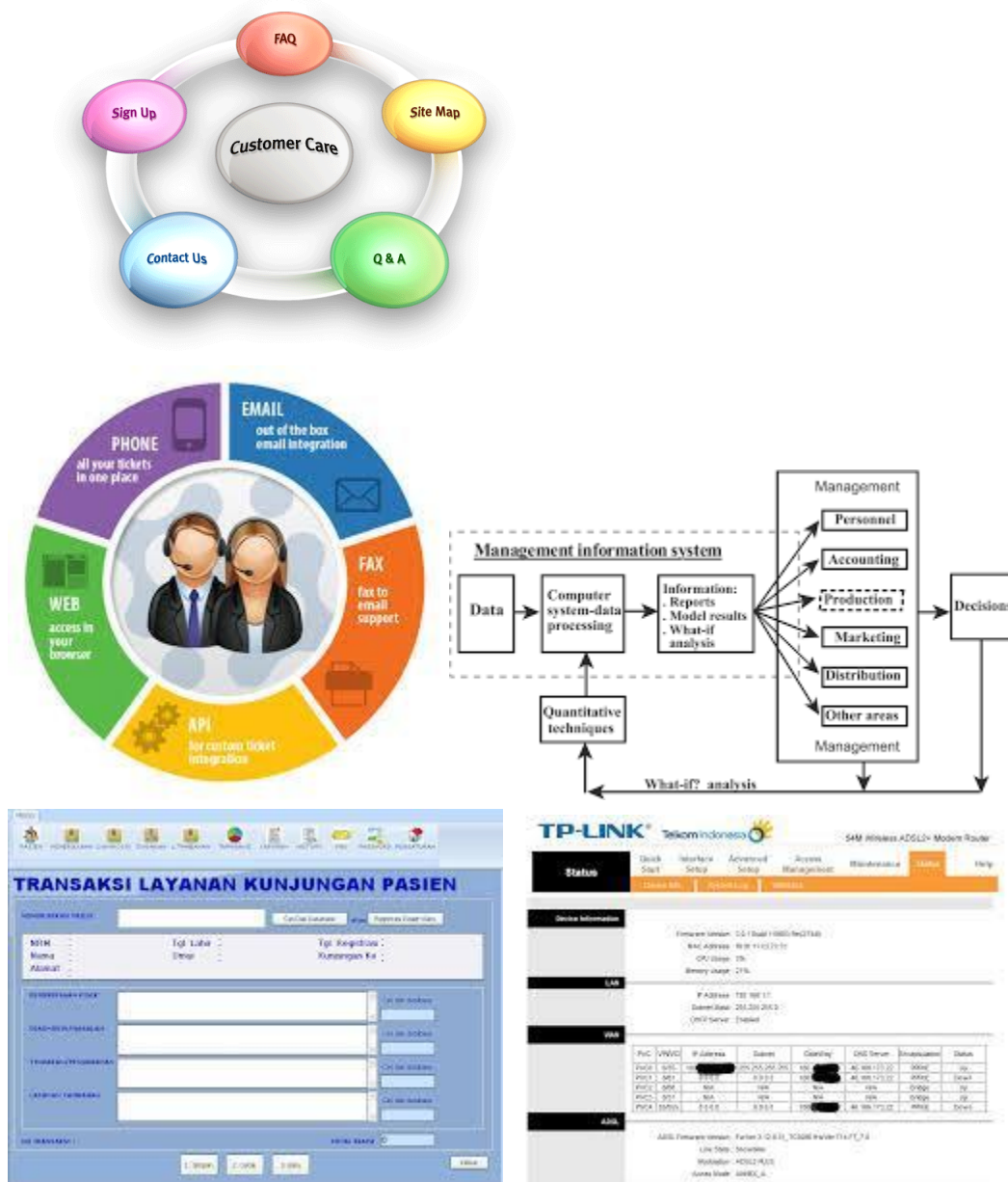


CONTOH USER SUPPORT



JENIS PENANGANAN KESALAHAN DAN HELP DOKUMENTASI

Tinjauan:

