

# **MATERI PEMBEKALAN JUNIOR WEB PROGRAMMER**



## **HARI KE-1**

<b>J.620100.011.01</b>	<b>Menggunakan Instalasi Software Tools</b>
<b>J.620100.005.02</b>	<b>Mengimplementasikan user interface</b>



**JUNIOR WEB  
PROGRAMMER**

**J.620100.011.01**

**Melakukan Instalasi Software  
Tools**

# 3

## MELAKUKAN INSTALASI SOFTWARE TOOLS

### Objektif:

1. Memilih dan menyiapkan *tools* pemrograman yang sesuai dengan kebutuhan.
  2. Instalasi tool pemrograman
  3. Menerapkan hasil pemodelan ke dalam eksekusi *script* sederhana.
- 

### 1. Menyiapkan tools pemrograman

Berisi informasi mengenai software yang digunakan. Untuk menjalankan kode PHP, setidaknya butuh 3 jenis program, yakni:

1. Aplikasi Web Server
2. Aplikasi PHP
3. Aplikasi Web Browser.

Untuk aplikasi web server, terdapat beberapa pilihan. Saat ini web server yang sering digunakan adalah **Apache**. Aplikasi selanjutnya adalah PHP itu sendiri. PHP bekerja sebagai “penerjemah” di dalam web server. Akan tetapi, proses instalasi web server Apache dan PHP secara terpisah butuh waktu yang cukup lama dan juga perlu pengetahuan tentang konfigurasi *Apache*. Terdapat banyak aplikasi yang membundel Apache+PHP. Beberapa diantaranya adalah *XAMPP* dan *WAMP*. Pada tutorial pembelajaran ini akan menggunakan **XAMPP**. Aplikasi terakhir yang dibutuhkan adalah web browser.

## 2. Instalasi Tool Pemrograman

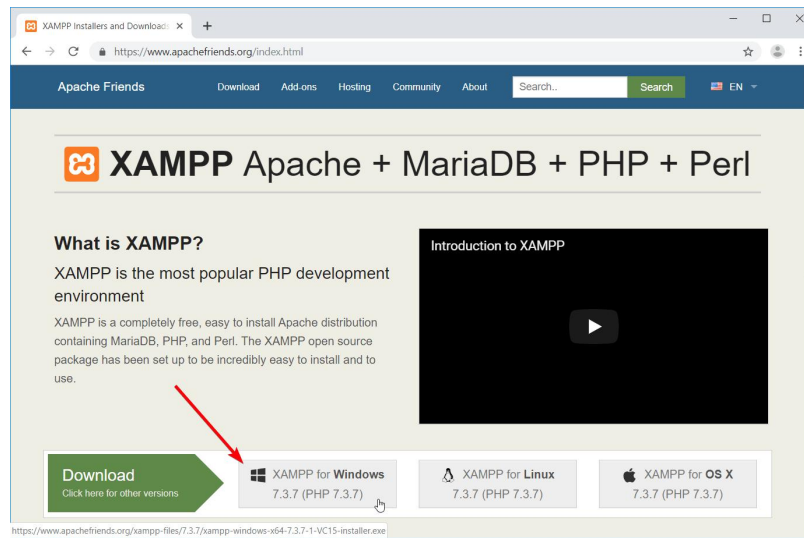
Instalasi (pengaturan) program komputer adalah tindakan membuat program siap untuk dijalankan. Instalasi mengacu pada konfigurasi tertentu dari perangkat lunak atau perangkat keras dengan tujuan agar dapat digunakan dengan komputer. Salinan lunak atau digital dari perangkat lunak (program) diperlukan untuk menginstalnya. Ada proses yang berbeda untuk menginstal perangkat lunak (program). Karena prosesnya berbeda-beda untuk setiap program dan setiap komputer, program (termasuk sistem operasi) sering kali disertai dengan penginstal, program khusus yang bertanggung jawab untuk melakukan apa pun yang diperlukan untuk instalasi. Penginstalan mungkin merupakan bagian dari proses penerapan perangkat lunak yang lebih besar.

