



Computer Science, University of Brawijaya

Putra Pandu Adikara, S.Kom

Interaksi Manusia dan Komputer

Tingkat Kegunaan (Usability)



- ❖ Paradigma
- ❖ Tingkat Kegunaan (Usability)
- ❖ Ukuran Tingkat Kegunaan (Usability)
- ❖ Pengukuran Tingkat Kegunaan (Usability)
- ❖ Prinsip Tingkat Kegunaan (Usability)
- ❖ Iterative Design untuk meningkatkan Tingkat Kegunaan (Usability)
 - Design
 - Prototype
 - Implement
 - Evaluate

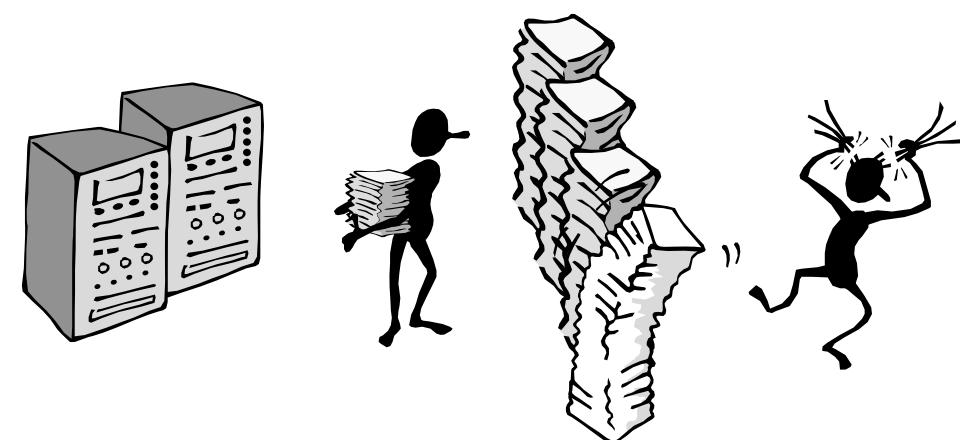


- ❖ Memahami paradigma dalam sistem interaktif
- ❖ Memahami tentang tingkat kegunaan dalam pengembangan perangkat lunak
- ❖ Memahami bagaimana mengukur tingkat kegunaan dalam pengembangan perangkat lunak
- ❖ Memahami tingkat kegunaan universal