- User mempunyai perbedaan kebutuhan
- User support seharusnya:
 - Tersedia tetapi tidak mencolok
 - Akurat dan kuat
 - Konsisten dan fleksibel
- Jenis-jenis user support:
 - Command Based Methods

- Context-Sensitive Help
- Tutorial Help
- On-line Documentation
- Intelligent Help
- Merancang user support harus memperhatikan:

- Presentasi
- Implementasi

Tipe-tipe Kesalahan (Errors):

- Kesalahan Persepsi
- Kesalahan Kognitif
- Kesalahan Motor (Gerak)

Tipe-tipe Slip:

- Kesalahan Capture
- Kesalahan Deskripsi
- Kesalahan Data Driven
- Pengaktifan Asosiatif
- Hilangnya Pengaktifan
- Kesalahan Mode

Panduan Pencegahan Kesalahan

- Menghapus mode-mode atau menyediakan petunjuk yang terlihat untuk mode-mode tersebut.
- Gunakan teknik koding yang baik (warna, gaya).
- Memaksimalkan pengenalan, mengurangi hafalan.
- Merancang urutan gerak atau perintah yang tidak sama.
- Mengurangi kebutuhan untuk mengetik.
- Uji dan memantau kesalahan-kesalahan dan memperbaikinya.
- Memungkinkan pertimbangan ulang aksi-aksi yang dilakukan oleh user, misalnya memindahkan file dari recycle bin.

Panduan Recovery Kesalahan

- Menyediakan tipe-tipe tanggapan yang sesuai.
- Query: bertanya pada user apa yang sudah dilakukan, kemudian melegalkan tindakan yang salah.
- Menyediakan fungsi “undo” dan pembatalan dari proses yang sedang berjalan.
- Meminta konfirmasi untuk perintah yang drastis dan bersifat merusak.
- Menyediakan pengecekan yang beralasan pada masukan data.
- Mengembalikan kursor ke area kesalahan, memungkinkan untuk melakukan perbaikan.
- Menyediakan beberapa kecerdasan buatan.
- Menyediakan akses cepat kepada bantuan untuk konteks-sensitif.

JENIS-JENIS DARI DOKUMENTASI/HELP

- Tidak pernah suatu penggantian untuk desain tidak baik, tetapi penting.
- Sistem sederhana, user memanggil dan menggunakannya, lalu memberikan nama.
- Hampir sebagian sistem dengan banyak fitur membutuhkan help/bantuan.

Jenis-jenis Bantuan:

- Tutorial
- Review/Referensi yang cepat

- Manual Referensi (Penjelasan lengkap)
- Bantuan untuk context-sensitive (spesifikasi tugas)

Ada sebagian pendapat menyatakan bahwa sistem yang interaktif dijalankan tanpa membutuhkan bantuan atau training. Hal ini mungkin ideal, akan tetapi jauh dari kenyataan. Pendekatan yang lebih membantu adalah dengan mengasumsikan bahwa user akan membutuhkan bantuan pada suatu waktu dan merancang bantuan (help) ke dalam sistem.

Empat Jenis Bantuan Yang Dibutuhkan User:

- Quick Reference
- Task-Spesifik Help
- Full Explanation
- Tutorial

Kebutuhan Dari User Support:

- Availability
- Accuracy dan Completeness
- Consistency
- Robustness
- Flexibility
- Unobtrusiveness

Knowledge Representation: User Modelling

- Quantification
- Stereotypes
- Overlay Models

Knowledge Representation: Domain dan Task Modelling

Pendekatan yang umum dari masalah ini adalah untuk mewakili tugas user dari urutan perintah yang tersedia untuk mengeksekusinya. Sebagaimana pada tugas user, command digunakan untuk membandingkan urutan tugas yang telah disimpan dan mencocokkan dengan urutan tepat. Jika urutan command user tidak cocok, maka dibutuhkan bantuan. Pendekatan ini digunakan pada sistem PRIAM.

