6.1 Defenisi & Tujuan Prototyping

**Prototype** adalah model pertama dari produk yang digunakan untuk men-testing konsep atau gambaran dari ide kita. Prototyping telah digunakan oleh banyak industri. Sebelum memulai membangun sebuah bangunan, arsitek harus menggambarkan blueprint dari bangaunan dan membuat model dari bangunan. Perusahaan pesawat terbang juga harus membuat sebuah prototype dari design pesawat sebelum mulai membuatnya. Perusahaan yang bergerak di bidang software, juga membuat prototype software untuk mengexplore ide sebelum memulai pengembangan aplikasi.

Dalam kontek pengembangan aplikasi, sebuah prototype bisa menjadi contoh awal dari aplikasi dan hal ini menentukan mana fitur yang tidak akan digunakan sehingga muncul gambaran dasar dari tampilan aplikasi.

**Defenisi Prototyping antara lain adalah :**

* **Prototyping** adalah proses merancang sebuah prototype dimana prototype / pawarupa sendiri adalah sebuah model dari sebuah model produk yang mungkin belum memiliki semua fitur produk sesungguhnya namun sudah memiliki fitur – fitur utama dari produk sesungguhnya dan biasa digunakan untuk keperluan testing/uji coba untuk bahan uji coba sebelum berlanjut ke fase pembuatan produk sesungguhnya.
* **Prototyping** merupakan salah satu metode pengembangan perangkat lunak yang banyak digunakan. Dengan metode prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem.
* **Prototiping** adalah bagian dari produk yang mengekspresikan logika maupun fisik antarmuka eksternal yang ditampilkan. Konsumen potensial menggunakan prototipe dan menyediakan masukan untuk tim pengembang sebelum pengembangan skal besar dimulai.
* **Prototyping** model sendiri mempunyai tujuan yaitu mengembangkan model awal software menjadi sebuah sistem yang final.

**Tujuan pembuatan prototyping adalah : ....**

* Prototyping model sendiri mempunyai tujuan yaitu mengembangkan model awal software menjadi sebuah sistem yang final.
* Mengembangkan model menjadi sistem final. Artinya sistem akan dikembangkan lebih cepat daripada metode tradisional dan biayanya menjadi lebih rendah dan ekonomis.
* 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.
* Investigas, explorasi dan pembandingan solusi perancangan yang berbeda.

**Selain itu, pendekatan prototyping dapat dipisahkan oleh tujuan aplikasi.**

* **Protoyping Eksploratif** : Profil persyaratan suatu perangkat lunak diperjelas selangkah demi selangkah dengan membuat prototipe secara iteratif dan cepat dalam lingkungan pengujian. Fungsionalitas kemudian disempurnakan untuk menilai apakah perangkat lunak menyelesaikan masalah yang diasumsikan dan mencakup kebutuhan pengguna. Dalam konteks ini, demonstran, pembuatan prototipe cepat dan kertas juga disebutkan. Demonstran digunakan dalam akuisisi dan fase awal proyek untuk mengklarifikasi dan mengomunikasikan persyaratan dan masalah abstrak selama pengembangan.
* **Protoyping Eksperimental**: Sebuah desain dibuat dengan fungsi dasar dan diperiksa sehubungan dengan kelayakannya. Temuan percobaan atau tes ini dimasukkan ke dalam produk yang sebenarnya. Sampel laboratorium, prototipe yang dibuang sering disebut di sini. Pola-pola ini harus membantu menjawab pertanyaan teknis dan untuk dapat memeriksa kelayakan proyek seperti itu.
* **Protoyping Evolusi** Perangkat lunak ini dibuat secara berturut-turut. Pada setiap tahap pengembangan, putaran umpan balik dengan pengguna, pengembang dan klien memastikan bahwa produk akhir memenuhi profil persyaratan. Biasanya, versi perangkat lunak selalu dapat dieksekusi. Ini juga disebut sebagai sistem percontohan. Mereka sudah mengandung sejumlah besar fungsi yang dimaksudkan dan dapat diperiksa dengan cermat oleh pengguna.