Instalasi biasanya melibatkan kode (program) yang disalin/dihasilkan dari file instalasi ke file baru di komputer lokal untuk akses yang lebih mudah oleh sistem operasi, membuat direktori yang diperlukan, mendaftarkan variabel lingkungan, menyediakan program terpisah untuk un-instalasi dan lain-lain. Karena kode itu umumnya disalin/dibuat di banyak lokasi, penghapusan instalasi biasanya melibatkan lebih dari sekadar menghapus folder program. Misalnya, file registri dan kode sistem lainnya mungkin perlu dimodifikasi atau dihapus untuk penghapusan total.

### 2.1 Cara Instalasi XAMPP

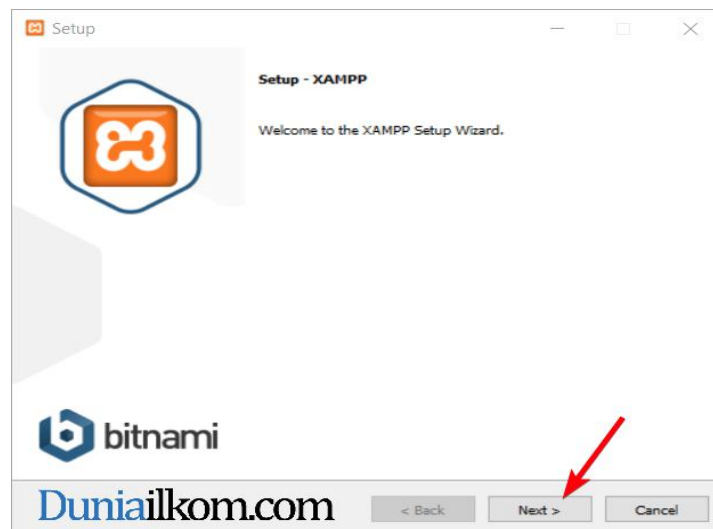
**XAMPP** adalah singkatan dari aplikasi dalam 'paketnya', yaitu: **X** (berarti *cross-platform*, maksudnya tersedia dalam berbagai sistem operasi), **A**pache Web Server, **M**ySQL / **M**ariaDB, **P**HP dan **P**erl. Dengan menginstall **XAMPP**, secara tidak langsung telah menginstal keempat aplikasi tersebut. Mulai dari XAMPP versi 5.6.14, database yang

dipakai adalah **MariaDB**, bukan lagi MySQL. MariaDB ini bisa disebut sebagai 'cloningan' dari MySQL. Hampir semua perintah yang biasa dilakukan dengan MySQL, juga bisa dipakai ke MariaDB tanpa perubahan apapun. Untuk memulai proses instalasi, silahkan download XAMPP dari [apachefriends.org](https://www.apachefriends.org).



