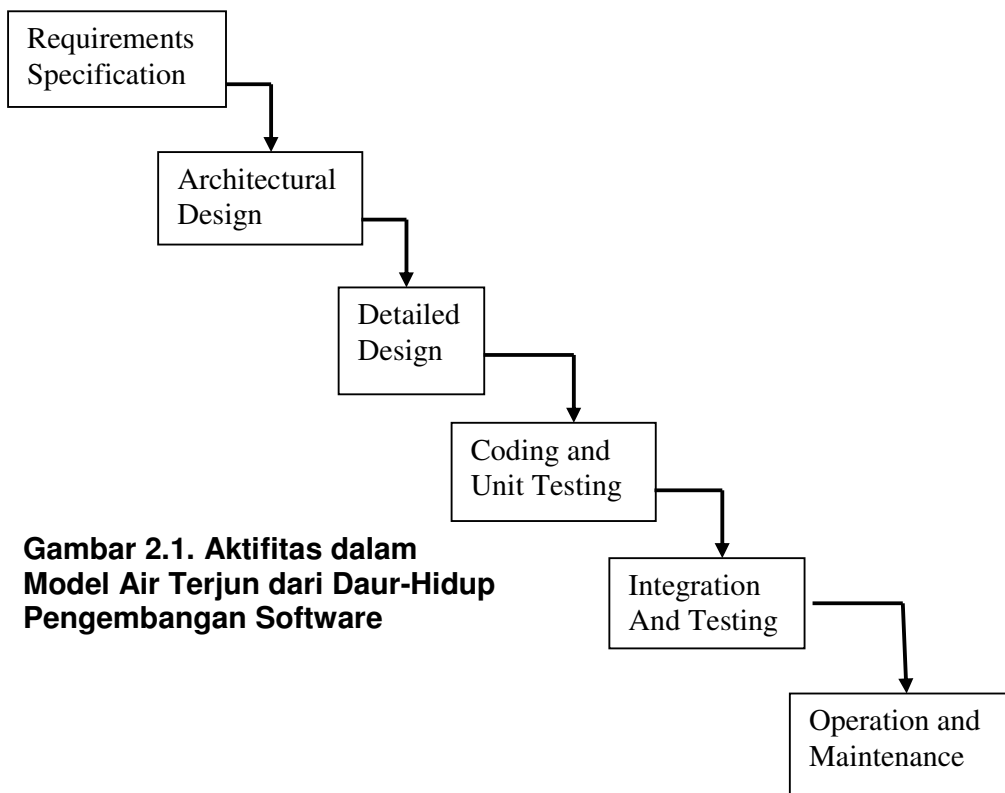
<b>Prinsip</b>	<b>Definisi</b>	<b>Prinsip yang Terkait</b>
<b>Observability</b>	Kemampuan user untuk mengevaluasi keadaan internal system dari representasi yang dapat dimengerti/dirasakan	<b>Browsability, static/dynamic defaults, reachability, persistence, operation visibility</b>
<b>Recoverability</b>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

