

PROTOTYPING

6.1 Tujuan Instruksional Umum

Mahasiswa dapat menguasai konsep-konsep interaksi manusia dan komputer dengan baik, sehingga dapat mengimplementasikannya dalam mendesain *software* sesuai dengan prinsip-prinsip *User Centered Design*.

6.2 Tujuan Instruksional Khusus

Mahasiswa mampu membedakan bentuk prototyping sketsa, *storyboard*, dan *scenario* prototyping tools yang dapat digunakan.

6.3 Pengertian Prototyping

Prototipe merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap sehingga segera dapat dievaluasi oleh pemakai. Prototipe mewakili model produk yang akan dibangun atau mensimulasikan struktur, fungsionalitas dan operasi sistem. Dalam pembuatan prototipe kita dapat menerapkan UCD (*User Centered Design*) yang cocok untuk user awam IT (*Technology Information*).

Karakteristik UCD (*User-Centered Design*) yaitu :

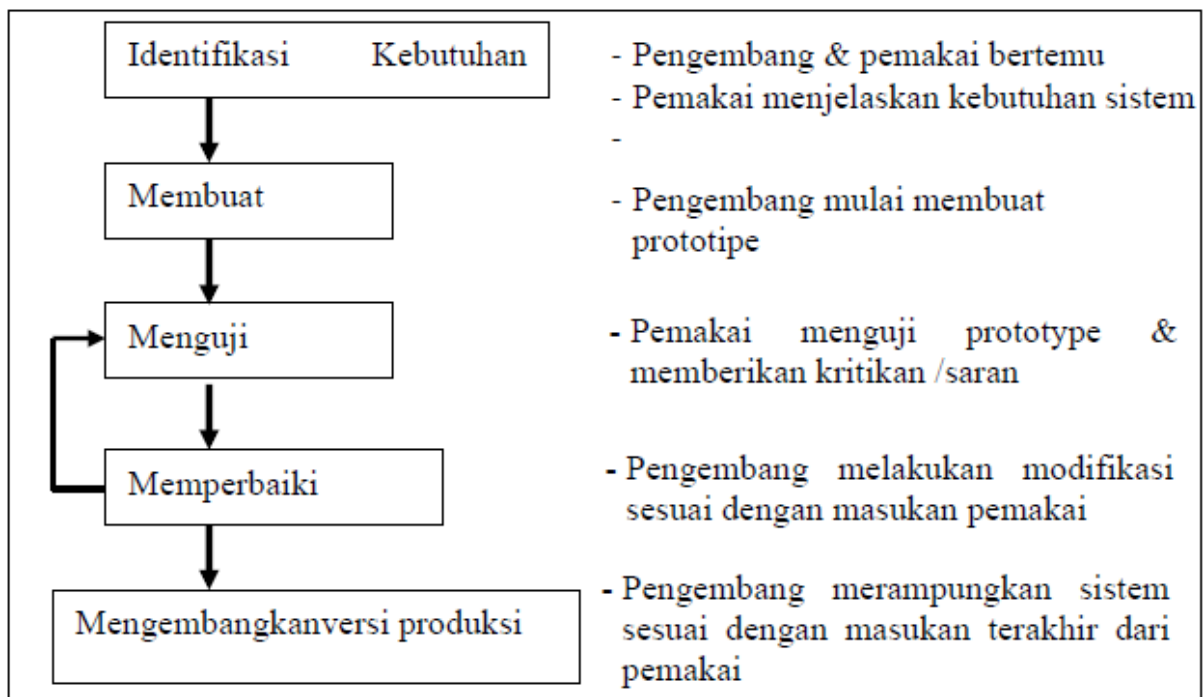
- Memahami user dan kebutuhannya.
- Fokus pada user pada tahap awal desain dan mengevaluasi hasil desain.
- Mengidentifikasi, membuat dokumentasi dan menyetujui kegunaan dan tujuan pengalaman user.
- Perulangan hampir dapat dipastikan. Para perancang tidak pernah berhasil hanya dalam satu kali proses.

Tujuan prototipe yaitu :

- Evaluasi dan feedback pada rancangan interaktif.
- Stakeholder (dalam hal ini user) dapat melihat, menyentuh, berinteraksi dengan prototype.