Contoh Paradigma

❖ Batch Processing

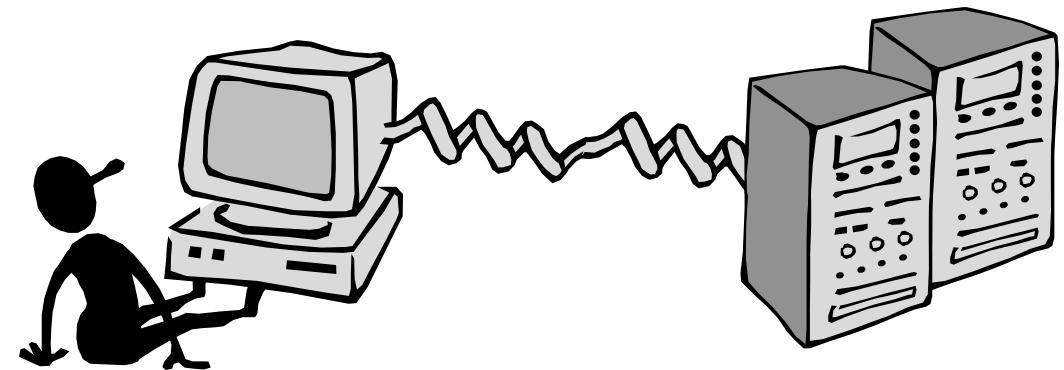


Impersonal computing



Contoh Paradigma

- ❖ Batch processing
- ❖ Time-sharing

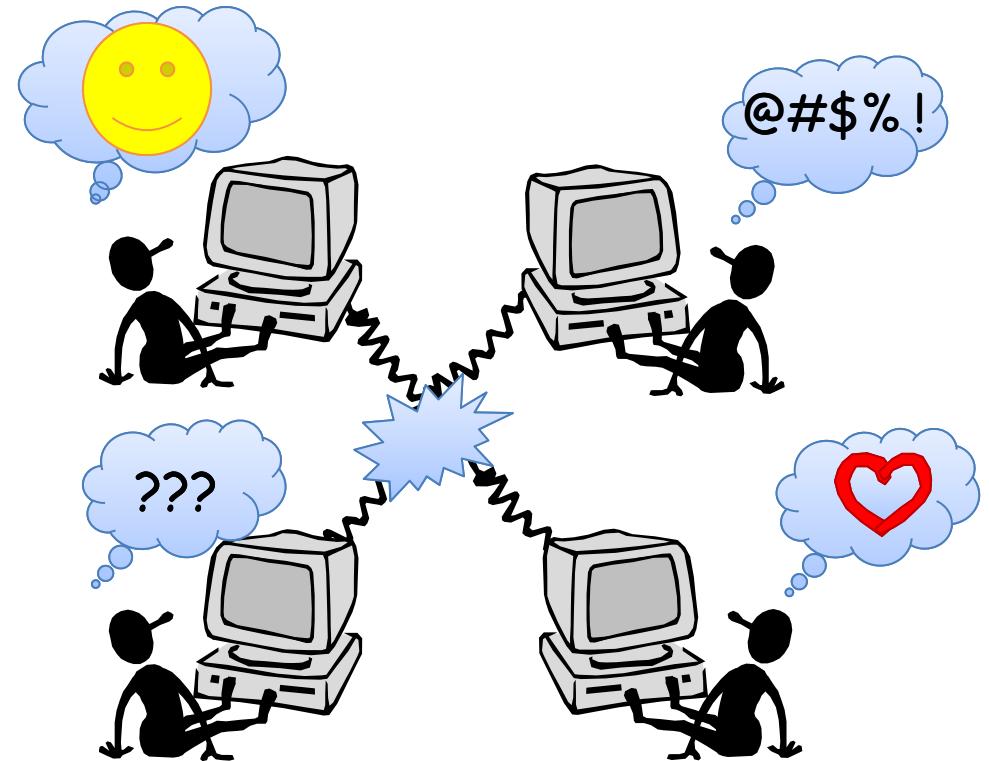


Interactive computing



Contoh Paradigma

- ❖ Batch processing
- ❖ Time-sharing
- ❖ Networking

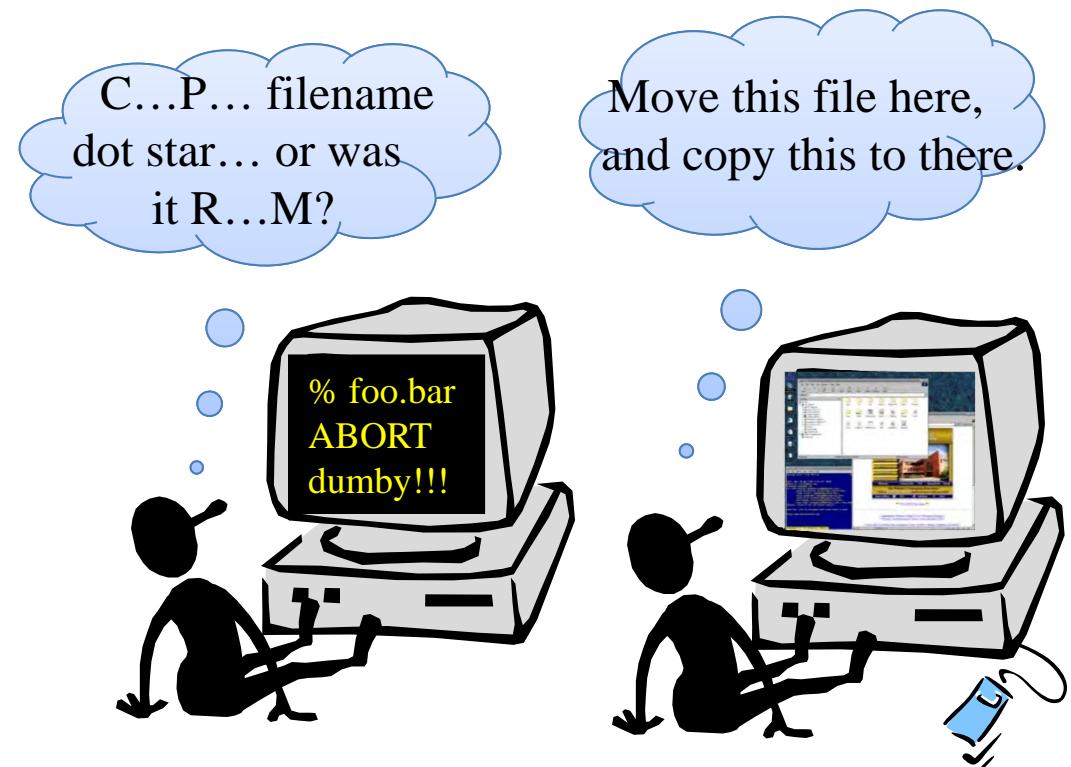


Community computing



- ❖ Batch processing
- ❖ Timesharing
- ❖ Networking
- ❖ **Graphical displays**

Contoh Paradigma



Direct manipulation



Contoh Paradigma

- ❖ Batch processing
- ❖ Timesharing
- ❖ Networking
- ❖ Graphical display
- ❖ Microprocessor