Knowledge Representation: Modelling Advisory Strategy

Sistem ini kadang disebut dengan intelligent help yang membuat modelling advisory atau strategi tutorial. Pada sistem ini tidak hanya membolehkan memilih nasehat yang cocok untuk user, tetapi juga menggunakan metode yang cocok.

Teknik Untuk Knowledge Representation

Terdapat empat grup utama dari teknik yang digunakan dalam knowledge representation untuk intelligent help system:

1. Rule Based Techniques
2. Frame Based Techniques
3. Network Based Techniques
4. Examples Based Techniques

Masalah dengan Knowledge Representation dan Modelling

Pengetahuan mewakili suatu issue pusat dalam intelligent help system, tetapi tidak tanpa masalah itu sendiri, pengetahuan kadang sulit didapatkan, terutama jika ada domain expert tidak tersedia. Masalah lain adalah menginterpretasikan informasi yang cocok.

Masalah lain:

- Inisiatif
- Effect
- Scope

Merancang User Support System

Terdapat banyak cara untuk merancangnya dan semua itu diserahkan pada perancang untuk memilih cara yang terbaik akan tetapi hal yang perlu diperhatikan adalah:

- Perancangan seharusnya tidak seperti “add-on” pada sistem. Secara ideal seharusnya merupakan bagian integral dalam sistem.
- Perancangan harus memperhatikan isi dari bantuan dan konteks sebelum teknologi tersedia.

Masalah Presentasi

- How is help request?
- How is help displayed?
- Effective presentation of help

Masalah Implementasi

- Para perancang harus membuat keputusan untuk implementasi berupa secara fisik maupun pilihan yang tersedia untuk user. Keputusan ini sudah termasuk dalam pernyataan command operating system, apakah berbentuk meta-command atau aplikasi. Hambatan fisik berupa screen space, kapasitas memori dan kecepatan.
- Masalah lain adalah bagaimana struktur data bantuan: apakah berbentuk single file, hierarchy file atau database.

PARADIGMA DAN PRINSIP INTERAKSI

- ③ Sistem Interaktif memungkinkan user mencapai suatu tujuan tertentu dalam suatu domain aplikasi. Sistem interaktif harus dapat didayagunakan (usability) untuk meningkatkan keberhasilan suatu sistem aplikasi.
- ③ Dua pertanyaan (masalah) tentang pendayagunaan (usability) sistem interaktif :
- ⌚ Bagaimana suatu sistem interaktif dibuat/ dibangun supaya
 - ⌚ Bagaimana pimanayai dayagunaan ya ng taing gi m ? n demontrasikan dayaguna (usability) suatu sistem interaktif ?
- ③ Dua pendekatan untuk menjawab pertanyaan/ masalah di atas: $\frac{3}{4}$ Paradigma :
- ♦ Sistem interaktif yang berhasil /sukses pada umumnya diyakini akan mening-katkan dayaguna (usability) dari sistem
- ⌚ Printesrispe :b u t.
- ♦ 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 nonlinear 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)

RINSIP 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 lain 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-Threading	Kemampuan system untuk mendukung interaksi user yang berhubungan dengan lebih dari satu task pada suatu saat (waktu)	Concurrent vs. interleaving, multimodality
Task Migratability	Kemampuan untuk melewati / 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