- Anggota tim dapat berkomunikasi secara efektif.
- Para perancang dapat mengeluarkan ide-idenya.
- Memunculkan ide-ide secara visual dan mengembangkannya.
- Dapat menjawab pertanyaan membantu pemilihan di antara alternatif-alternatif.
- Mendapatkan informasi dan pemahaman tentang pengalaman pengguna.
- Estimasi dan validasi tentang kesulitan perancangan, keputusan dan keuntungan biaya.
- Investigasi, eksplorasi dan perbandingan solusi perancangan yang berbeda.

Tahapan pembuatan prototype seperti pada gambar berikut :



Gambar 6.1 Tahapan Prototyping

Keunggulan menggunakan prototipe yaitu :

1. Adanya komunikasi yang baik antara pengembang dan pelanggan
2. Pengembang dapat bekerja lebih baik dalam menentukan kebutuhan pelanggan
3. Pelanggan berperan aktif dalam pengembangan sistem
4. Lebih menghemat waktu dalam pengembangan sistem
5. Penerapan menjadi lebih mudah karena pemakai mengetahui apa yang diharapkannya.

6. Mengurangi biaya pengembangan dan pemeliharaan
7. Dapat bereksperimen dengan perancangan alternatif
8. Meningkatkan kualitas, kecepatan spesifikasi dan perancangan sistem
9. Dapat dipadukan dengan metode terstruktur dan CASE tools
10. Dapat digunakan pada berbagai level, detail termasuk konsep, kebutuhan, spesifikasi dan perancangan.

spesifikasi dan perancangan.

Kelemahan prototipe yaitu :

1. Pelanggan kadang tidak melihat atau menyadari bahwa perangkat lunak yang ada belum mencantumkan kualitas perangkat lunak secara keseluruhan dan juga belum memikirkan kemampuan pemeliharaan untuk jangka waktu lama
2. Pengembang biasanya ingin cepat menyelesaikan proyek. Sehingga menggunakan algoritma dan bahasa pemrograman yang sederhana untuk membuat prototyping lebih cepat selesai tanpa memikirkan lebih lanjut bahwa program tersebut hanya merupakan cetak biru sistem .
3. Hubungan pelanggan dengan komputer yang disediakan mungkin tidak mencerminkan teknik perancangan yang baik

6.4 Dimensi Prototyping

1. Penyajian

- Bagaimana desain dilukiskan atau diwakili?
- Dapat berupa uraian tekstual atau dapat visual dan diagram.

2. Lingkup

Apakah hanya interface atau apakah mencakup komponen komputasi?

3. Executability (Dapat dijalankan)

- Dapatkah prototype tersebut dijalankan?
- Jika dikodekan, apakah akan ada periode saat prototype tidak dapat dijalankan?

4. Maturation

- Apakah tahapan-tahapan produk ini mengikuti?
- Revolusioner: mengganti yang lama. Jika dirasa sistem yang lama sudah

- tidak dapat mengikuti perkembangan teknologi.
- Evolusioner : terus melakukan perubahan pada perancangan yang sebelumnya.

6.5 Metode Prototyping

Metode pembuatan prototipe dibedakan menjadi 2 yaitu :

1. Metode *Non-Computer*

Metode ini biasanya dikerjakan lebih awal dalam proses pembuatan.

Jenis metode *non-computer* yaitu :

- Sketsa, Mock-Ups
 - - interface (antarmuka) dideskripsikan menggunakan kertas
 - - Baik untuk mengungkapkan pendapat.
 - - Difokuskan pada orang dengan desain tingkat tinggi.
 - - Tidak terlalu baik untuk menggambarkan alur dan rinciannya.
 - - Murah dan cepat umpan balik sangat menolong.
- Storyboarding
 - Storyboard adalah lembaran kertas yang berisi contoh tampilan antarmuka pengguna, dengan setiap antarmuka layar diperlihatkan pada lembar kertas yang berbeda.
 - Suatu storyboard akan terdiri dari tampilan layar yang menggambarkan fitur sistem seperti menu, kotak dialog dan window.
 - Pensil dan simulasi catatan atau walkthrough dari kemampuan dan tampilan sistem.
 - Menggunakan urutan diagram/gambar.
 - Menunjukkan kunci snap shots.
 - Setiap halaman terdapat keterangan sehingga pengguna dapat menjelajah ke seluruh aplikasi.
 - Cepat dan murah

Kapan storyboard ini digunakan?