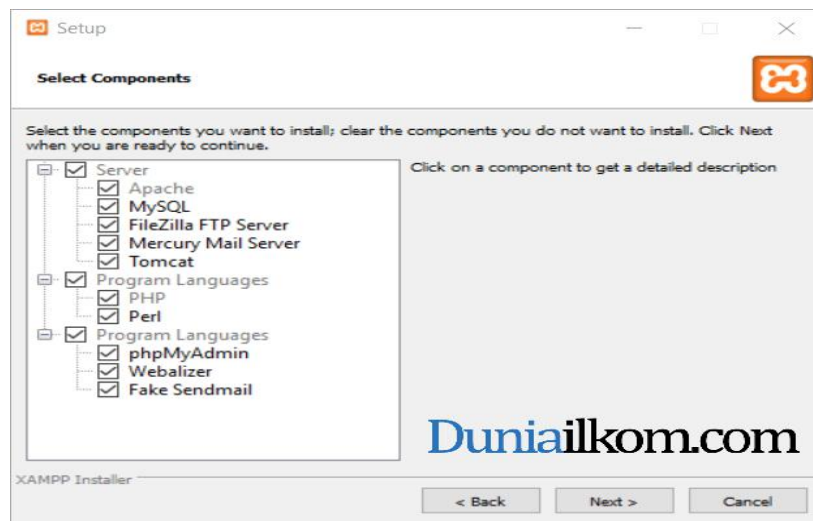
Gambar 1. Mengunduh Software XAMPP

Setelah aplikasi installer XAMPP berhasil diunduh, langkah selanjutnya adalah menginstall XAMPP. Double klik file **xampp-xxxxx-installer.exe** (sesuai dengan versi yang diunduh).



Gambar 2. Tampilan Awal Instalasi XAMPP

Tampilan berikutnya adalah "**Select Component**", pada layar ini anda bisa memilih aplikasi apa saja yang akan diinstall. Disarankan untuk membiarkan pilihan **default**, dalam hal ini semua aplikasi akan diinstal. Namun jika memiliki kendala akan keterbatasan ruang harddisk, bisa memilih aplikasi penting saja, yakni: **Apache, MySQL, PHP**, dan **phpmyadmin**. Setelah itu, lanjutkan dengan klik tombol **Next**.



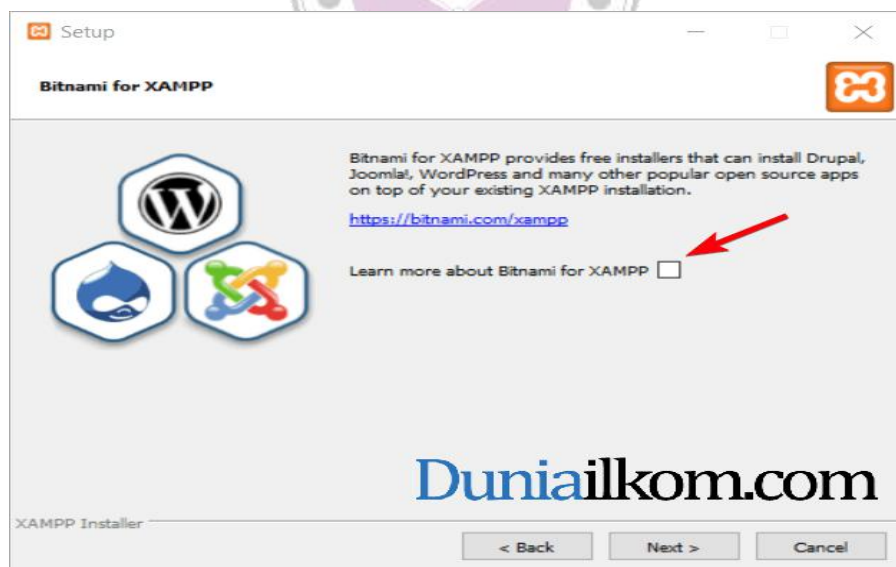
Gambar 3. Tampilan Select Components XAMPP

Pada tampilan berikutnya, "**Installation Folder**" dapat mengubah folder instalasi default XAMPP ke folder lain, dalam contoh ini akan membiarkan pilihan default, yakni di **C:\xampp**. Lalu klik **Next** untuk melanjutkan.