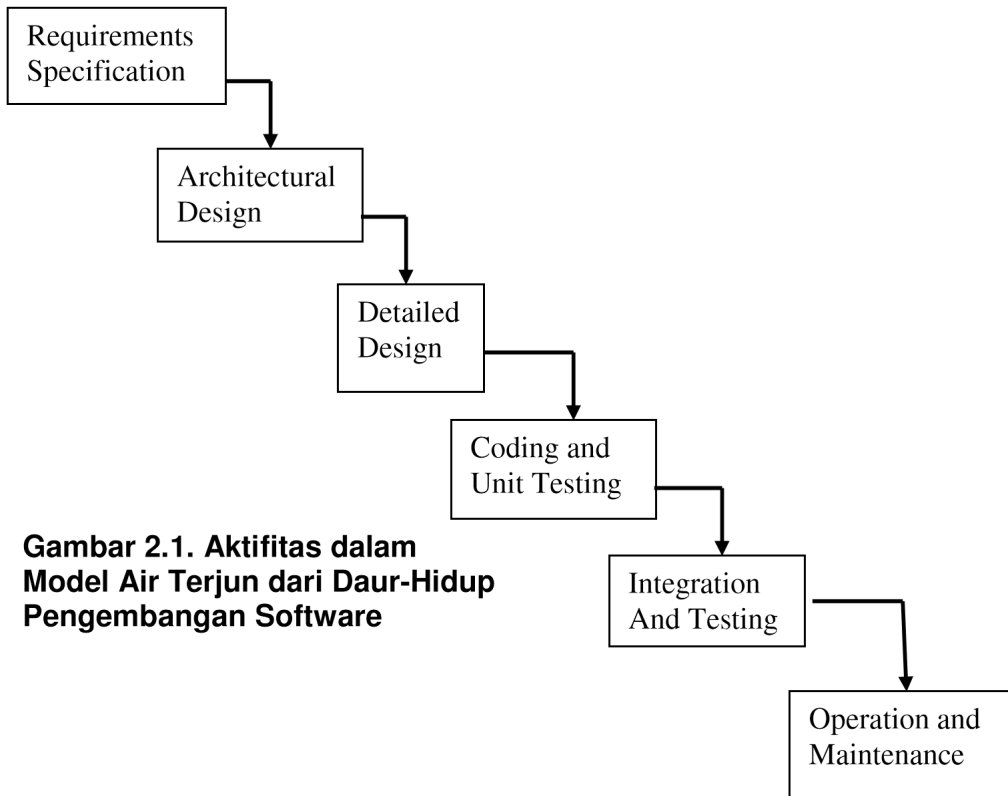
Prinsip	Definisi	Prinsip yang Terkait
Observability	Kemampuan user untuk mengevaluasi keadaan internal system dari representasi yang dapat dimengerti/dirasakan	Browsability, static/dynamic defaults, reachability, persistence, operation visibility
Recoverability	Kemampuan user untuk melakukan koreksi bila sebuah error (kesalahan) telah dikenali	Reachability, forward/back-ward 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

ROSES PERANCANGAN (DESAIN)

Objectives (tujuan umum):