## PROSES PERANCANGAN (DESAIN)

### Objectives (tujuan umum):

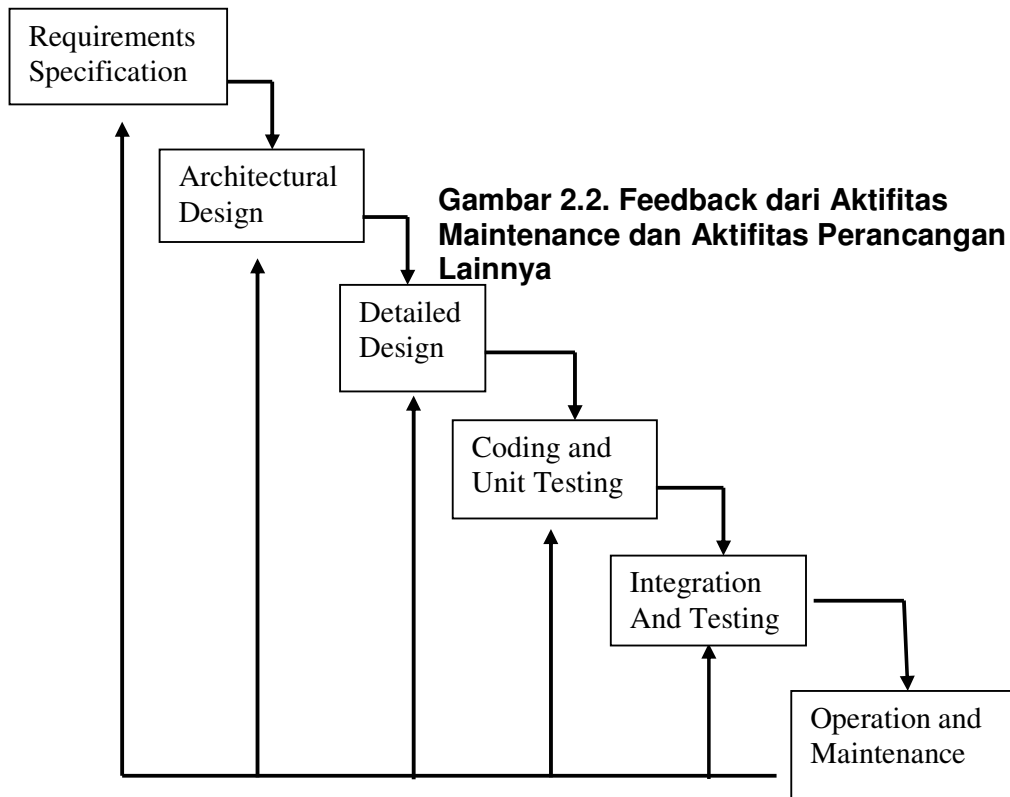
- ◆ Software engineering memberikan suatu cara untuk memahami struktur proses perancangan (desain), dimana proses tersebut dapat mendukung efektivitas perancangan sistem interaktif.
- ◆ Aturan-aturan perancangan (design rules) dalam bentuk standard dan guidelines memberikan arah perancangan, baik dalam 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