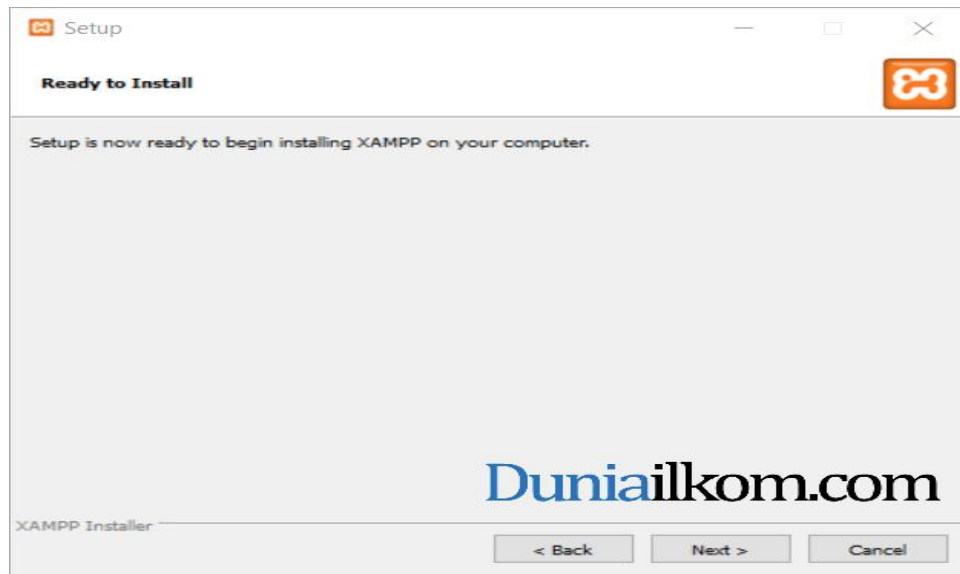
Gambar 4. Tampilan Installation Folder XAMPP

Tampilan layar "**Bitnami for XAMPP**" adalah iklan produk installer CMS dari Bitnami (sponsor XAMPP). Karena tidak memerlukannya, hapus centang pilihan "**Learn more about BitNami for XAMPP**" dan klik tombol **Next**.



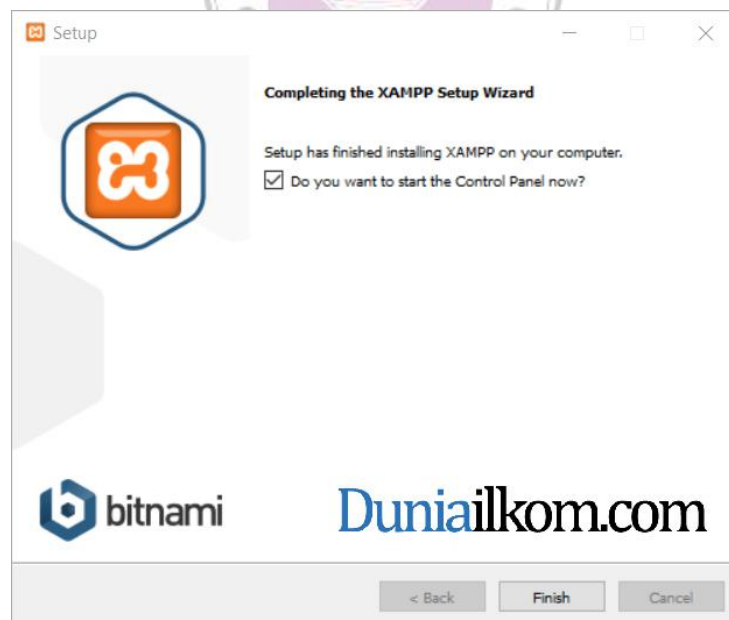
Gambar 5. Tampilan Iklan Bitnami XAMPP

Jika semuanya berjalan lancar, tampilan berikutnya adalah "**Ready to Install**", klik tombol **Next** untuk memulai proses instalasi.



Gambar 6. Tampilan Install XAMPP

Proses intallasi akan berjalan beberapa saat, hingga layar tampilan menjadi "**Completing the XAMPP**". Akhiri proses installasi XAMPP dengan klik tombol **Finish**.