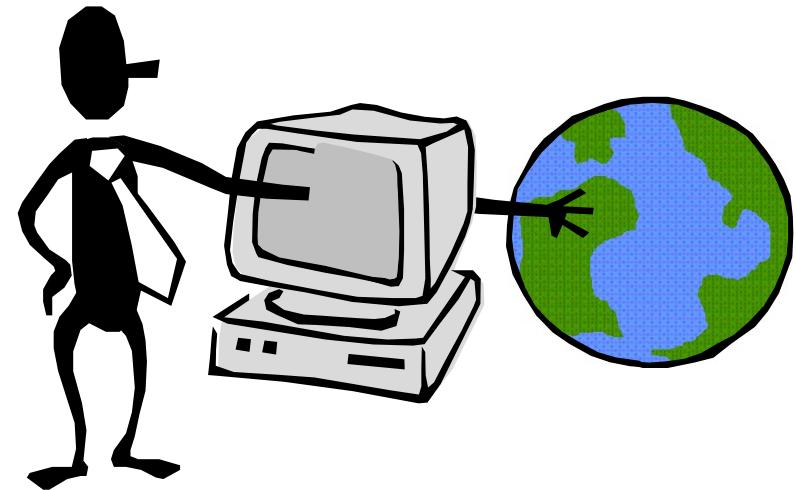


Personal computing



Contoh Paradigma

- ❖ Batch processing
- ❖ Timesharing
- ❖ Networking
- ❖ Graphical display
- ❖ Microprocessor
- ❖ **WWW**

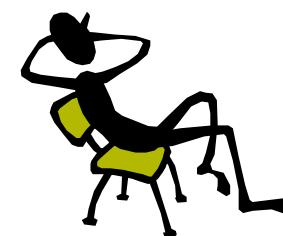
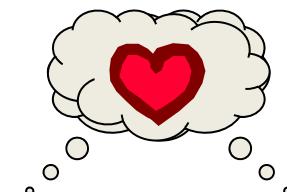


Global information



Contoh Paradigma

- ❖ Batch processing
- ❖ Timesharing
- ❖ Networking
- ❖ Graphical display
- ❖ Microprocessor
- ❖ WWW
- ❖ **Ubiquitous Computing**
- ❖ A symbiosis of physical and electronic worlds in service of everyday activities.





Paradigma – Time Sharing





Paradigma - Video Display Units





Paradigma - Programming Toolkits





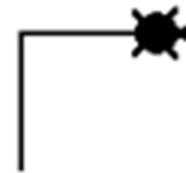
Paradigma - Personal Computing



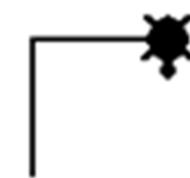
forward 50 right 90



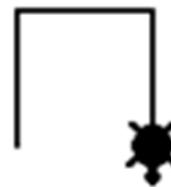
forward 50 right 90



forward 50 right 90



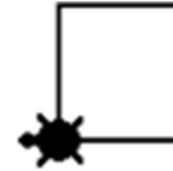
forward 50 right 90



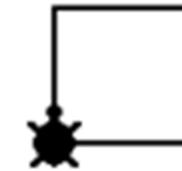
forward 50 right 90



forward 50 right 90



forward 50 right 90



forward 50 right 90



Paradigma - WIMP

UserView . static

```
Wednesday  
October 12, 1977  
10:49 am  
1314 disk pages
```

UserView . workspace

```
XEROX - Interim Research Group  
screen restore  
Smalltalk quit  
Changes  
Files  
Fonts  
Classes  
Messages  
Hierarchy
```

DocWindow . class

```
Class new title: 'DocWindow';  
subclassof Window;  
fields: 'document scrollBar editMenu'  
asfollows:
```

User events

```
windowToBe  
scrollbar and  
gain control
```

Event Responder

```
enter [self set  
editMenu  
leave [to: ur  
editMenu  
outside  
(editMenu  
scrollBar 3  
urraise)  
pendown [d  
keyboard [d  
Image  
show [super  
showDoc  
[document  
title [super  
Font 1
```

window . class

```
Class new title: 'Window';  
fields: 'frame';  
asfollows:
```

This is a superclass for presenting windows on the display. It holds content until the status is deposited outside. While it holds content, it distributes messages to itself based user actions.

