

PROSES PERANCANGAN INTERAKSI (MODEL SIKLUS HIDUP DAN PROSES)

- Mengapa perancangan interaksi menghasilkan interface yang buruk?
 - Perancang terlalu memperhatikan ke fungsi dibanding penggunaan
 - Perancang tidak mempunyai pemahaman yang cukup dalam merancang interface
 - Rancangan yang baik tidak mudah – tidak sekedar masalah warna, layout maupun penggunaan ikon
 - Perancang merancang untuk dirinya sendiri dan menggenalisir yang lain
 - Perancang merencanakan untuk "menambah interface yang baik " di akhir proses perancangan namun kemudian kehabisan waktu
 - User selalu "toleran" terhadap interface yang buruk.

Perancangan berpusat pada **pengguna** memerlukan :

1. Berfokus pada ketrampilan user
Mendukung pencapaian tujuan dan sasaran yang diinginkan
2. Mengembangkan kriteria penggunaan secara spesifik
Identifikasi dokumen dan penggunaannya secara khusus dan harapan pengguna yang berpengalaman, dan lakukan "uji regresi penggunaan"
3. Gunakan aturan yang terukur
Reaksi user dan skenario penggunaan petunjuk, simulasi, dan prototipe diawasi, dicatat dan dianalisa terhadap pengguna lain
4. Berulang
Nothing is perfect first time. Rancang, bangun, evaluasi dan kemudian rancang ulang, bangun kembali, re-evaluasi.

- Mengapa mementingkan user :
 - Apakah sistem membantu pencapaian tujuan?
 - Apakah mudah dalam berinteraksi dengan sistem?
- Mengelola harapan / keinginan :
 - Apakah keinginannya cukup realistis dibanding kemampuannya
 - Tidak ada kejutan, tidak mengecewakan
 - Pelatihan yang cukup waktu
 - Komunikasi yang intens

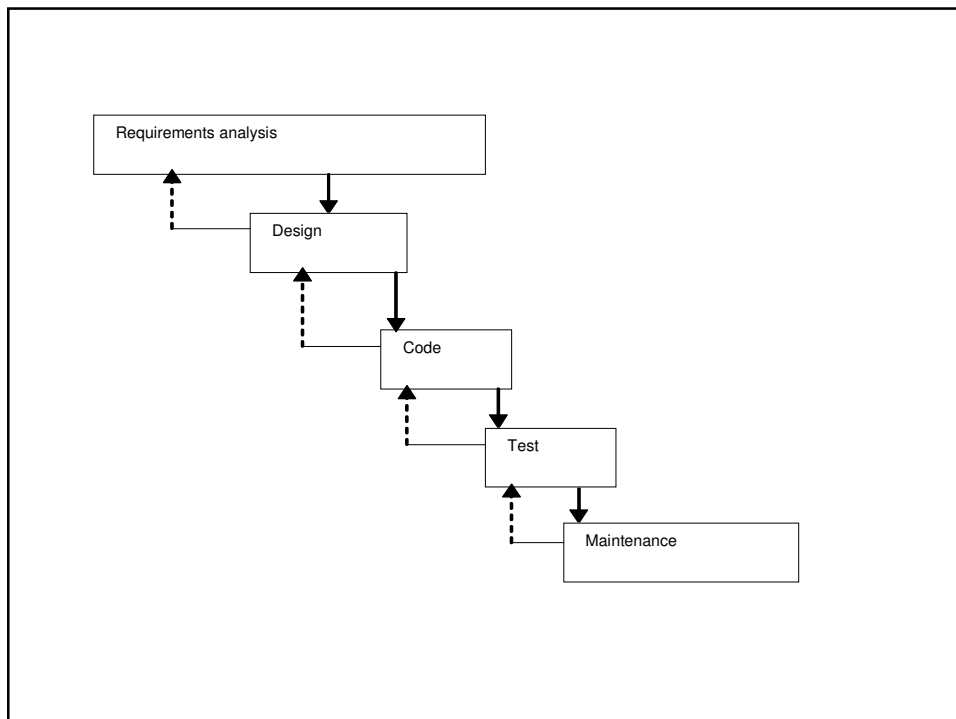
- Kembangkan "kepemilikan"
 - Anggap pengguna sebagai pemilik
 - Lebih menerima dan memaklumi segala masalah
 - Dapat memberi perbedaan yang cukup untuk produk baru
- Siapakah pengguna (user)?