- Storyboard dapat diperlihatkan pada teman satu tim atau pengguna potensial, sehingga orang lain dapat melihat visualisasi dan komposisi dari antarmuka yang diharapkan, serta memberikan kritik.

- Storyboard dapat digunakan pada awal siklus perancangan yang dapat mendukung eksplorasi kemungkinan perancangan dan verifikasi awal dari kebutuhan pengguna.

No. sheet: 5

PENGISIAN DATA PRIBADI

Nama:

Alamat:

Kota:

Tgl. Lahir:

Kelamin: ☒ Laki-laki ☐ Perempuan

Pendidikan: ☐ SD ☐ Sarjana S1
☐ SLTP ☐ Master
☐ SLTA ☐ Doktor

NAVIGASI

- Klik 'Selesai', ganti tampilan dengan No. 1.
- Klik 'Simpan', aktifkan penyimpanan berkas dan kosongkan borang.
- Klik 'Batal', kosongkan borang.

Keterangan:

- * Judul jendela pakai Arial, 12 point.
- * Teks pakai Arial, 10 point, hitam.
- * Medan data, Arial, 10 point, miring.
- * Teks 'push button' Arial, 12 point.

Gambar 6.2 Storyboarding

- Skenario
 - Hipotesis atau imajinasi penggunaan.
 - Biasanya menyertakan beberapa orang, peristiwa, lingkungan dan situasi.
 - Menyediakan konteks operasi.
 - Terkadang dalam format naratif, tetapi dapat berupa sketsa atau bahkan video.
 - Mengurangi level fungsionalitas dan jumlah fitur
 - Sangat mudah untuk dirancang dan diimplementasikan
 - Berukuran kecil, dapat diubah dan ditest berulang-ulang.
 - Dapat dikembangkan menggunakan satu atau beberapa skenario tugas
- Utilitas skenario
 - Menjanjikan dan menarik
 - Mengijinkan perancang untuk melihat masalah dari pandangan orang lain
 - Memudahkan umpan balik dan pendapat
 - Dapat sangat kreatif dan modern

- Tutorial dan manual
 - Mungkin menuliskannya lebih berguna daripada disimpan dalam kepala
 - Memaksa perancang untuk membuat keputusan dengan tegas
 - Menulis/meletakkannya di atas kertas lebih berharga.
 -

2. Metode *Computer-Based*

Metode ini biasanya dikerjakan kemudian. Sifat metode *computer-based* yaitu:

- Menirukan lebih banyak kemampuan sistem.
- Pada umumnya hanya baru beberapa aspek atau fitur
- Dapat berpusat pada lebih banyak detail
- Para pemakai lebih segan untuk menyarankan perubahan sekali ketika mereka melihat prototype yang lebih realistis.
- Prototype cepat digambarkan sebagai suatu metode berbasis komputer yang dapat membantu untuk mengurangi iterasi siklus pengembangan
- Prototype interaktif dikembangkan agar dapat dengan cepat diganti atau diubah sejalan dengan umpan balik perancangan.
- Umpan balik ini dapat diperoleh dari kolega atau dari pengalaman pengguna selama bekerja dengan prototype untuk menyelesaikan tugasnya.

6.6 Terminologi Prototipe

1. Prototipe horizontal

- Mencakup seluruh antarmuka pengguna namun tanpa fungsi pokok, berupa simulasi dan belum dapat digunakan untuk melakukan pekerjaan yang sesungguhnya.
- Misal, pengguna dapat mengeksekusi seluruh navigasi dan perintah pencarian tapi tanpa memanggil informasi secara nyata.
- Mengurangi level fungsionalitas, tetapi semua fitur ada.