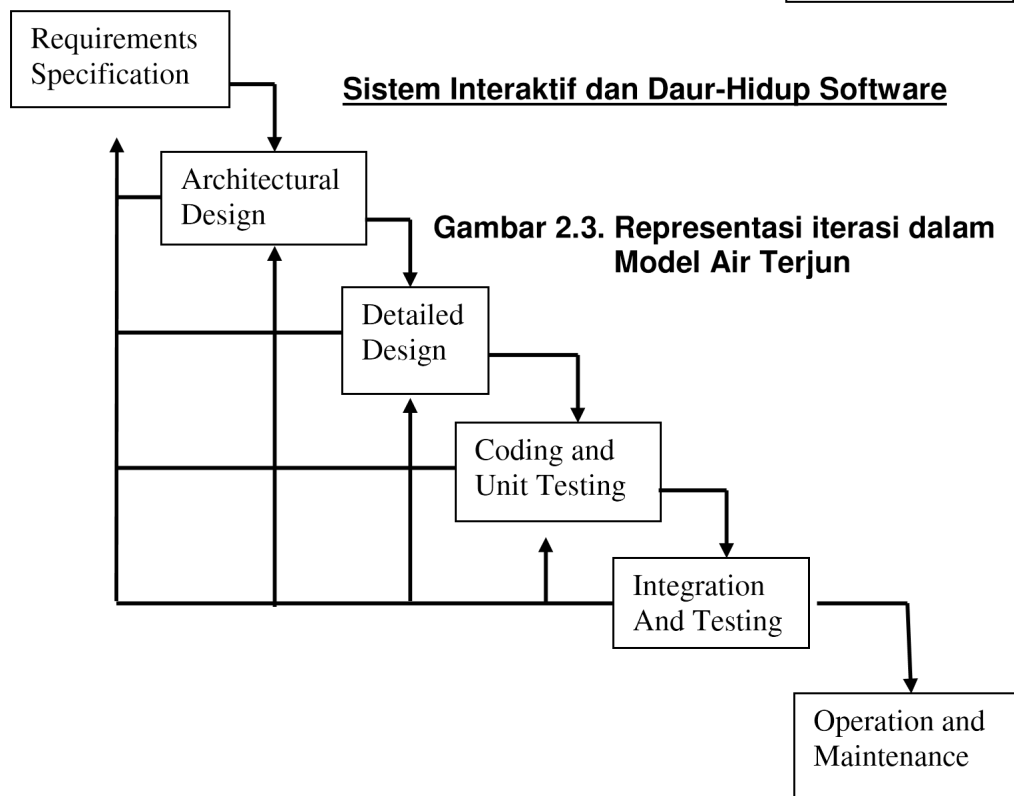
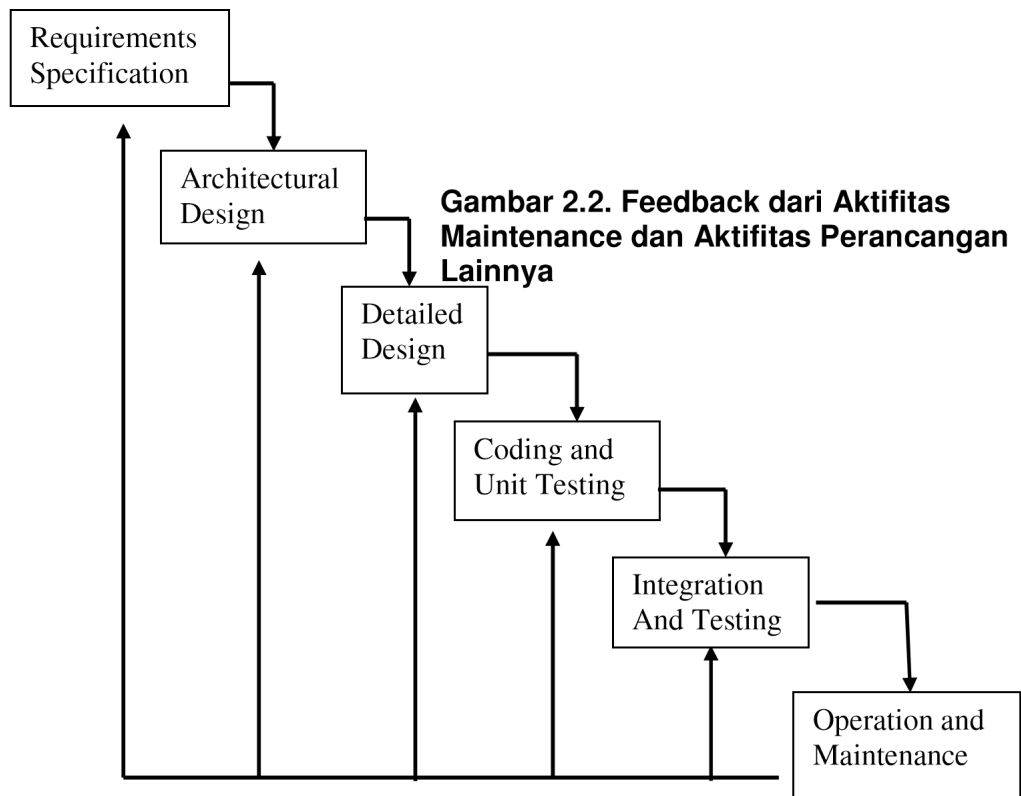
- ♦ Software engineering memberikan suatu cara untuk memahami struktur proses perancangan (desain), dimana proses tersebut dapat mendukung efektivitas perancangan sistem interaktif.
- ♦ Aturan-aturan perancangan (design rules) dalam bentuk standard dan guidelines memberikan arah perancangan, baik dalam 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. Aktivitas dalam Model Air Terjun dari Daur-Hidup Pengembangan Software