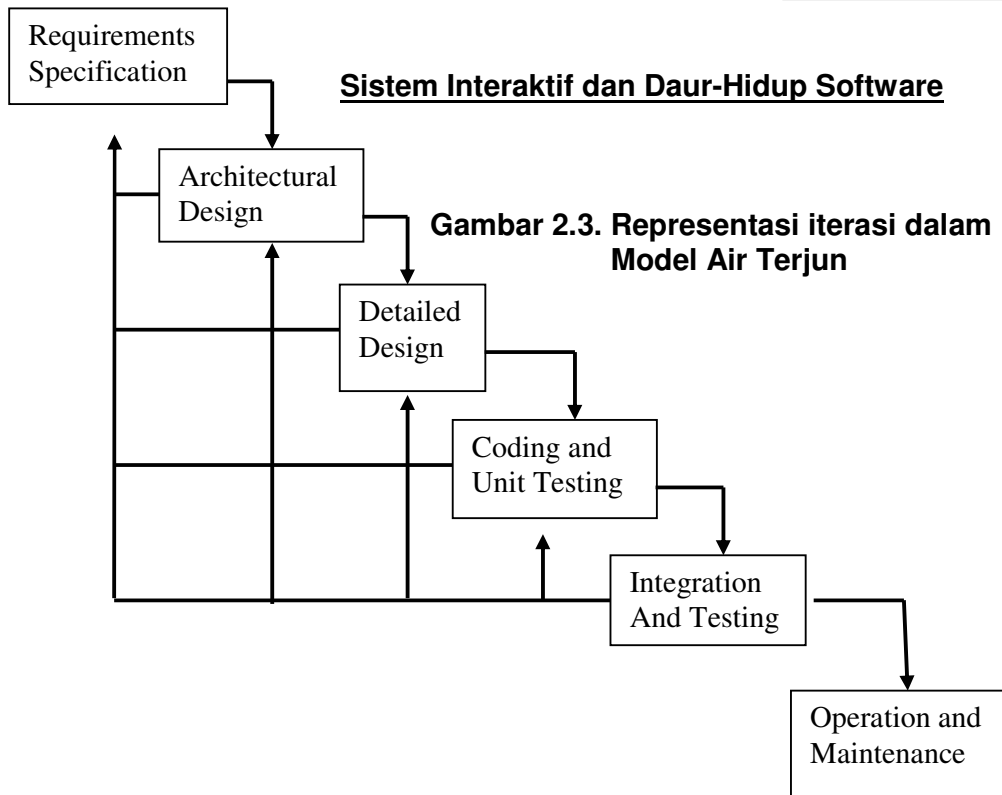


**Gambar 2.1. Aktifitas dalam Model Air Terjun dari Daur-Hidup Pengembangan Software**

**Validasi dan Verifikasi**



**Gambar 2.2. Feedback dari Aktivitas Maintenance dan Aktivitas 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

Tabel 5. Contoh Usability Metrics dari ISO 9241

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



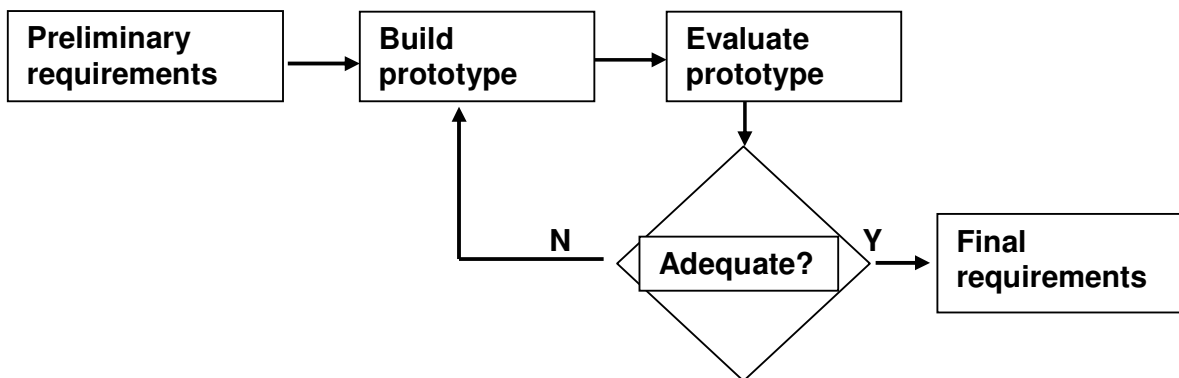
## Desain Iteratif dan Prototyping

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.