Scheduling

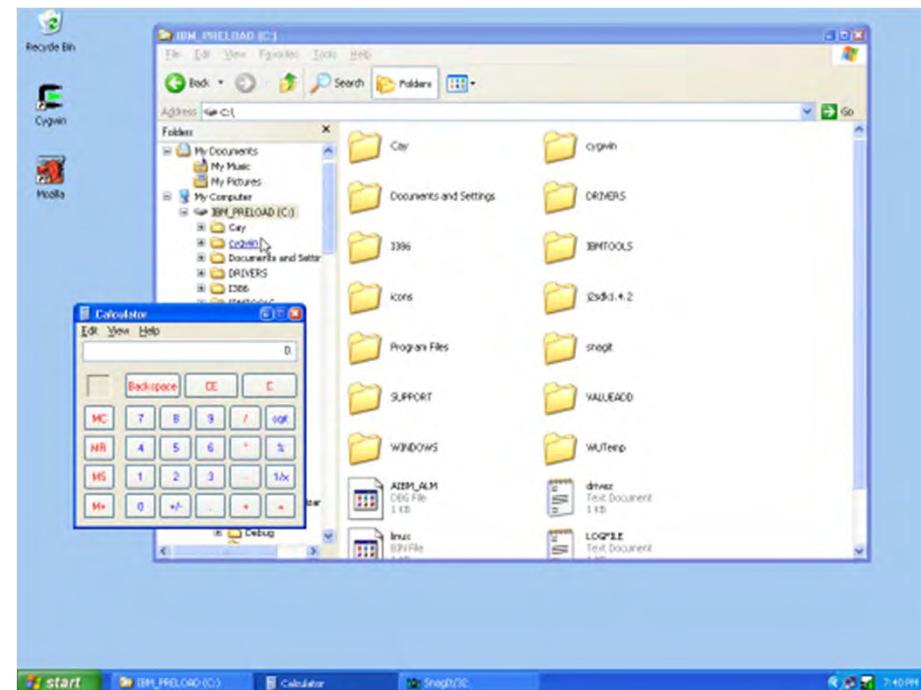
```
startup  
[frame contains: stylusArea  
[self enter.  
repaint  
[frame contains: stylusArea  
[Keyboard active [self isKeyboard]  
stylus down [self pendown]  
self outside []  
stylus down [self leave]]]  
urraise]
```

Default Event Responses

```
enter [-elp show]  
leave  
outside [urraise]  
pendown  
Keyboard [Keyboard next; frame flash]
```

Image

```
show  
[super activate]  
filename put [self title at: maxNameLength + titleLength  
filename complement]
```





Paradigma - Metafora

A screenshot of a desktop environment. At the top is a file manager window titled 'Places' showing a 'File System' view with folders like 'bin', 'boot', 'cdrom', and 'dev'. Below it is a budget spreadsheet titled '10th Worksheet' and '25th Worksheet', showing spending details across various categories like rent, phone, and credit cards, along with percentage breakdowns and subtotals.



Paradigma

- ❖ Manipulasi langsung
- ❖ Bahasa vs Aksi
- ❖ Hypertext
- ❖ Computer Supported Cooperative Work - CSCW



Usability

"Setiap desainer ingin membangun sistem interaktif berkualitas tinggi yang dikagumi oleh kolega, dirayakan oleh pengguna, beredar luas, dan sering ditiru."

(Shneiderman, 1992, hal.7)



Definisi Tingkat kegunaan (Usability)

- ❖ Tingkat Kegunaan → tingkat produk dapat digunakan yg ditetapkan oleh user utk mencapai tujuan scr efektif dan tingkat kepuasan dlm menggunakannya (ISO 9241).
- ❖ Tingkat Kegunaan → seberapa baik pengguna menggunakan fungsionalitas sistem
- ❖ Atribut Tingkat Kegunaan:
 - Efektivitas → ketelitian & kelengkapan di mana user mencapai tujuan
 - Efisiensi → sumber daya pembelajaran dlm hub dgn ketelitian dan kelengkapan utk user
 - Kepuasan → bebas dari ketidaknyamanan dan sikap positif dlm menggunakan produk



Ukuran Usability

❖ Dimensi/Faktor penentu suatu sistem itu *usable*:

- **Efektivitas**
 - ketelitian & kelengkapan di mana user mencapai tujuan
- **Learnabilitas**
 - mudah dipelajari user baru
- **Efisiensi**
 - sumber daya pembelajaran dlm hub dgn ketelitian dan kelengkapan utk user
- **Memorabilitas**
 - mudah digunakan dan perintahnya gampang diingat
- **Kesalahan**
 - Tingkat kesalahan yg kecil dan dapat diperbaiki
- **Kepuasan subjektif**
 - Sistem nyaman digunakan