2. Prototipe vertikal

- Lebih sedikit aspek atau fitur dari interface yang disimulasikan, tetapi dilaksanakan dengan rincian yang sangat baik.
- Misal: dalam sistem informasi penerbangan, pengguna dapat mengakses suatu basisdata dengan data real dari penyedia informasi, tetapi tidak untuk

- keseluruhan data.
- Mempunyai performance lebih rendah daripada sistem akhir.
- Tidak dalam jaringan

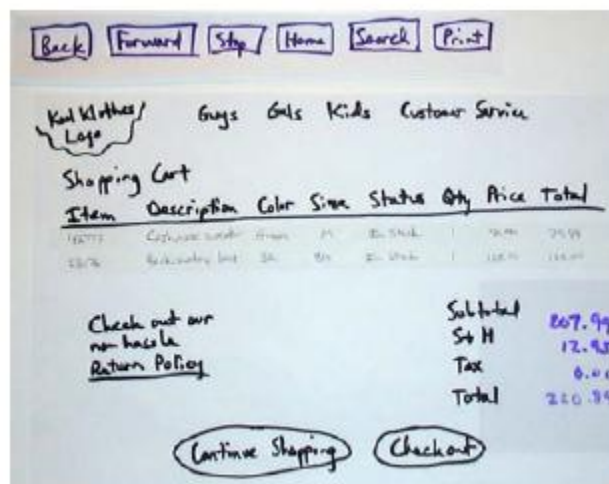
3. *Early prototyping* (prototipe cepat)

4. *Late prototyping* (prototipe lambat)

5. *Low-fidelity prototyping* (prototipe dengan tingkat ketepatan yang rendah)

Fidelity mengacu pada tingkat kerincian dengan produk akhir. *Low fidelity* mempunyai karakteristik antara lain :

- Gambaran cepat dari sistem akhir
- Mempunyai fungsi atau interaksi yang terbatas
- Lebih menggambarkan konsep, perancangan, alternatif dan layout layar dibanding model interaksi pengguna dengan sistem
- Mendemostrasikan secara umum 'feel and look ' dari antarmuka pengguna.
- Tidak untuk memperlihatkan secara rinci bagaimana operasi sistem aplikasi.
- Digunakan pada awal siklus perancangan
- Memperlihatkan konsep pendekatan secara umum tanpa harus membuang banyak tenaga, biaya dan waktu.



Gambar 6.3 Low-fidelity prototyping

Contoh low-fidelity prototyping:

- Storyboard
 - Digunakan di awal desain.

- Biasanya digunakan dengan skenario, lebih terinci, dan dapat diputar ulang.
- Kumpulan dari sketsa/frame individual.
- menyajikan urutan inti cerita.
- menunjukkan bagaimana kemungkinan user dapat mengalami peningkatan melalui setiap aktifitas.
-
- Sketsa
 - Sketsa sangat penting untuk low-fidelity prototyping.
 - Jangan takut dengan kemampuan menggambar.
 - Menyajikan “tampilan” cepat dari interface, konsep desain, dll
- Wizard of Oz
 - Digunakan tampilan maket dan berinteraksi dengan pemakai
 - Baik untuk mensimulasikan sistem yang sulit dibuat

6. *Mid-fidelity prototyping* (prototipe dengan tingkat ketepatan sedang)