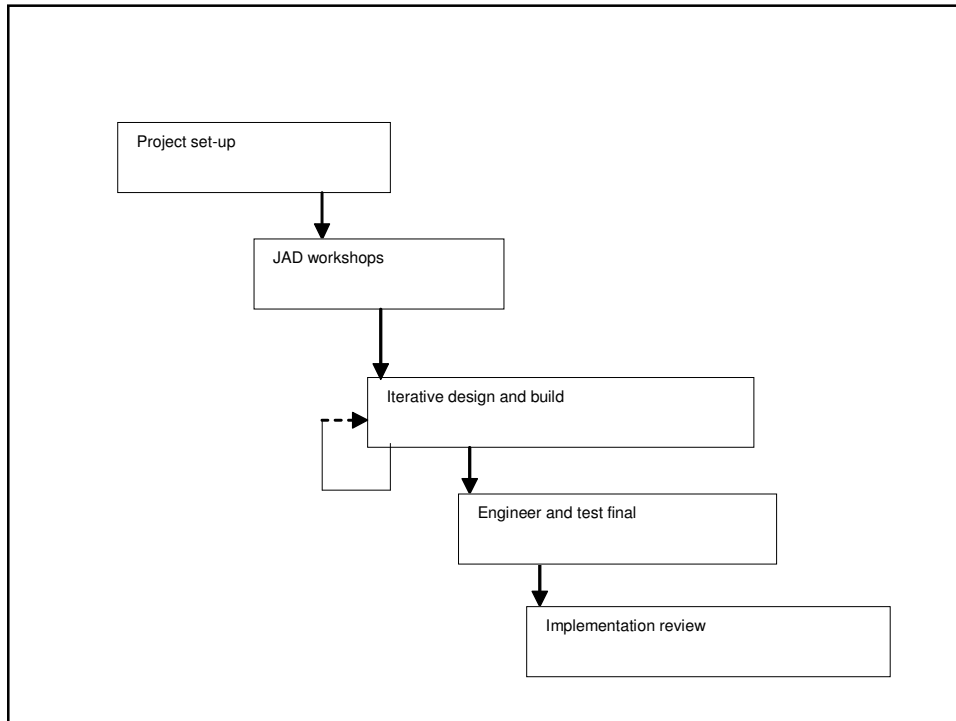
Mereka yang :

 - Bekerja menggunakan sistem untuk mencapai tujuan secara terus-menerus dan terampil

Tidak rutin dan tidak terampil

 - Mengelola user langsung
 - Memerlukan pengembangan diri berulang





- • **Siklus Hidup Untuk Pengembangan (RAD : Linier Sequential)**
 - Versi cepat dari waterfall, dengan pengembangan modular
 - Menggunakan Joint Application Development
 - Yang paling dibutuhkan user adalah workshop JAD

- **V-Model**

- Dikembangkan di Jerman untuk aplikasi pertahanan
- Perlu tuntunan untuk pengujian jika terjadi perbedaan (Newman & Lamming, 1995)

Verifikasi : apakah sudah bekerja seperti yang diharapkan ? (uji dengan rancangan semula)

Validasi : apakah rancangan seperti yang diinginkan ? (uji dengan daftar kebutuhan)

- Secara umum tes diperlukan untuk mendeteksi perbedaan

- Model siklus hidup yang berpusat pada user

- Pengulangan selalu dibutuhkan jika kebutuhan untuk user belum terpenuhi "ketidaktahuannya" tanpa harus memberikan software dengan lingkungan yang baru.
- Resiko pada setiap tahap dalam pengembangan dapat dikurangi dengan memahami kebutuhan user.

- **Penggunaan Model Siklus Hidup Rekayasa (Mayhew,1999)**

Termasuk teknik yang berpusat pada pemakai pada setiap tahap.