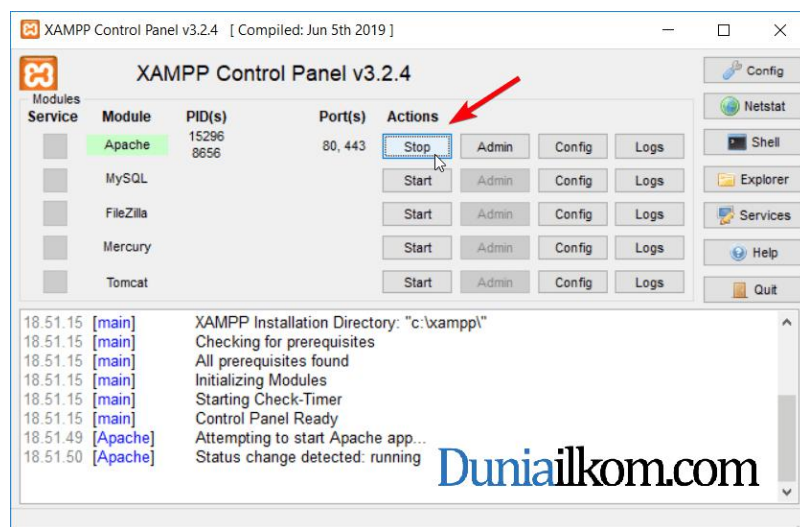
Gambar 7. Tampilan Selesai Install XAMPP



Setelah XAMPP berhasil diinstal, akan muncul layar "**XAMPP Control Panel**". Pada jendela inilah kita akan menjalankan seluruh aplikasi yang ada di XAMPP, yang diantaranya adalah **Apache Web Server** dan **MySQL**.

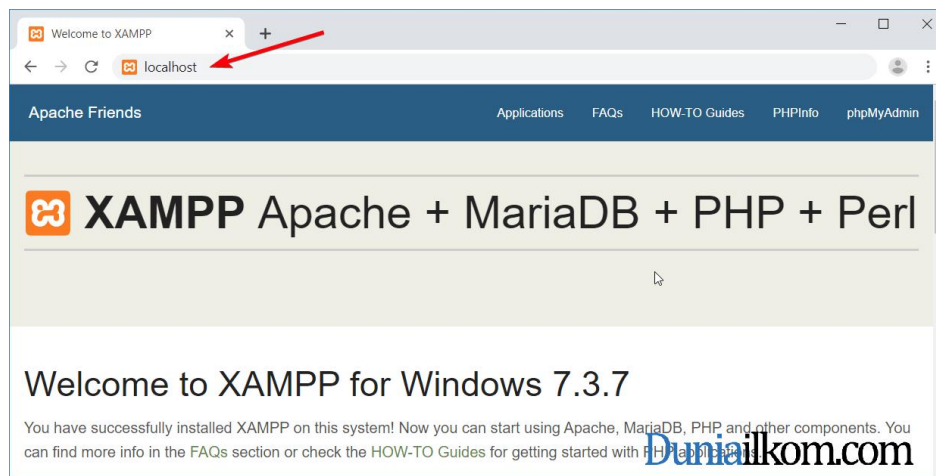
## 2.2 Menjalankan APACHE

Untuk menjalankan dari XAMPP, langsung saja klik tombol **Start** pada bagian **Action Apache**. Sesaat kemudian **Module Apache** akan berwarna hijau menandakan aplikasi Web server Apache sudah berjalan.



Gambar 8. Tampilan XAMPP Control Panel

Selanjutnya buka web browser seperti **Mozilla Firefox**. Pada kolom address ketikkan **localhost** dan tekan **Enter**. Jika anda melihat halaman **XAMPP**, berarti instalasi XAMPP telah berhasil dan Web Server Apache telah sukses berjalan.



Gambar 9. Tampilan Localhost

Alamat **localhost** adalah alamat khusus yang diterjemahkan oleh web browser untuk mengakses web server pada komputer itu sendiri dan diakses secara offline (local). Selain menggunakan alamat **localhost**, bisa juga menggunakan alamat IP **127.0.0.1**.