Tingkat Kegunaan hanya satu atribut dari sistem

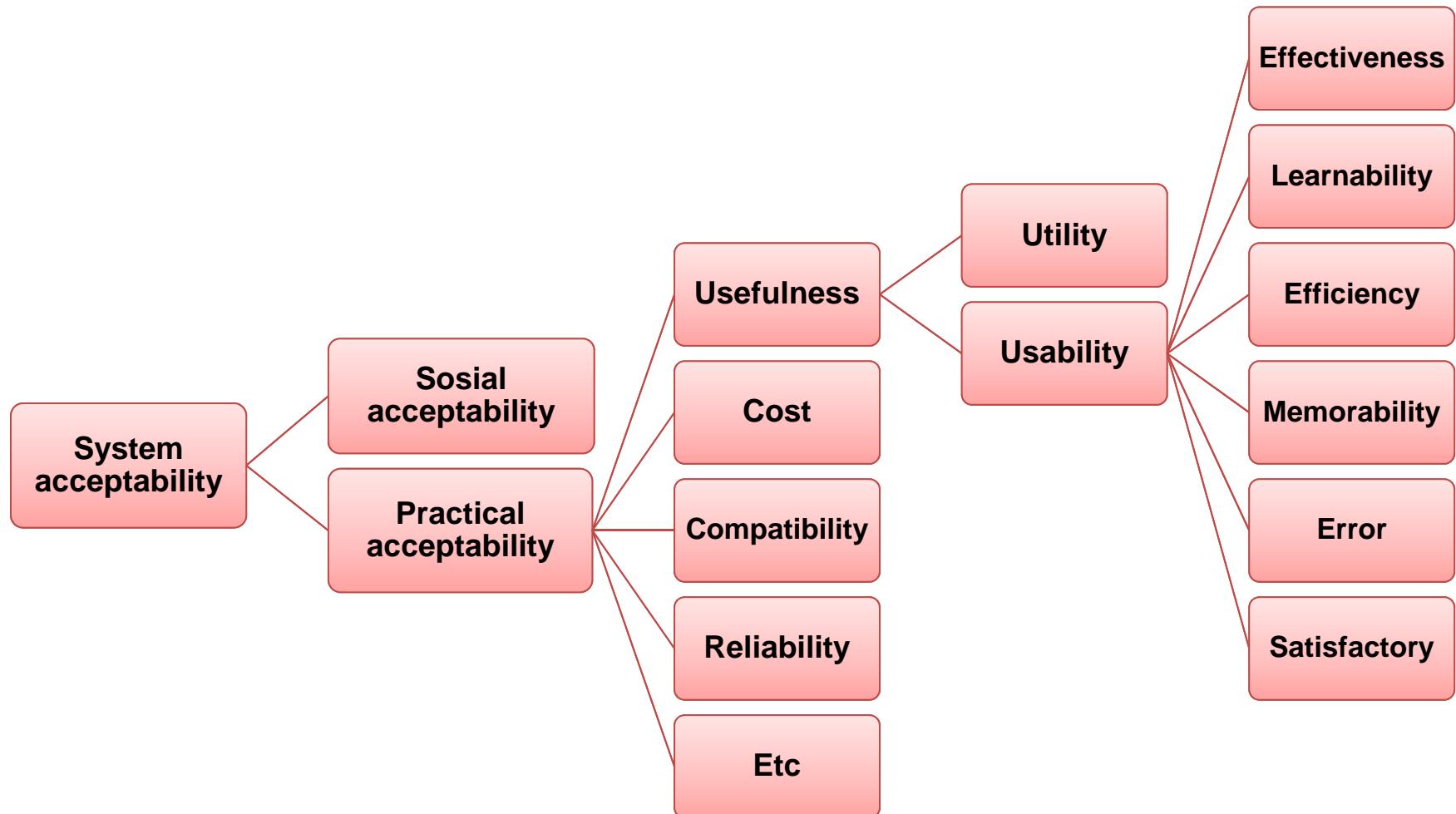
- ❖ Software desainer harus mempertimbangkan:
 - Functionality
 - Size
 - Cost
 - Security
 - **Usability**
 - Performance
 - Reliability
 - Standards

jadi tidak hanya usability

- ❖ Banyak keputusan dlm desain melibatkan pengorbanan di antara atribut-atribut lain