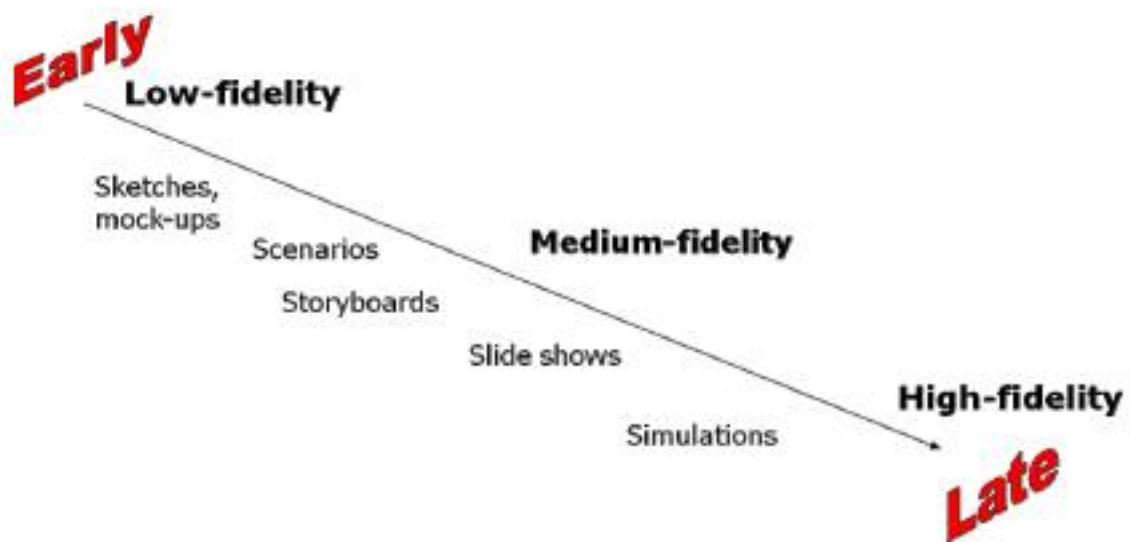
- Form skematik.
- Navigasi dan fungsi yang disimulasikan biasanya berbasis pada apa yang tampil pada layar atau simulasi layar.
- Contoh tools yang digunakan: powerpoint, illustrator, dll.

7. *High-fidelity prototyping* (prototipe dengan tingkat ketepatan yang tinggi) Karakteristik dari prototipe ini yaitu :

- Mempunyai interaksi penuh
- Pengguna dapat memasukkan data ke dalam medan masukan, menanggapi pesan, memilih ikon untuk membuka window, berinteraksi dengan UI
- Mewakili fungsi-fungsi inti dari antarmuka pengguna produk
- Dapat mensimulasikan sebagian besar fungsi sistem akhir.
- Umumnya dibuat dengan 4GLs seperti Smalltalk atau bahasa pemrograman berbasis visual seperti Visual basic.
- Trade off kecepatan dengan ketelitian
- Tidak secepat dan semudah membuat prototipe low-fidelity
- Mewakili antarmuka pengguna yang akan diimplementasikan dalam produk akhir
- Mempunyai penampilan yang sangat mirip dengan produk aktual



Gambar 6.4 High-fidelity prototyping



Gambar 6.5 Pemodelan terminologi prototype

6.7 Prototyping Tools

1. Draw/Paint Program

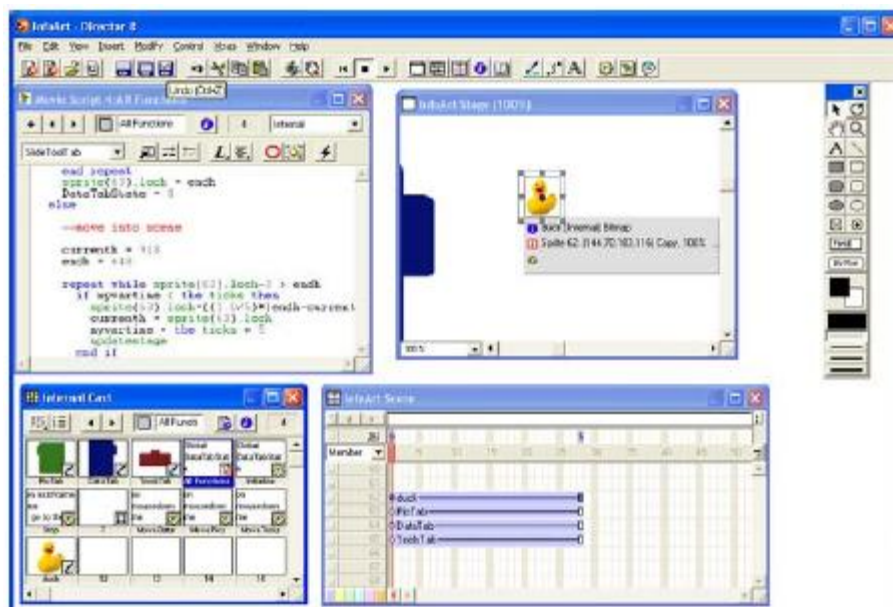
- Menggambar setiap layar, baik untuk dilihat
- Prototipe horizontal, tipis
- Contoh : Adobe Photoshop, Corel Draw



