

Interaksi Manusia dan Komputer

Desain

Oleh :

ANISYA, M.KOM

Teknik Informatika

Fakultas Teknologi Informatika

ITP

2013



SUB TOPIK

- Tujuan dari Desain
- Cara pendekatan
- Prinsip desain
- Ide dalam membuat desain

Tujuan dari Pembelajaran :

- Dapat mengerti dan menjelaskan prinsip desain
- Dapat mengerti pentingnya ide dan mengetahui cara mendapatkan ide dalam proses pembuatan design
- Dapat mengerti dan mampu menjelaskan tantangan dalam membuat design yang baik dan memiliki nilai saing yang baik.
- Dapat menyebutkan beberapa contoh disain dalam kehidupan sehari-hari.
- Dapat menyebutkan dan mampu menerangkan konsep disain grafik (Filosofi Desain, Prinsip Desain Grafik, Pembuatan User Interface, Teknik Koding, Tipografi, fonts, Warna, Desain Icon)



Tujuan dari desain HCI ini =

suatu teknik yang andal
untuk rancangan yang
menghasilkan suatu sistem
yang dapat dipakai.

Isu dari HCI yang relevan
disemua bagian adalah
siklus hidup perangkat lunak.



Bagi perancang antarmuka, hal yang sangat penting untuk ia perhatikan adalah bahwa ia sebaiknya (atau seharusnya) mendokumentasikan semua pekerjaan yang ia lakukan. Dengan dokumentasi yang baik, ia dapat mengubah rancangannya ketika ia berubah pikiran atau menemukan bahwa rancangannya tidak mudah diimplementasikan. Perubahan yang terjadi bisa karena usulan pengguna yang berubah atau karena alasan lain.



Dokumentasi rancangan dapat dikerjakan atau dilakukan dengan beberapa cara :

1. Membuat sketsa pada kertas
2. Menggunakan peranti purwarupa GUI
3. Menuliskan keterangan yang menjelaskan tentang kaitan antara satu jendela dengan jendela yang lain.
4. Menggunakan peranti bantu yang disebut CAE.

Cara kedua dan cara keempat ini jarang untuk digunakan karena mahal, jadi cara ini digunakan untuk pembuatan antarmuka grafis untuk suatu pekerjaan dengan skala besar.



Cara pendekatan

Program aplikasi pada dasarnya dapat dikelompokkan dalam 2 kategori besar yakni program aplikasi untuk keperluan khusus dengan pengguna yang khusus pula (special purpose software) dan program aplikasi yang akan digunakan oleh pengguna umum (general purpose software / public software)



Pada kelompok pertama, yakni pada program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk keperluan khusus, misalnya program aplikasi untuk inventori gudang, pengelolaan data akademis mahasiswa, pelayanan reservasi hotel, dan program-program aplikasi yang serupa, kelompok calon pengguna yang akan memanfaatkan program aplikasi tersebut dapat dengan mudah diperkirakan, baik dalam hal keahlian pengguna maupun ragam antarmuka yg akan digunakan.

Dalam kelp ini ada satu pendekatan yakni dengan pendekatan perancangan berpusat-ke-pengguna (user-centered design approach).



perancangan berpusat-ke-pengguna (user-centered design approach). Berbeda dengan user design approach.

UCDA ini melibatkan pengguna, tetapi bukan dalam hal si pengguna memikirkan bagaimana implementasinya nanti, tetapi pengguna diajak aktif untuk berpendapat ketika perancang antar muka sedang menggambar “wajah” antarmuka yang mereka inginkan. Jadi disini si pengguna dan perancang antarmuka duduk bersama-sama untuk merancang wajah antarmuka yang diinginkan pengguna. Pengguna menyampaikan keinginannya sementara perancang menggambar keinginan pengguna tersebut sambil menjelaskan keuntungan dan kerugian wajah antarmuka yang diinginkan oleh pengguna. Dengan cara ini si pengguna sudah mendapat gambaran nyata mengenai antarmuka yang nanti akan mereka gunakan.

Pada pendekatan user design approach, pengguna sendirilah yang merancang wajah antarmuka yang diinginkan. Di satu sisi, cara ini akan mempercepat proses pengimplementasian modul antarmuka. Tetapi di sisi yang lain, hal ini justru sangat memberatkan pemrogram karena apa yang diinginkan pengguna belum tentu dapat diimplementasikan dengan mudah atau bahkan tidak dapat dikerjakan dengan menggunakan peranti yang ada.



Untuk public software, perlu menganggap bahwa program aplikasi tersebut akan digunakan oleh pengguna dengan berbagai tingkat kemampuan dan karakteristik yang sangat beragam. Disatu sisi keadaan ini dapat dikatakan “memaksa”, memaksa pengguna untuk menerima hasil rancangan, namun hal ini akan berdampak pada tidak banyaknya pengguna thdp aplikasi ini. Satu kunci penting dalam pembuatan model antarmuka untuk program-program aplikasi pada kelompok ini adalah customization. Dengan customization, pengguna dapat menggunakan program aplikasi dengan wajah antarmuka yang sesuai dengan selera masing-masing pengguna.