### 2.3 Folder htdocs

Agar dapat mengakses halaman PHP dari web browser, maka harus menempatkan file **PHP** dalam folder khusus yang merupakan folder **home** dari **web server**. Pada aplikasi **XAMPP**, folder tersebut adalah folder **htdocs** yang berada di **C:\xampp\htdocs**. Seluruh file PHP **harus** diletakkan dalam folder htdocs ini.

## 3 Membuat skrip sederhana

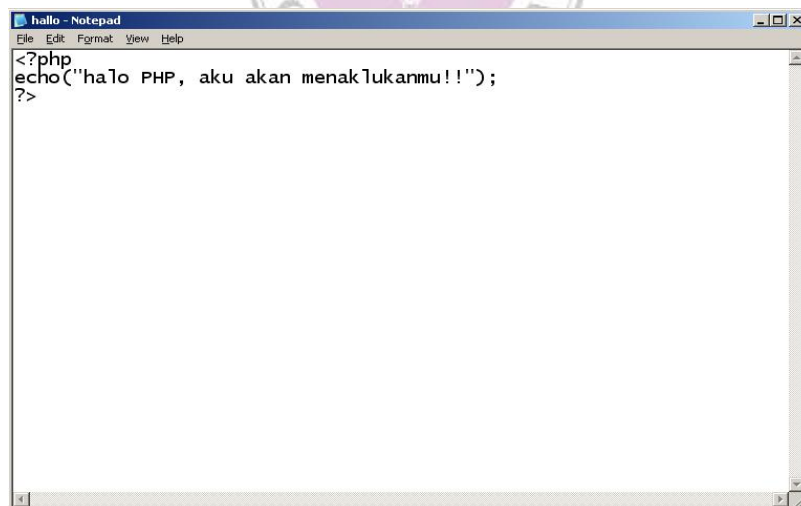
Bahasa skrip (*scripting language*) merupakan bahasa pemrograman yang menyediakan fasilitas penerjemahan serta kompilasi kode dalam satu rangkaian proses secara integratif sehingga memungkinkan kode dibuat dapat langsung dijalankan sebagai program secara dinamis. Skrip atau kode program yang akan dijalankan umumnya akan diterjemahkan terlebih dulu dari kode sumber ke dalam P-Code, sebagai tambahan,

beberapa bahasa skrip bahkan telah menyediakan kompiler JIT terintegrasi yang akan mengevaluasi serta menerjemahkan bagian-bagian kritis atas P-Code tersebut menjadi kode mesin sehingga menghasilkan eksekusi proses yang lebih cepat hingga secara relatif bisa menandingi program yang ditulis dengan menggunakan bahasa pemrograman kompilatif seperti C.