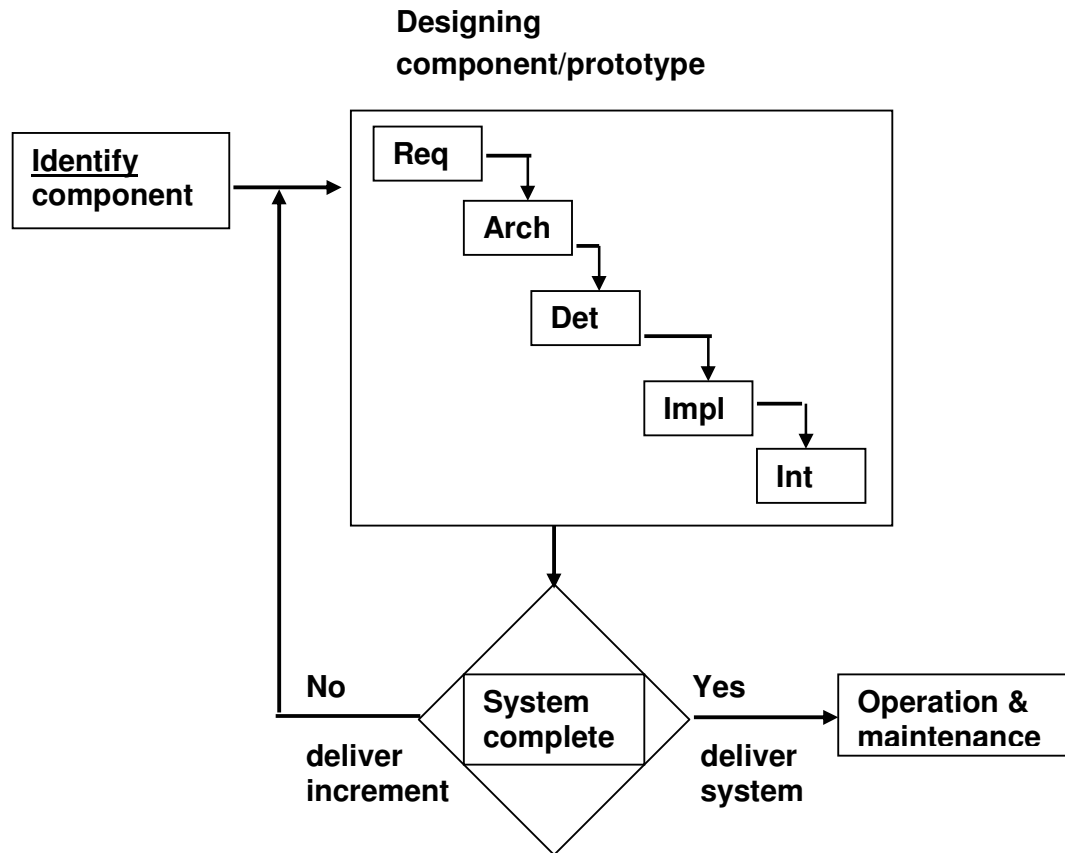
Tiga Metode (Pendekatan) Prototyping

- **Throw-away** : Prototype dibuat dan ditest, pengalaman yang diperoleh dari latihan ini digunakan untuk membuat produk akhirnya, tetapi prototype-nya sendiri dibuang.



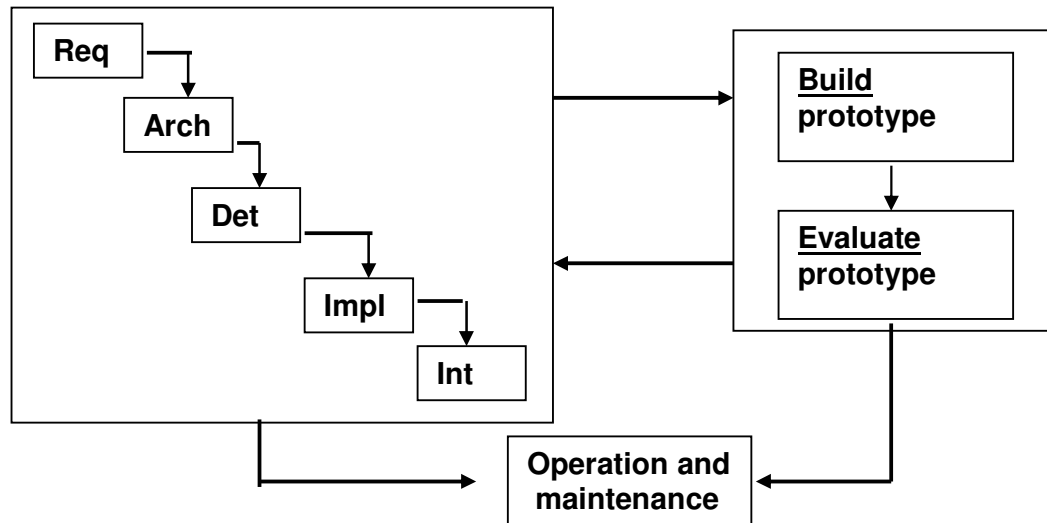
Gambar 2.4. Throw-away Prototyping

- **Incremental** : Produk akhirnya dibuat dalam bentuk komponen-komponen yang terpisah, dan dikerjakan satu komponen dalam satu saat. Sebetulnya hanya ada satu rancangan final dari sistem, tetapi sistem ini dibagi-bagi dalam komponen-komponen terpisah yang lebih kecil.



Gambar 2.5. Incremental Prototyping

- **Evolutionary** : Pada metode ini, prototype-nya tidak dibuang, melainkan digunakan sebagai dasar untuk iterasi perancangan selanjutnya. Dalam hal ini, sistem yang sesungguhnya dipandang sebagai evolusi dari versi awal yang terbatas menuju produk akhirnya.