Prinsip Desain

- Model pengguna
- Bahasa perintah
(command language)
- Umpan balik
- Penampil informasi



Model Pengguna

Model mental pengguna merupakan model konseptual yang dimiliki oleh pengguna ketika ia menggunakan sebuah sistem atau program aplikasi.

Model ini memungkinkan seorang pengguna untuk mengembangkan pemahaman mendasar tentang bagian yang dikerjakan oleh program.

Dengan pertolongan model itu pengguna dapat mengantisipasi pengaruh suatu tindakan yang ia lakukan dan dapat memilih strategi yang cocok untuk mengoperasikan prog tersebut.

Model pengguna dapat berupa suatu simulasi tentang keadaan yang sebenarnya dalam dunia nyata, sehingga ia tidak perlu mengembangkannya sendiri dari awal.



Command Language

Setelah pengguna mengetahui dan memahami model yang ia inginkan, dia memerlukan peranti untuk memanipulasi model itu.

Idealnya komputer kita mempunyai bahasa perintah yang alami, sehingga model pengguna dengan cepat dapat dioperasionalkan.



Umpan Balik

- Maksudnya diartikan sebagai kemampuan sebuah program yang membantu pengguna untuk mengoperasikan program itu sendiri. Umpan balik dapat berbentuk pesan penjelasan, pesan penerimaan perintah, indikasi adanya objek terpilih, dan penampilan karakter yang diketikkan lewat papan ketik. Beberapa umpan balik ini ditujukan untuk yang belum berpengalaman.
- Umpan balik dpt digunakan untuk memberi keyakinan bahwa program telah menerima perintah pengguna dan dpat memahami perintah tersebut.



Tampilan Informasi

Komponen ini digunakan untuk menunjukkan status informasi atau program ketika pengguna melakukan suatu tindakan.

Perancang harus menampilkan informasi seefektif mungkin sehingga mudah dipahami.



Merencanakan Suatu Proyek Aplikasi

Jika tidak direncanakan, maka :

1. Banyak kode sedikit fitur
2. Banyak bug
3. Banyak waktu akan dihabiskan
4. Kehilangan fitur

Untuk memperoleh hasil yang baik, maka :

1. Perencanaan
2. Ide
3. Pengumpulan kebutuhan
4. Pengujian
5. Evaluasi
6. Penambahan fitur



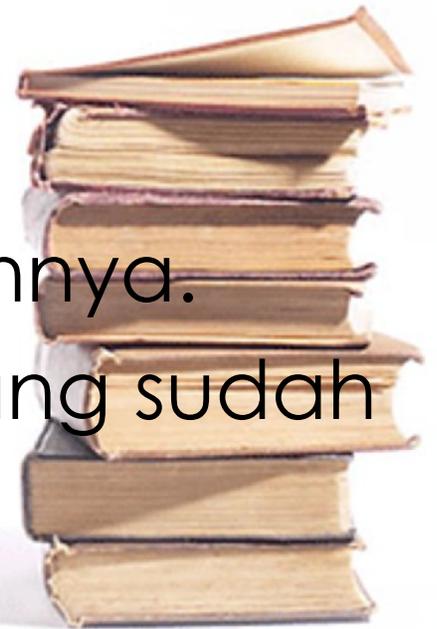
Sekarang kita akan melihat bagaimana merencanakan suatu proyek. Untuk keperluan ini dibutuhkan suatu ide,. Ada beberapa tipe ide guna memulai suatu proyek :

1. Memperbaiki atau membuat sesuatu lebih baik, meniru atau memperbaiki sesuatu yg telah dibuat. Dengan berbekal ide ini maka pembuatan proyek akan memerlukan lebih sedikit langkah dan lebih sedikit waktu.
2. Ide baru dimana pemrogram akan membuat sesuatu yang benar-benar baru. Untuk menggunakan ide ini tentu diperlukan waktu dan langkah yang tidak mudah.
3. Kebutuhan pasar. Ide juga bisa didapat dengan melihat kebutuhan pasar.



Setelah semua ide terchecklist, sebaiknya seorang pemrogram tidak langsung mewujudkannya, tetapi ada beberapa hal yang harus diperhatikan :

- a. Sudah berapa banyak persiapan check list yang telah dibuat??
- b. Berapa biaya yang dibutuhkan
- c. Berapa luas penggunaannya.
- d. Berapa lama proses pembuatannya.
- e. Apakah ide ini lebih baik dari yang sudah ada



Jika sudah diputuskan untuk membuat suatu proyek aplikasi, tugas berikutnya adalah membuat rincian persyaratan proyek yg mmberikan deskripsi rincian hasil akhir :



Jika persyaratan sudah dipetakan secara detail, baik fungsi input maupun outputnya, maka perlu ditinjau kembali apakah masih perlu diubah dan diperbaiki. Bila ternyata sudah baik, maka tiba saatnya kita memilih bahasa pemrograman yang tepat untuk membuat aplikasi tersebut.

Bahasa pemrograman yg digunakan sangat menentukan hasil yang akan diperoleh.



- Pengujian terhadap suatu aplikasi yang telah dibuat dilakukan untuk menemukan berbagai kesalahan yang mungkin terjadi sewaktu aplikasi tersebut dijalankan. kesalahan2 itu harus dibuatkan dokumentasinya.



Apa beda preview testing
dan beta testing??