Kebutuhan yang penting :

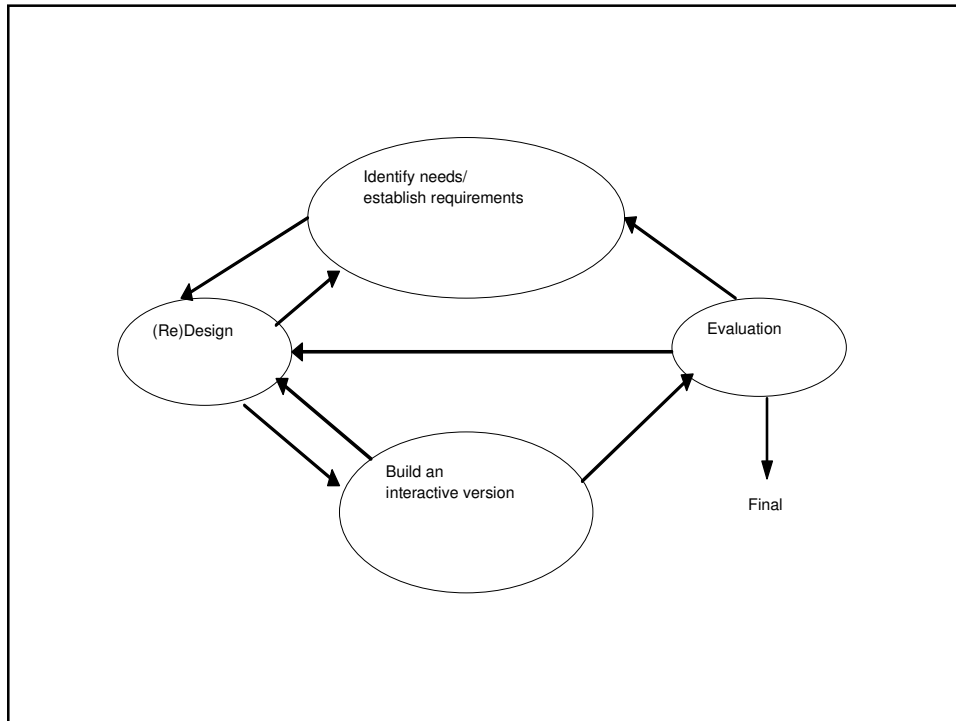
- Pandangan umum tentang penciptaan kegunaan
- Mempunyai hubungan dengan pendekatan perancangan software

Tahap pengidentifikasian kebutuhan, perancangan, evaluasi, prototyping

- Dapat dipecah menjadi proyek-proyek yang lebih kecil
- Menggunakan suatu tuntunan untuk menangkap tujuan yang diharapkan

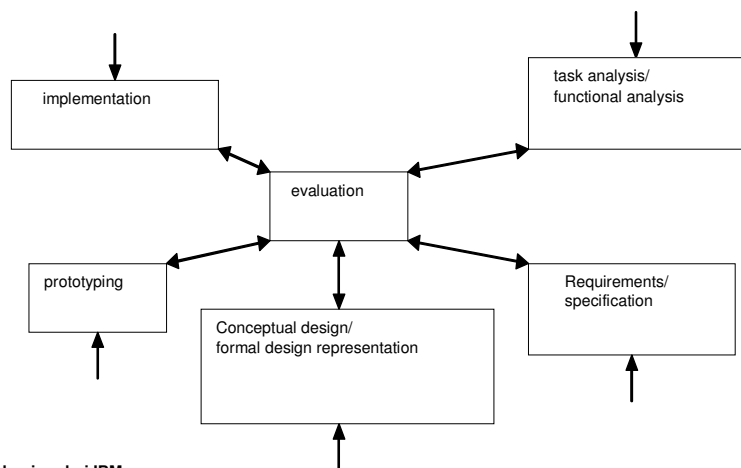
- **Model Rancangan Interaksi Sederhana**

- Satu titikan masukan
- Rancangan menghasilkan prototipe yang interaktif yang dapat dievaluasi
- Evaluasi dapat dilakukan dimana saja
- Evaluasi harus dikaitkan dengan hasil akhir



• Model Siklus Hidup Star (Hartson & Hix, 1989)

3. Pengembangan dari pengamatan perancangan interface



- Efek domino dari IBM
- Mendefinisikan pasar

Efek domino dari IBM

- - Mendefinisikan pasar
- Fasilitas yang penting :
 1. Tidak kegiatan kecil yang harus selalu urut. Pengembangan dapat dilakukan di satu bagian saja
 2. Pengujian dilakukan terus-menerus, tidak harus diakhir
- Tentukan sasaran yang akan dibidik, identifikasi pesaing, dan temukan kebutuhan umum dari user dan pastikan bahwa produk akan sukses.
- Metode : ajukan proposal pada sasaran untuk mengetahui sejauh mana keinginan untuk mendapatkan produk yang baru, identifikasi prioritas dan solusi yang dilakukan saat ini.