Akseptabilitas Sistem





Pengukuran Tingkat Kegunaan Produk

Usability objective	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 of learning
Error tolerance	Percentage of errors corrected successfully	Time spent on correcting errors	Rating scale for error handling



Ukuran Tingkat Kegunaan Berubah dlm Kepentingannya

- ❖ Tergantung pada pengguna
 - Pengguna pemula perlu *learnability*
 - Pengguna yg jarang memakai perlu *memorability*
 - Para ahli perlu *efisiensi*
- ❖ Tidak ada pengguna yg bisa disamakan sbg pemula atau ahli
 - pengalaman domain
 - pengalaman aplikasi
 - pengalaman feature
- ❖ **Karena itu kita harus tahu siapa pengguna sistem → User Analysis**
- ❖ **Apa yang dilakukan pengguna → Task Analysis**



Pentingnya UI

❖ ***“The interface is the system”***

- ❖ User interface sangat mempengaruhi persepsi perangkat lunak
 - perangkat lunak yg berguna terjual lebih baik
 - Situs web yg kurang berguna ditinggalkan
- ❖ Persepsi ini kadang-kadang dangkal
 - Users menyalahkan diri sendiri atas kegagalan UI
 - Orang-orang yang membuat keputusan utk membeli tidak selalu end-user



Resiko Kesalahan Desain

- ❖ Bila ada kesalahan dalam pembuatan produk:
- ❖ Waktu user tidaklah semakin murah, tapi semakin mahal
- ❖ Desain dengan benar sekarang, atau bayar kerugian yg besar nanti karena kesalahan desain!
 - Therac-25 radiation therapy machine menjadi standar studi kasus dlm informatika kesehatan dan rekayasa perangkat lunak, 6 pasien jadi korban akibat overdosis radiasi 100x dosis normal.



Mendesain UI itu susah

- ❖ Anda bukan pengguna
 - Kebanyakan software engineering adalah tentang berkomunikasi dengan programmer lain
 - UI adalah tentang berkomunikasi dengan pengguna
- ❖ Pengguna selalu benar
 - Masalah yg konsisten/selalu ada/sama adalah kesalahan sistem
- ❖ ...tetapi pengguna tidak selalu benar
 - Pengguna bukanlah desainer



Membuat UI itu susah

- ❖ User interface membutuhkan banyak usaha pengembangan perangkat lunak
- ❖ Perhitungan waktu UI ~ 50% utk:
 - Waktu desain
 - Waktu implementasi
 - Waktu pemeliharaan
 - Ukuran kode



Usability – Types of Systems

- ❖ Banyak interface dirancang dengan buruk dan hal ini berlaku di seluruh domain
 - Peran antarmuka "yang dirancang dengan baik" (dengan kriteria sebelumnya) dikenal dan diketahui
- ❖ Sistem yang berbeda memiliki kebutuhan yang berbeda jenis
- ❖ Life-critical, industri dan komersial, kantor dan rumah, eksplorasi, teknis-sosial



Usability – Types of Systems 1

❖ 1. Sistem life-critical

- Kontrol lalu lintas udara, reaktor nuklir, utilitas daya/energi, polisi & sistem alarm kebakaran
- Biaya tinggi, kehandalan dan efektivitas diharapkan
- Panjang periode pelatihan dapat diterima mengesampingkan biaya keuangan
 - Memberikan kinerja bebas kesalahan dan menghindari frekuensi rendah dan biaya tinggi kesalahan
- Perihal kepuasan kurang menjadi masalah karena pengguna termotivasi dengan baik

❖ 2. Industri dan komersial

- Perbankan, asuransi, order entry, inventory management, pemesanan, penagihan, dan sistem point-of-sales (POS)
- Kemudahan belajar adalah penting untuk mengurangi biaya pelatihan
- Kecepatan dan tingkat kesalahan relatif terhadap biaya
- Kecepatan kinerja itu penting karena jumlah transaksi
- Kepuasan subjektif cukup penting untuk membatasi operator kelelahan



Usability – Types of Systems 2

❖ 3. Aplikasi kantor, rumah, dan hiburan

- Word processing, email, conference, dan sistem video-game, paket pendidikan, search engine, perangkat mobile, dll
- Kemudahan belajar, tingkat kesalahan rendah, dan kepuasan subjektif adalah hal yang terpenting karena sering digunakan (menjadi kebebasan & persaingan sengit)
- Penggunaan yg jarang beberapa aplikasi
 - berarti antarmuka harus intuitif dan pentingnya online help yg mudah digunakan
- Memilih fungsionalitas itu sulit
 - populasi memiliki beragam baik pemula dan pengguna ahli
- Persaingan menyebabkan kebutuhan biaya rendah