Gambar 2.6. Evolutionary Prototyping

### Teknik-Teknik untuk Prototyping

- Storyboards + Animation
- Limited Functionality Simulations
- High-Level Programming Support (Hypertalk)

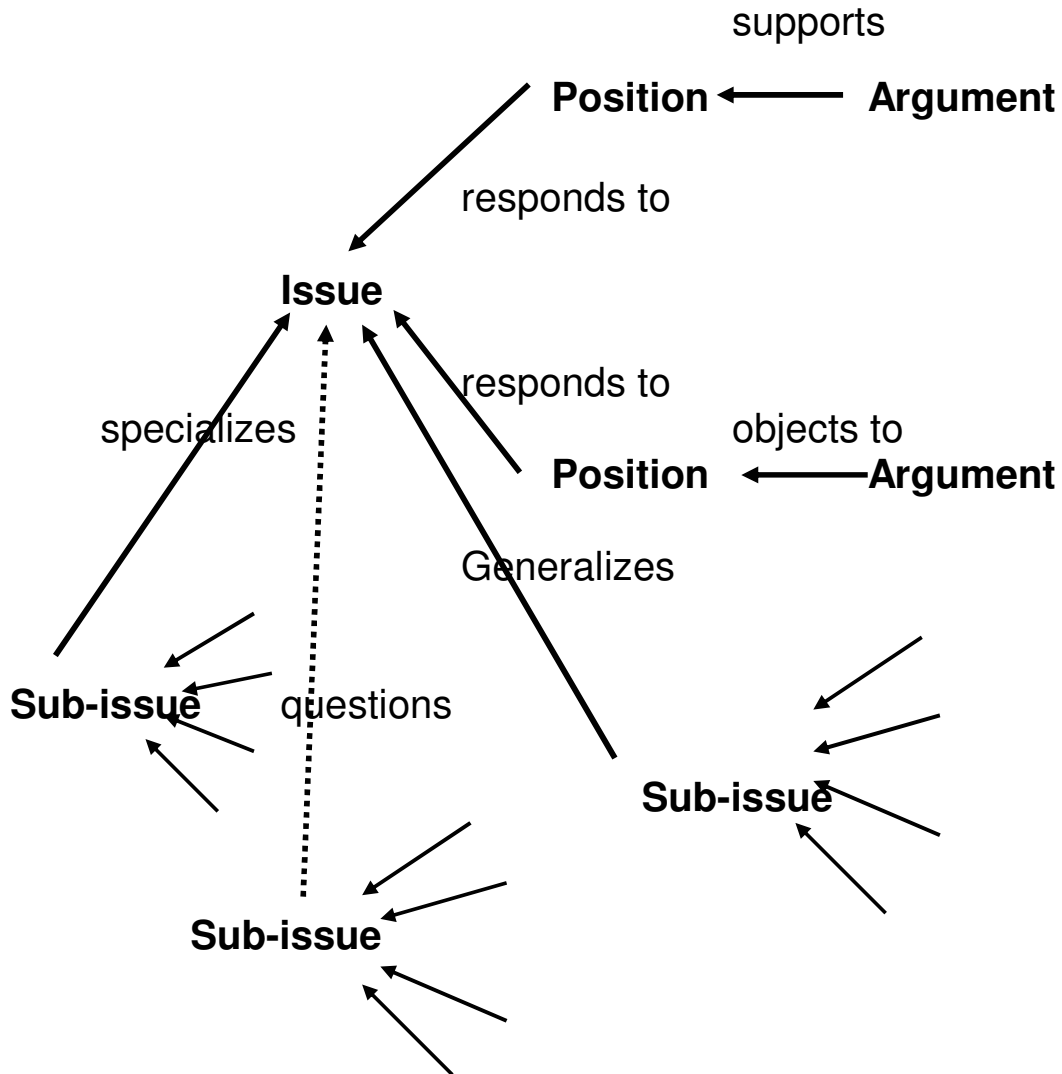
### Rasionalitas Desain/Perancangan

Rasionalitas Desain / Perancangan adalah informasi yang menjelaskan alasan mengapa suatu keputusan dalam suatu tahap perancangan / desain dibuat atau diambil.