6.2 Tahapan Pembuatan Prototyping & Kelebihan/Kekurangan Prototyping

**Tahapan dalam pembuatan Prototyping yaitu :**

1. **Pengumpulan kebutuhan.**Pelanggan dan pengembang bersama-sama mendefinisikan format dan kebutuhan kesseluruhan perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. **Membangun prototyping.**Membangun prototyping dengan membuat perancangan sementara yang berpusat pada penyajian kepada pelanggan (misalnya dengan membuat input dan contoh outputnya).
3. **Evaluasi protoptyping.**Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginan pelanggan. Jika sudah sesuai maka langkah keempat akan diambil. Jika tidak, maka prototyping diperbaiki dengan mengulang langkah 1, 2 , dan 3.
4. **Mengkodekan system.**Dalam tahap ini prototyping yang sudah disepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. **Menguji system.**Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.
6. **Evaluasi Sistem.**Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan . Jika sudah, maka langkah ketujuh dilakukan, jika belum maka mengulangi langkah 4 dan 5.
7. **Menggunakan system.**Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan.

**Kelebihan prototyping adalah :**

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.

**Kekurangan prototyping adalah :**

1. Pelanggan tidak melihat bahwa perangkat lunak belum mencerminkan kualitas perangkat lunak secara keseluruhan dan belum memikirkan peneliharaan dalam jangka waktu yang lama.
2. Pengembang biasanya ingin cepat menyelesaikan proyek sehingga menggunakan algoritma dan bahasa pemrograman sederhana.
3. Hubungan pelanggan dengan komputer mungkin tidak menggambarkan teknik perancangan yang baik.

6.3 Jenis Prototyping

Secara umum, metode prototyping dapat dibedakan berdasarkan apakah mereka vertikal atau horizontal.

* + Prototipe horisontal berfokus pada area spesifik perangkat lunak aplikasi – misalnya, antarmuka pengguna. Tidak ada referensi ke fungsi teknis dari keseluruhan sistem dan implementasinya. Tujuannya adalah untuk membawa pengguna atau klien berhubungan dengan GUI dan menerima umpan balik awal.
  + Prototyping vertikal mengambil area khusus dari perangkat lunak dan menunjukkan interaksi dengan komponen lain dari sistem. Antarmuka pengguna sudah dipetakan di sini bersama dengan manajemen data dan bagian lain dari sistem. Tujuannya adalah untuk menjelaskan fungsionalitas yang kompleks dan untuk memeriksa bagian-bagian perangkat lunak oleh pengguna.

**Prototype Horisontal**

* + Sangat luas, mengerjakan sebagian besar interface, tetapi tidak mendalam
  + 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 real.
  + Mengurangi level fungsionalitas, tetapi semua fitur ada.

**Prototype Vertikal**

* + Lebih sedikit aspek atau fitur dari interface yang disimulasikan, tetapi dilaksanakan dengan rincian yang sangat baik
  + Mengandung fungsi yang detail tapi hanya untuk beberapa fitur terpilih, dan tidak pada keseluruhan sistem
  + Misalnya dalam sistem informasi penerbangan, pengguna dapat mengakses suatu basisdata dengan data real dari penyedia informasi, tetapi tidak untuk keseluruhan data
  + Mempunyai performans lebih rendah dibanding sistem final.
  + Tidak dalam jaringan

**Early Prototyping (prototipe cepat)  
Late Prototyping (prototipe lambat)  
Low-fidelity Prototyping (prototype dengan tingkat ketepatan yang rendah)**.  
***Low - fidelity prototype Characteristics :***

* + Gambaran cepat dari sistem final
  + Mempunyai fungsi atau interaksi yang terbatas
  + Lebih menggambarkan konsep , perancangan, alternativ, dan layout layar dibanding model interaksi pengguna dengan sistem.
  + Mendemonstrasikan 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.

**Mid-fidelity prototyping (prototype dengan tingkat ketepatan sedang)**

*Contoh tools yang digunakan: powerpoint, illustrator, dll.*  
  
**High-fidelity prototyping (prototype dengan tingkat ketepatan yang tinggi)**

*High - fidelity prototype Characteristics :*

* + Mempunyai interaksi penuh
  + Pengguna dapat memasukkan data kedalam medan masukan, menanggapi pesan, memilih icon untuk membuka window,berinteraksi dengan UI
  + Mewakili fungsi-fungsi inti dari antarmuka pengguna produk
  + Umumnya dibuat dengan 4GLs seperti Smalltalk atau bahasa pemrograman berbasis visual
  + Dapat mensimulasikan sebagaian besar fungsi dari sistem akhir
  + Trade off kecepatan dengan ketelian
  + 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

**Low VS high-fidelity prototypes**

* + Fidelity mengacu pada tingkat kerincian dengan produk akhir
  + Low-fidelity prototype (tidak terlalu rinci)
  + High fidelity prototype (seperti produk akhir)