Gambar 6.6 Draw/Paint program

2. Scripted Simulation/Slide Show

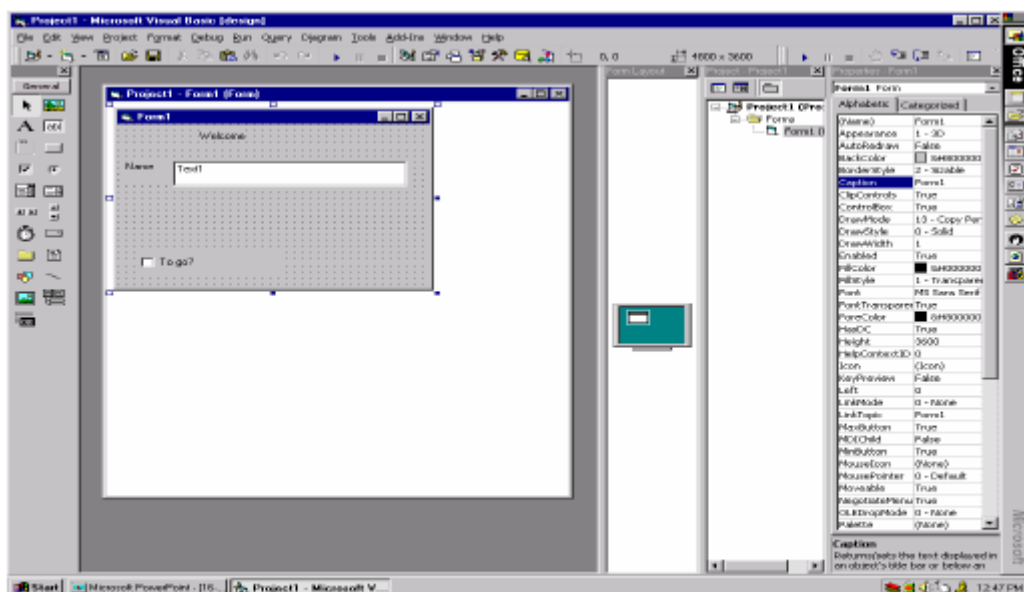
- Meletakkan tampilan seperti storyboard dengan animasi perubahandiantaranya.
- Dapat memberikan user catatan yang sangat spesifik
- Contoh : Powerpoint, Hypercard, Macromedia Director, HTML



Gambar 6.7 Scripted Simulation

3. Interface Builder

- Tools untuk menampilkan jendela, kendali, dan lain-lain dari interface.
- Mudah dikembangkan dan memodifikasi layar.
- Mendukung jenis interface yang dikembangkan.
- Mendukung berbagai macam device Input/Output.
- Mudah untuk memodifikasi dan menghubungkan layar.
- Mengijinkan memanggil prosedur eksternal dan program.
- Mengijinkan mengimpor teks, grafik, media lain.
- Mudah untuk dipelajari dan digunakan.
- Dukungan yang baik dari vendor.
- Contoh: Visual Basic, Delphi.



Gambar 6.8 Interface Builder

6.8 Bahan Diskusi

Menurut Anda, Apakah manfaat menggunakan prototipe sistem?Jelaskan!

6.9 Rangkuman

- Prototipe merupakan suatu metode dalam pengembangan sistem yang menggunakan pendekatan untuk membuat sesuatu program dengan cepat dan bertahap

- Tahapan prototipe yaitu identifikasi kebutuhan pemakai, membuat prototipe, menguji prototipe, memperbaiki prototipe, mengembangkan versi produksi.
- Dimensi prototipe terdiri dari penyajian, lingkup, executability dan maturation.
- Metode prototipe dibedakan menjadi metode *non-computer* dan metode *computer-based*.

6.10 Latihan Soal-soal

1. Mengapa kita membutuhkan prototipe sistem?
2. Apakah sebenarnya tujuan dari prototipe sistem?