- Analisa
- Identifikasi kemampuan user, strategi yang digunakan untuk meningkatkan ketrampilannya, alat yang saat ini dipakai, masalah-masalah yang dialami, perubahan yang diinginkan baik dalam ketrampilan maupun peralatan.
- Metode : tanya kemampuan user dan buat daftar dengan skala prioritas, observasi ketrampilan di lapangan.

- Evaluasi kompetisi
- Tentukan kekuatan dan kelemahan rancangan
- Metode : pengguna diminta untuk mencoba menggunakan berbagai produk dan minta untuk menyebutkan kelebihan dan kelemahan dari masing-masing produk.
-

- Rancang sambil jalan
- Gunakan hasil analisa untuk membuat alternatif solusi, minta masukan sampai dengan penentuan pilihan yang terbaik.
- Metode : tanyai user sehubungan dengan pengalaman menggunakan prototipe

- Evaluasi dan validasi
- Secara periodik user memberikan masukan selama pengembangan dan perancangan akan diulang berdasarkan masukan tadi.
- Metode : amati kebutuhan pokok user dalam menggunakan sistem.
- Benchmark
- Memadukan hal-hal terbaik yang dimiliki pesaing untuk diterapkan dalam sistem yang dibangun
Metode : menggali informasi dari user hal-hal yang sebaiknya ada dibandingkan dengan kompetitor, contoh : situs IBM.

- **Paradigma dan prinsip dalam IMK**
- Bagaimana merancang sistem supaya dapat berdaya guna besar?
Sejarah pengembangan sistem interaktif masa lalu menjadi paradigma dalam pengembangan sistem baru. Prinsip keberdayaagunanya sistem lebih penting dari penggunaan sistem itu.

- Pemahaman
- Bagaimana sebuah sistem yang interaktif dapat dikembangkan untuk lebih berdayaguna
- Bagaimana kegunaan dari sebuah sistem yang interaktif dapat didemonstrasikan atau diyakinkan.

Pendekatan

- Paradigma kegunaan, contoh tentang keberhasilan teknik interaktif
- Prinsip kegunaan, teori ditentukan oleh pengetahuan psikologi, perhitungan dan sosiologi.

1. Paradigma kegunaan

- Perspektif sejarah dalam merancang sistem yang interaktif - Berdasar waktu
- Tahun 40-50 an terjadi ledakan pertumbuhan teknologi
- Tahun 60-an kebutuhan akan kekuatan jaringan
- ARPA menerapkan teknik jaringan yang memungkinkan sebuah komputer dapat diakses banyak orang/pengguna

- Unit peraga gambar
- Media yang lebih baik daripada kertas 1962 – komputer digunakan untuk menampilkan dan mengolah data. Berkat jasa seseorang dapat mengubah sejarah perkomputeran secara drastis

- Peralatan pemrograman :
 - Engelbart dari Lembaga Riset Stanford
 - 1963 – bertambahnya kecerdasan manusia
 - 1968 – NLS/bertambahnya nilai sistem. Peralatan pemrograman semakin lengkap untuk membangun sistem interaktif yang semakin kompleks - Komputer Pribadi (PC)
 - 70 – an bahasa pemrograman LOGO yang mudah digunakan anak untuk membuat gambar. Sistem semakin baik namun semakin mudah dioperasikan
 - Selanjutnya komputer makin kecil, namun semakin canggih
 - Munculnya notebook sebagai alternatif dari PC

- Sistem Window dan antarmuka WIMP
- Manusia dapat melakukan lebih dari satu pekerjaan pada waktu yang sama
- Window dipakai sebagai alat dialog untuk berganti topik 1981 XeroxStar memelopori sistem window untuk tujuan komersil Wimp saat ini sudah menjadi mekanisme interaksi yang biasa.