### 3.1 Membuat Skrip PHP

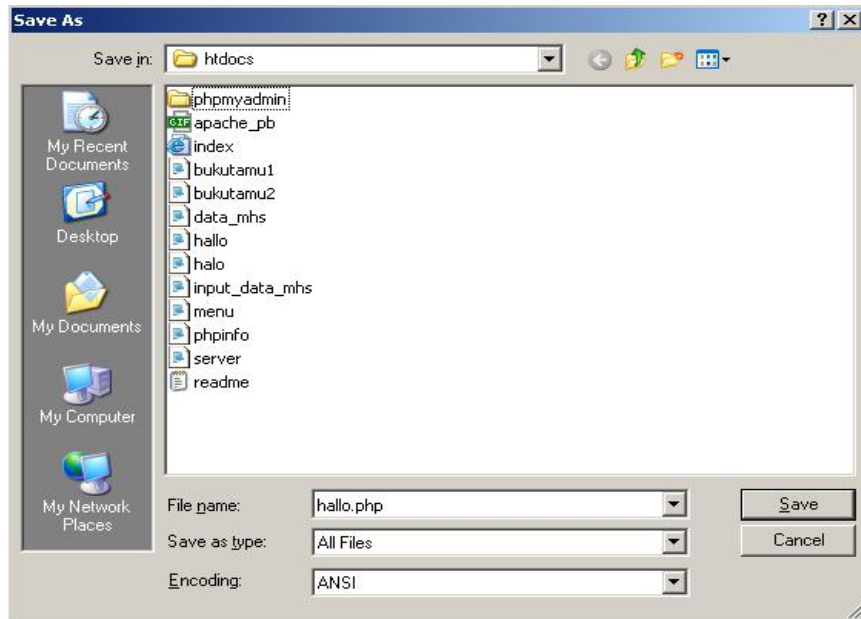
File **PHP** pada dasarnya merupakan file text biasa yang bisa dibuat menggunakan aplikasi notepad bawaan windows, namun untuk penggunaan yang lebih baik, kita bisa pakai aplikasi text editor khusus programming seperti **Notepad++**, **Atom**, **VS Code** atau **Sublime Text**. Teks editor yang paling ringan adalah Notepad++, aplikasi ini hanya berukuran 4MB dan sudah cukup untuk membuat perintah PHP sederhana.

1. Buka program **Notepad**, lalu tuliskan skrip PHP seperti pada gambar 10.



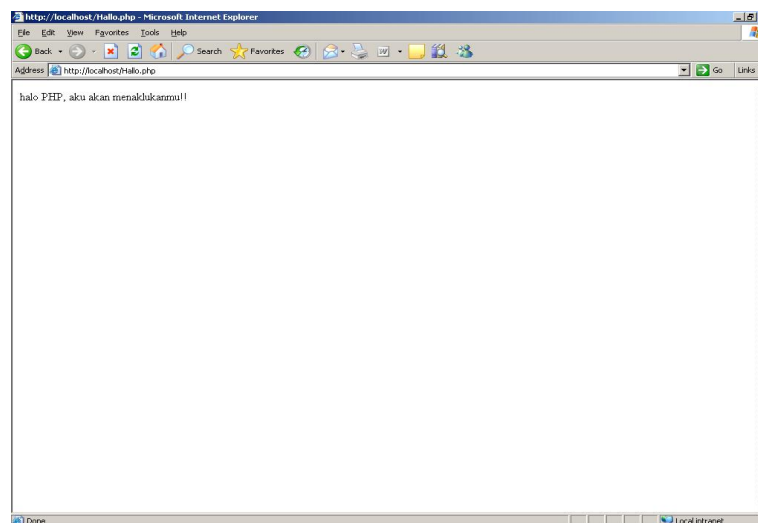
Gambar 10. Menulis Skrip PHP di Notepad

- Kemudian, klik menu **File > Save As**, maka akan tampil kotak dialognya, pilih pada bagian **Save in:** htdoc (C:\XAMPP\htdocs), lalu isikan pada bagian **Save As type** : All files dan **File name** : Hallo.php. Lihat gambar 11.



Gambar 11. Kotak Dialog Save As

- Hidupkan server **Apache**, lalu buka **browser** dan isikan pada bagian **address** : <http://localhost/Hallo.php>, maka akan tampil hasilnya.



Gambar 12. Menjalankan Skrip PHP di Browser

### 3.2 Memasukkan kode PHP kedalam HTML

PHP menyediakan 4 cara untuk masuk ke dalam mode PHP. Namun padap PHP 7 beberapa cara sudah tidak bisa digunakan.