❖ 4. Eksplorasi, kreatif, dan sistem kooperatif

- Web browsing, search engine, alat bantu seni, desain arsitektur, pengembangan perangkat lunak, musik komposisi, dan sistem pemodelan ilmiah
- Kerja kolaboratif
- Benchmark sulit untuk menggambarkan tugas eksplorasi dan perangkat
- Dengan aplikasi ini, komputer harus "menghilang" sehingga pengguna dapat diserap dalam domain tugas mereka



Usability – Types of Systems 3

❖ 5. Sistem teknis-sosial

- Sistem yang kompleks yang melibatkan banyak orang dalam jangka waktu panjang
 - Voting, health support, verifikasi identitas, pelaporan kejahatan
- Kepercayaan, privasi, tanggung jawab, dan keamanan adalah isu-isu penting
- Sumber yg dpt diverifikasi dan status umpan balik itu penting
- Kemudahan pembelajaran bagi pemula dan umpan balik untuk membangun kepercayaan
- Administrator perlu alat untuk mendeteksi pola-pola penggunaan yang tidak biasa



Universal Usability

- ❖ "Tingkat Kegunaan Universal" adalah tentang desain dan pengorbanan
 - "Mengakomodasi keragaman" adalah semata-mata mengenali adanya perbedaan yg fakta
 - Kadang-kadang dengan manfaat tak terduga
 - Dan, semakin banyak yang dapat menggunakannya, semakin banyak yang akan membelinya
- ❖ Singkatnya, ada perbedaan/variasi dalam:
 - Kemampuan fisik dan lingkungan kerja
 - Kemampuan kognitif dan persepsi
 - Perbedaan kepribadian
 - Keragaman budaya dan internasional
 - Pengguna yang memiliki keterbatasan
 - Pengguna lansia



Usability Principles



Prinsip Daya Guna

- ❖ Prinsip Tingkat Kegunaan dipengaruhi beberapa sifat:
 - **Learnability** → mudah dalam berinteraksi scr efektif dan mencapai performansi maksimal
 - **Flexibility** → menyediakan banyak cara bagi user dan sistem utk bertukar informasi
 - **Robustness** → tingkat dukungan yg diberikan agar user dpt menentukan keberhasilan atapun tujuan yg diinginkan



Prinsip Learnability

- ❖ Predictability
- ❖ Synthesizability
- ❖ Familiarity
- ❖ Generalizability
- ❖ Consistency



Prinsip Flexibility

- ❖ Dialogue initiative
- ❖ Multithreading
- ❖ Task migratability
- ❖ Substitutivity
- ❖ Customizability