- - Proses perubahan
Menghubungkan komputer dengan aktivitas dunia nyata adalah teknik pengajaran yang baik.
Permainan LOGO
Pengelolaan file pada komputer kantor
Penggunaan pengolah kata untuk menetik
Analisa keuangan dengan spreadsheet
- Masalah
Beberapa fungsi tidak berjalan seperti seharusnya karena memerlukan perubahan budaya

- Manipulasi langsung

1982 – Schneiderman memperkenalkan interaksi berbasis grafis Objek yang terlihat

Menambahkan kemampuan dan tanggapan yang cepat

Memungkinkan pembetulan dalam semua kegiatan

Menggantikan bahasa dengan aksi

1984 – Apple Macintosh

What you see is what you get (WYSIWIG)

- Bahasa dan aksi

Aksi tidak selalu berupa kata-kata

DM – antar muka menggantikan paradigma bahasa sistem

Antarmuka sebagai mediator

Antarmuka berlaku seperti agen yang pintar

Pemrograman adalah contoh penyatuan aksi dengan bahasa

- - Hypertext
- 1945 – Vannevar Bush dan Memex sebagai kunci sukses dalam menangani ledakan informasi Pertengahan 60an Nelson menjelaskan hypertext sebagai struktur penelusuran nonlinear Hypermedia dan multimedia
- Proyek Xanadu Nelson masih merupakan mimpi
- - Multi modal
- Modelnya adalah saluran komunikasi manusia
- Pada waktu yang bersamaan harus mampu menjadi saluran input/output
- -

- Komputer yang dapat mendukung pekerjaan
- Menghilangkan istilah single user/single computer
- Tidak mempunyai dampak sosial yang panjang
- Email adalah salah satu bukti sukses

2. Prinsip mendukung kegunaan

Sebuah presentasi terstruktur tentang prinsip-prinsip umum yang diterapkan selama proses perancangan sistem interaktif - Dapat dipelajari

Seorang pengguna baru dapat segera mulai efektif berinteraksi dengan baik.

- **Fleksibilitas**

Memungkinkan pengubahan sistem informasi dengan berbagai cara

- **Kekuatan**

Menjamin tercapainya tujuan pengguna sesuai yang diharapkan

a. Prinsip Pembelajaran

- **Kemungkinan peramalan**

Memperkirakan kejadian masa depan berdasarkan sejarah masa lalu.

Kelayakan operasional

- **Kemungkinan melakukan sintesa**

Menangani dampak dari kegiatan masa lalu

Segera vs kejadian yang sebenarnya

- **Berlaku umum**

Menyediakan pengetahuan interaksi secara khusus dalam suasana baru.

- **Konsisten / tetap**

Dalam proses input/output dalam segala suasana.

b. Prinsip Fleksibilitas

- Inisiatif dialog
Kebebasan sistem dalam dialog untuk input data Sistem vs pengguna lama.
- Multithreading
Mampu mendukung sistem untuk melakukan beberapa pekerjaan secara bersamaan
Concurrent vs interleaving
- Proses bisa dipindahtempatkan
Selalu tanggap terhadap perintah yang diberikan user kepada sistem
- Penggantian
Memungkinkan nilai yang sama dari input dan output untuk dipertukarkan satu dengan yang lain
- Mudah disesuaikan
Dapat dimodifikasi sesuai lingkungan atau sistem

• Prinsip ketegasan

- **Kemampuan mengamati**
User dapat melakukan evaluasi internal sistem dari gambaran yang dia peroleh. Memungkinkan melakukan penelusuran, standarisasi, kedayatahanan, mudah dioperasikan
- **Perlindungan**
Memungkinkan user untuk melakukan perbaikan apabila ditemukan kesalahan.
Dapat dilakukan sendiri, perlindungan kedepan dan kebelakang, seimbang
- **Tanggap**
Bagaimana user mampu menerima komunikasi dengan sistem
- **Kemampuan menyesuaikan diri**
Mampu memberi pelayanan kepada berbagai tingkatan user
Tugas lengkap dan sesuai