Validasi dan Verifikasi



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

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

Desain Iteratif dan Prototyping

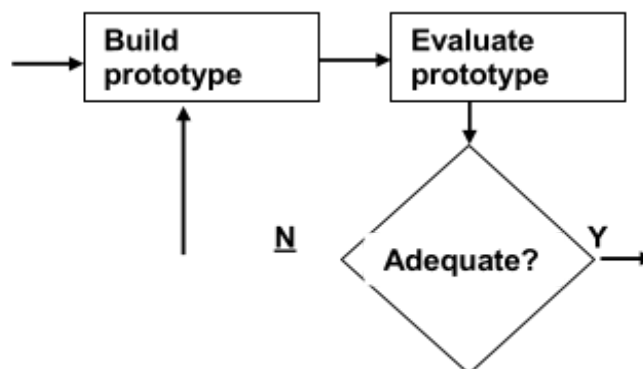
Tiga pendekatan utama prototyping:

- **Throw-away** : prototype dibuat dan dites. 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.

Preliminary
requirements

Tiga Metode (Pendekatan) Prototyping

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



Gambar 2.4. Throw-away Prototyping

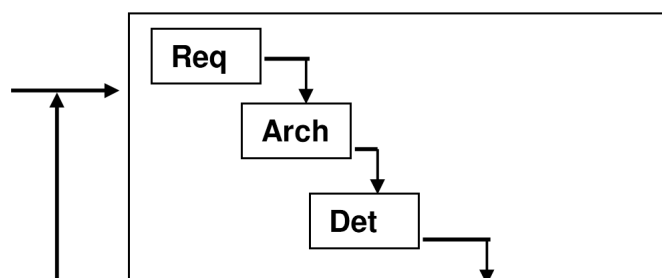
Final
requirements

- **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.