**Scenario-based prototyping**

* + Mengurangi level fungsionalitas dan jumlah fitur
  + Sangat mudah untuk dirancang dan diimplementasikan
  + Hanya mampu untuk mensimulasikan UI sepanjang pengujian pengguna mengikuti rencana pengujian sebelumnya.
  + Berukuran kecil, dapat diubah dan ditest berulang-ulang.
  + Dapat dikembangkan menggunakan satu atau beberapa skenario tugas

**Major Prototype Types**  
*\* Low-Fidelity Prototypes*

* + Prototipe dengan menggunakan sketsa (contoh/tiruan) dan kertas.
  + Fiktif, dengan menggunakan komponen antarmuka yang dibuat terlebih dulu pada kertas

**\****Medium-Fidelity Prototypes*

* + Membuat gambar pada komputer Storyboards
  + Membangun antarmuka pengguna dalam mode test
  + Membuat Slides show dan simulasi

**\****High-Fidelity Prototyping Techniques*

* + Berupa prototipe yang berbasis perangkat lunak dengan tool RAD, GUI Builders dan Editor Script

**Paper prototyping**

* + Teknik ini memanfaatkan materi dan peralatan sederhana (kertas dan pensil) untuk membuat antarmuka sistem pada kertas.
  + Model prototipe ini merupakan salah satu cara yang efektif dan berguna untuk mengevaluasi dan meng-iterasi perancangan sebelum tim memutuskan untuk mengimplementasikannya
  + Elemen antarmuka seperti menu, window, dialog dan icon dapat dibuat sketsanya pada kertas.

**Jenis-jenis Prototyping**

1. Feasibility prototyping
2. Requirement prototyping
3. Desain Prototyping
4. Implementation prototyping

6.4 Bentuk & Fase-Fase Metode Prototyping

Prototyping merupakan salah satu metode pengembangan perangat lunak yang banyak digunakan. Dengan metode prototyping ini pengembang dan pelanggan dapat saling berinteraksi selama proses pembuatan sistem.Prototyping dapat diartikan sebagai proses yang digunakan untuk membantu pengembang perangkat lunak dalam membentuk model dari perangkat lunak yang harus dibuat.

**Berdasarkan karakteristiknya prototipe sebuah sistem dapat berupa :**

* **Low fidelity prototype** tidak terlalu rinci menggambarkan sistem. Karakteristik dari low fidelity prototype adalah mempunyai fungsi atau interaksi yang terbatas, lebih menggambarkan kosep perancangan dan layout dibandingkan dengan model interaksi, tidak memperlihatkan secara rinci operasional sistem, mendemostrasikan secara umum feel and look dari antarmuka pengguna dan hanya menggambarkan konsep pendekatan secara umum (Walker et al, 2003).
* **High fidelity protoype** lebih rinci menggambarkan sistem. Prototipe ini mempunyai interaksi penuh dengan pengguna dimana pengguna dapat memasukkan data dan berinteraksi dengan dengan sistem, mewakili fungsi-fungsi inti sehingga dapat mensimulasikan sebagian besar fungsi dari sistem akhir dan mempunyai penampilan yang sangat mirip dengan produk sebenarnya (Walker et al, 2003).

**Model dapat berupa tiga bentuk yaitu :**

1. Bentuk prototype di atas kertas/model berbasis komputer yang menggambarkan interaksi manusia yang mungkin terjadi.
2. Working prototype, yang mengimplementasikan sebagian dari fungsi yang ditawarkan perangkat lunak.
3. Program jadi yang melakukan sebagian atau seluruh fungsi yang akan dilakukan, tapi masih ada fitur yang masih dikembangkan.