Beberapa keuntungan rasionalitas perancangan:

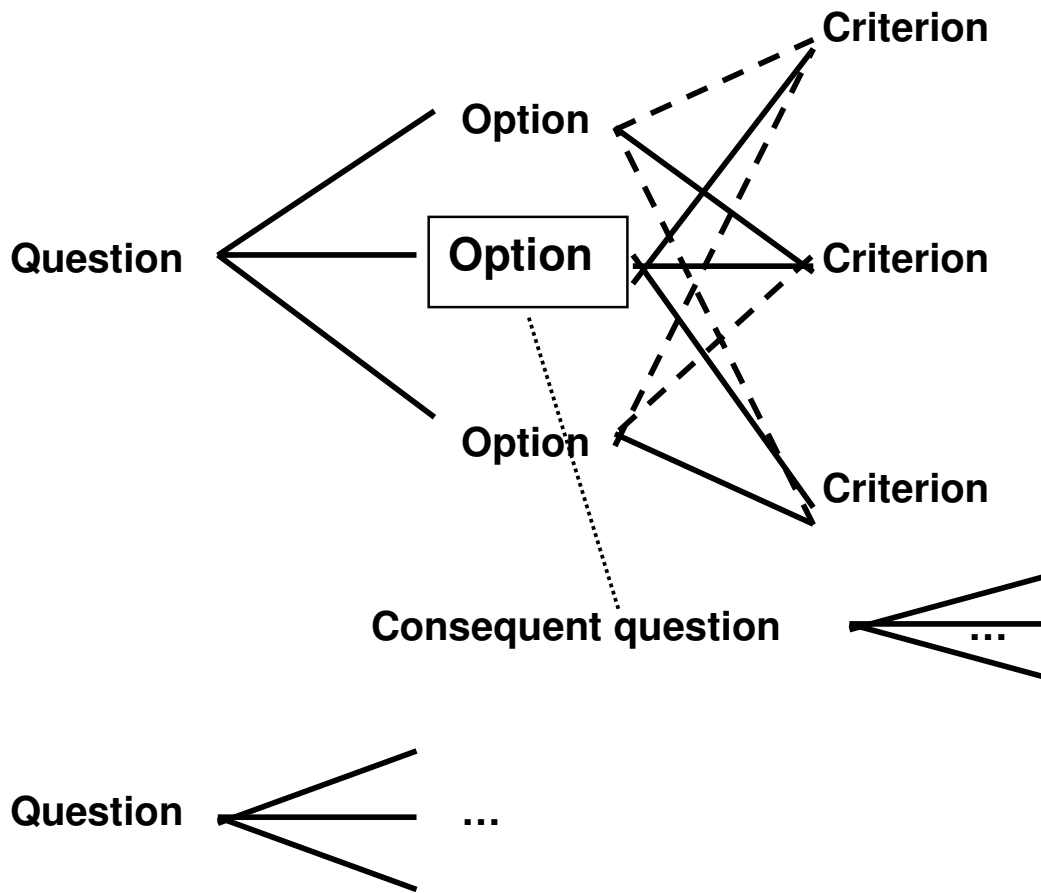
- Dalam bentuk yang eksplisit, rasionalitas perancangan menyediakan mekanisme komunikasi diantara anggota team desain sehingga pada tahapan desain dan atau pemeliharaan (maintenance), anggota team memahami keputusan kritis / penting mana yang telah dibuat, alternatif apa saja yang telah diteliti, dan alasan apa yang menyebabkan suatu alternatif dipilih diantara alternatif lainnya.
- Akumulasi pengetahuan dalam bentuk rasionalitas desain untuk suatu set produk dapat digunakan kembali untuk mentransfer apa saja yang telah bekerja dalam suatu situasi ke situasi lainnya yang mirip.

- Usaha yang diperlukan untuk menghasilkan sebuah rasionalitas desain memaksa desainer (perancang) untuk bersikap hati-hati dalam mengambil suatu keputusan desain.
- Rasionalitas Desain Berorientasi Proses



Gambar 2.7. The Structure of a gIBIS Design Rationale

➤ Analisa Ruang Desain (Design Space Analysis)



Gambar 2.8. The QOC Notation (Questions, Options, Criteria)