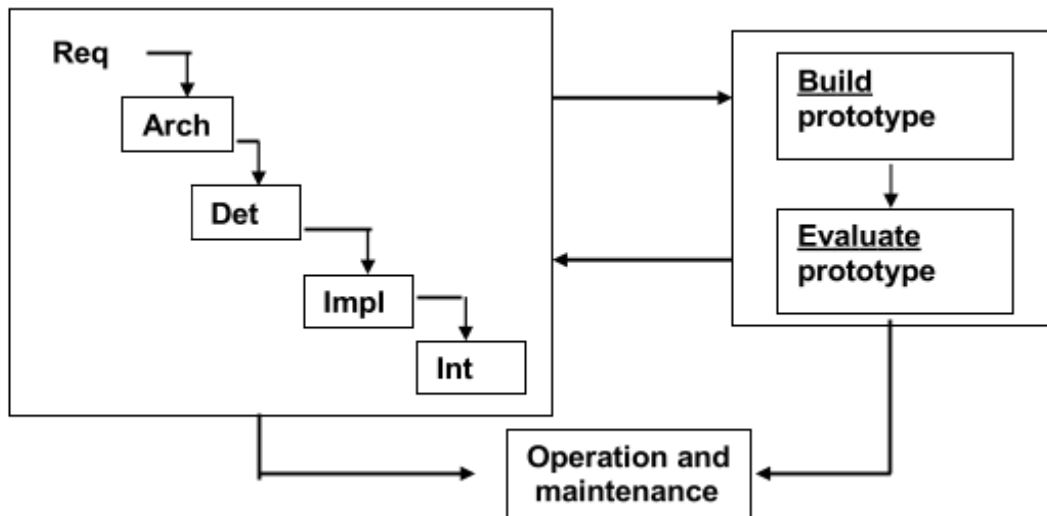
Identify
component

Designing

component/prototype



- **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

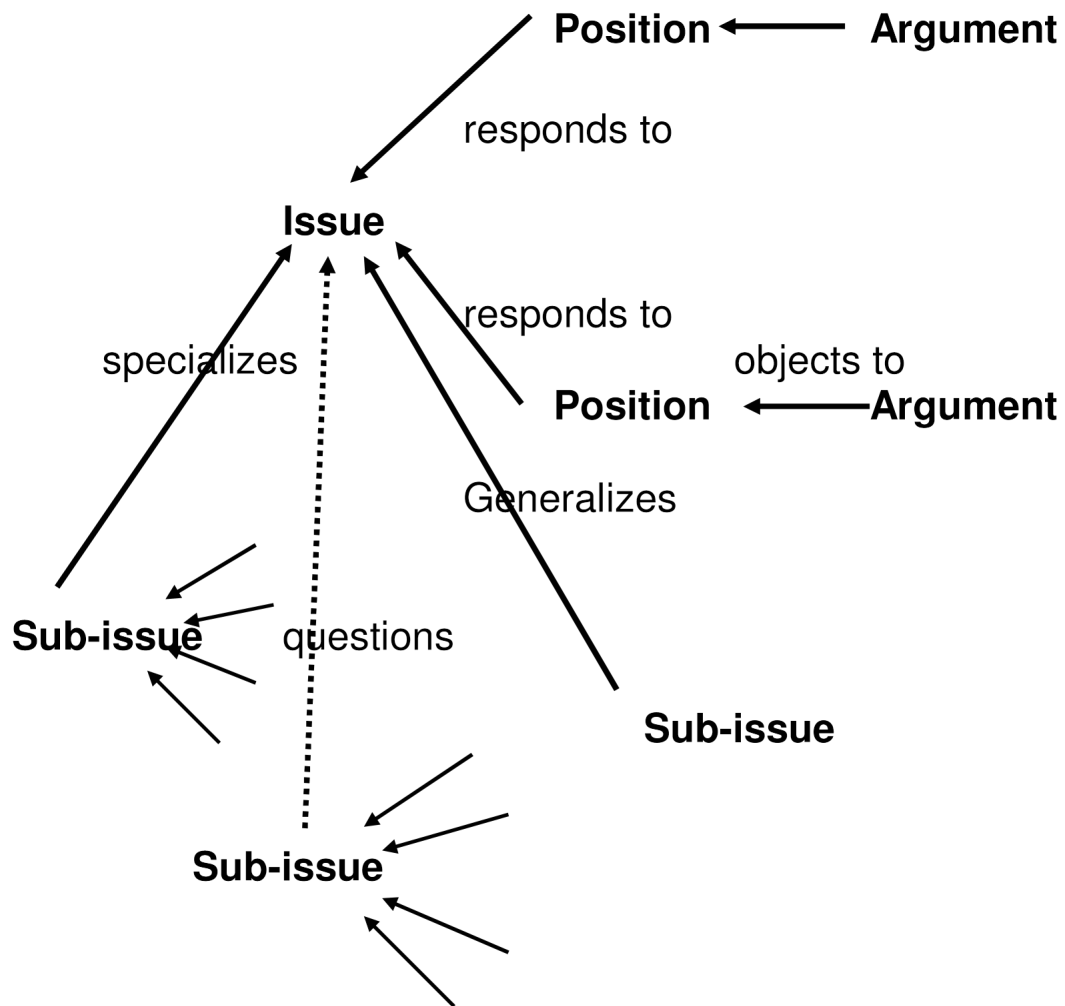
- Storyboards + Animation
- Limited Functionality Simulations
- High-Level Programming Support (Hypertalk)

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.



Gambar 2.7. The Structure of a gIBIS Design Rationale