**Fase fase dalam *Prototyping* adalah sebagai berikut :**

1. Pengumpulan kebutuhan. Pelanggan dan pengembang bersama-sama mendefinisikan format seluruh perangkat lunak, mengidentifikasikan semua kebutuhan, dan garis besar sistem yang akan dibuat.
2. Membangun prototyping. Membangun prototyping dengan membuat perancangan sementara yang berfokus pada penyajian kepada pelanggan (misalnya dengan membuat input dan format output).
3. Evaluasi protoptyping. Evaluasi ini dilakukan oleh pelanggan apakah prototyping yang sudah dibangun sudah sesuai dengan keinginann pelanggan. Jika sudah sesuai maka langkah 4 akan diambil. Jika tidak prototyping direvisi dengan mengulang langkah 1, 2 , dan 3.
4. Mengkodekan sistem. Dalam tahap ini prototyping yang sudah di sepakati diterjemahkan ke dalam bahasa pemrograman yang sesuai.
5. Menguji sistem. Setelah sistem sudah menjadi suatu perangkat lunak yang siap pakai, harus dites dahulu sebelum digunakan. Pengujian ini dilakukan dengan White Box, Black Box, Basis Path, pengujian arsitektur dan lain-lain.
6. Evaluasi Sistem. Pelanggan mengevaluasi apakah sistem yang sudah jadi sudah sesuai dengan yang diharapkan. Jika ya, langkah 7 dilakukan; jika tidak, ulangi langkah 4 dan .
7. Menggunakan sistem. Perangkat lunak yang telah diuji dan diterima pelanggan siap untuk digunakan

6.5 Metode Pembuatan 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.
* 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.
* 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 :

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

6.6 Prototyping Tools

**1. Draw/Paint Program, contoh: Photoshop, Coreldraw**

* + Menggambar setiap layar, baik untuk dilihat.
  + Prototype horisontal, tipis.
  + Adobe Photoshop.

**2. Scripted Simulations/Slide Show,**

* + Contoh : Powerpoint, Hypercard,Macromedia Director, HTML.
  + Letakkan tampilan seperti storyboard dengan (animasi) perubahan diantaranya.
  + Dapat memberikan user catatan yang sangat spesifik.
  + Disebut chauffeured prototyping.
  + Macromedia Director.

**3. Interface Builders, contoh: Visual Basic, Delphi, UIMX.**

* + Tools untuk menampilkan jendela, kendali, dan lain-lain dari interface.

6.7 Prototyping Strategies

**Menurut strategi dan tujuan pembuatan, *prototyping* dapat dibedakan menjadi 2 yaitu :**

**Requirements Prototyping**

Pada *Requirements Prototyping, prototype* memiliki tujuan utama yaitu  untuk mencari kebutuhan dari pengguna / *user requirements*yang dibutuhkan di dalam perancangan dan pembuatan produk yang sesungguhnya.

Kelebihan mencari kebutuhan pengguna menggunakan metode prototyping adalah :

1. Prototype membuat pengguna mengalami menggunakan produk/aplikasi tersebut terlebih dahulu, sehingga hasil yang didapat akan lebih akurat karena berdasarkan fakta pengalaman pengguna ketika menggunakan aplikasi tersebut, bukan hanya pengalaman pengguna ketika membayangkan memiliki aplikasi tersebut.
2. Kegunaan dan Fleksibilitas, dengan menggunakan *prototype* sebagai alat untuk mencari kebutuhan pengguna, pengguna dapat memberikan masukan yang langsung tepat sasaran, karena pengguna sudah terlebih dahulu mendapatkan pengalaman menggunakannya sehingga nantinya produk sesungguhnya akan memiliki kegunaan yang lebih dan lebih fleksibel terhadap keinginan pengguna.
3. Memudahkan implementasi, dengan melihat *prototype*, perancang dapat berkomunikasi secara langsung dengan pihak yang akan membuat aplikasi tersebut sehingga proses implementasi dapat lebih jelas karena telah dikomunikasikan terlebih dahulu.

Pada *Requirements Prototyping*, setelah *user requirements* diketahui maka *prototype*akan dibuang dan perancang akan membuat produk tanpa meneruskan dari *prototype.*

**Evolutionary Prototyping**

*Evolutionary Prototyping* adalah metode berbeda dari *Requirements Prototyping*. Tujuan utama ketika menggunakan *Evolutionary Prototyping* adalah untuk membangun *prototype* yang kuat dengan cara yang terstruktur dan terus-menerus memperbaikinya, dengan cara mengembangkan terus menerus yang sebelumnya digunakan untuk menentukan kebutuhan user/*user requirements*sampai menjadi sistem akhir / produk sesungguhnya.

Dengan alasan untuk meminimalkan risiko, pengembang tidak mengimplementasikan fitur kurang dipahami. Sistem parsial dikirim ke situs pelanggan. Sebagai pengguna bekerja dengan sistem, mereka mendeteksi kesempatan untuk fitur-fitur baru dan memberikan permintaan fitur ini untuk pengembang. Pengembang kemudian mengambil permintaan tambahan ini bersama-sama dengan mereka sendiri dan menggunakan konfigurasi suara-praktik manajemen untuk mengubah perangkat lunak-persyaratan spesifikasi, update desain, recode dan tes ulang.