Prinsip Robustness

- ❖ Observability
- ❖ Recoverability
- ❖ Responsiveness
- ❖ Task conformance



Usability Engineering adalah proses

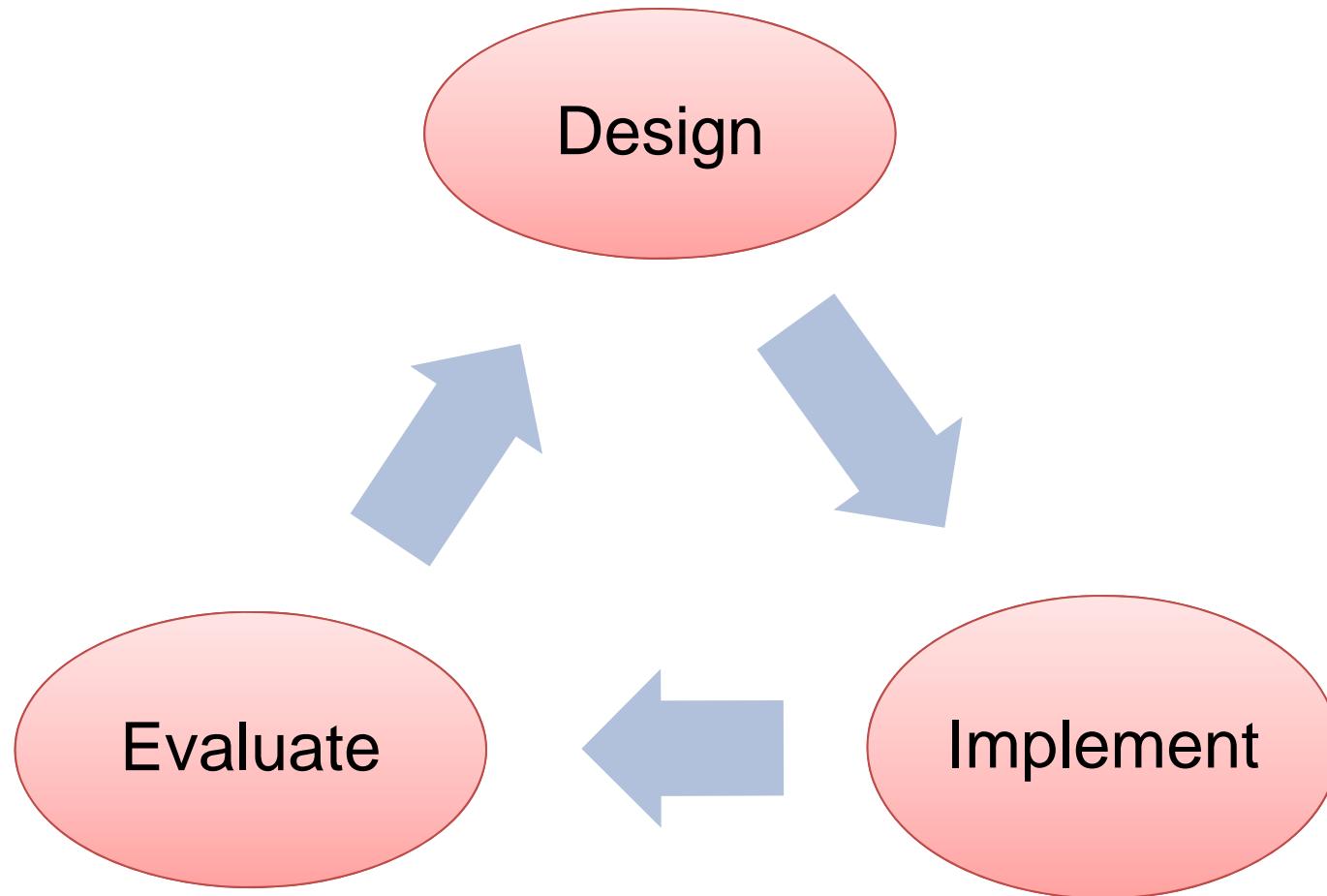


Rekayasa Tingkat Kegunaan

- ❖ Untuk meningkatkan Tingkat Kegunaan maka dilakukan Desain Iteratif/Berulang (Iterative Design)
- ❖ Setiap iteratif diharapkan akan Tingkat Kegunaan semakin baik dan meningkat



Iterative Design





Iterative Design

- ❖ Desain iteratif mengatasi masalah yang melekat akibat requirement yg tidak lengkap.
- ❖ kita tidak melakukan design-implement-evaluate sekali saja.
- ❖ Kita mengakui diri kita bahwa kita tidak akan melakukannya dengan benar pada percobaan pertama, dan memang merencanakan itu.
- ❖ Menggunakan hasil evaluasi, kita merancang ulang antarmuka, membangun prototipe baru, dan melakukan lebih banyak evaluasi.
- ❖ Akhirnya, mudah-mudahan, proses menghasilkan antarmuka yang cukup berguna.



Design

- ❖ Analisis Pengguna dan Tugas (*User & Task Analysis*)
 - ketahui penggunanya
 - Ketahui domainnya
- ❖ Prinsip Desain (*Design Principles*)
- ❖ Standar Desain (*Design Standards*)
- ❖ Pedoman Desain (*Design Guidelines*)
 - Menghindari kesalahan-kesalahan bodoh
 - Mungkin samar atau bertentangan



Prototyping

❖ Prototipe

- mensimulasikan atau menghidupkan beberapa fasilitas dari sistem dimaksudkan
- jenis prototipe:
 - Throw-away
 - Incremental
 - Evolutionary

❖ Isu-isu pengelolaan:

- Waktu
- Perencanaan
- Fitur non-fungsional
- Kontrak



Implement

❖ Prototyping

- Murah, implementasi throw away
- Low-fidelity: kertas, Wizard of Oz
- Medium-fidelity: HTML, Visual Basic

❖ Teknik Implementasi GUI

- Input/output model
- Toolkit
- UI builder



Evaluate

- ❖ Evaluasi menempatkan prototipe untuk diuji
- ❖ Evaluasi ahli
 - Heuristik dan Walkthroughs
- ❖ Evaluasi yg diperkirakan
 - Pengujian terhadap sebuah model rekayasa (simulasi pengguna)
- ❖ Evaluasi empiris
 - Melihat pengguna menggunakannya



- ❖ Lebih detil akan dijelaskan pada pertemuan berikutnya



Referensi

- ❖ Jakob Nielsen. *Usability Engineering*. 1993. Academic Press.
- ❖ *Human Computer Interaction 3rd Ed.*
<http://www.hcibook.com/e3/>