#### 1. Metode Standar (XML Style PHP)

Untuk masuk ke dalam mode PHP, kita menggunakan kombinasi "<?php" sebagai tag pembuka dan tag "?>" sebagai tag penutup.

contoh:

```
1 <?php
2     echo "Hello, world";
3 ?>
```

#### 2. Metode tag singkat (SGML Style) PHP

Metode SGML hanya menggunakan tag <? sebagai pembuka mode PHP, dan tag ?> sebagai tag penutup.

contoh:

```
1 <?
2     echo "Hello, world";
3 ?>
```

#### 3. Metode ASP (ASP Style) PHP

Metode ASP menggunakan tag <% sebagai tag pembuka, dan tag %> sebagai tag penutup. Mulai dari **PHP 7.0**, ASP style tag ini sudah tidak didukung lagi oleh PHP, sehingga tidak bisa di pakai lagi.

Contoh:

```
1 <%
2     echo "Hello, world";
3 %>
```

#### 4. Metode Script (Script Style) PHP

Metode Script menggunakan tag <script language="php"> sebagai tag pembuka, dan tag </script> sebagai tag penutup. Mulai dari **PHP 7.0**, ASP style tag ini sudah tidak didukung lagi oleh PHP, sehingga tidak bisa di pakai lagi.

Contoh:

```
1 <script language="php">
2     echo "Hello, world";
3 </script>
```

Selain keempat metode di atas, terdapat satu lagi cara untuk masuk ke mode PHP. Terjemahan bebasnya saya sebut saja sebagai metode **PHP instant**, karena cara ini menggabungkan metode **PHP SGML** dan perintah **echo** dalam satu tag. Tag pembuka untuk metode **PHP instan** ini adalah **<?=** dan **?>** sebagai tag penutup. Contoh penggunaannya adalah sebagai berikut:

```
1 Saya Sedang Belajar PHP di <?= "Gunadarma" ?>
```

Hal ini sebenarnya sama dengan :

```
1 Saya Sedang Belajar PHP di <?php echo "Gunadarma" ?>
```

Metode instan ini sering dipakai untuk masuk ke PHP mode secara cepat dan tidak memerlukan kode yang panjang. Misalnya dalam pembuatan form, karena akan lebih mudah masuk PHP mode pada saat dibutuhkan saja dari pada seluruh form dibuat dengan **PHP**:

```
1 <input type="text" name="nama_user" value="<?= "$nama"; ?>">
```

### 3.3 Membuat Koneksi Database

Sebelum melakukan operasi apapun di MySQL, hal pertama yang harus dilakukan tentunya adalah membuka koneksi dengan MySQL.

1. Fungsi pertama kali yang digunakan untuk membuka koneksi dengan MySQL.

```
mysql_connect($host, $username, $password);
```

2. Memilih database yang akan digunakan.

```
mysql_select_db($db,$link_id);
```


### Contoh:

```
<?php
//buat koneksi MySQL untuk user: root, tanpa password, alamat: localhost
$link=mysql_connect('localhost','root','');

//cek apakah koneksi dengan MySQL berhasil
if ($link)
{
    //koneksi berhasil
    echo "Koneksi dengan MySQL berhasil";
}
else
{
    //koneksi gagal
    echo "Koneksi dengan MySQL gagal";
}

//memeriksa nilai dari $link
echo "<br />";
echo 'hasil var_dump variabel $link : ';
var_dump($link);

?>
```





**JUNIOR WEB  
PROGRAMMER**

**J.620100.005.02**

**Mengimplementasikan User  
Interface**



## 2

## MENGIMPLEMENTASIKAN USER INTERFACE

### Objektif:

3. Mengidentifikasi rancangan user interface
  4. Melakukan implementasi rancangan user interface
- 

Interaksi pengguna dengan sistem dapat dilakukan dengan berbagai cara, antara lain:

- Membaca dan menafsirkan informasi mengenai bagaimana menggunakan sistem
- Memberikan perintah pada sistem untuk melakukan tugas tertentu
- Memasukkan kata atau angka ke dalam sistem sebagai suatu data
- Membaca dan menafsirkan hasil yang dihasilkan oleh sistem dalam tampilan layar atau laporan tercetak
- Menanggapi dan mengoreksi kesalahan

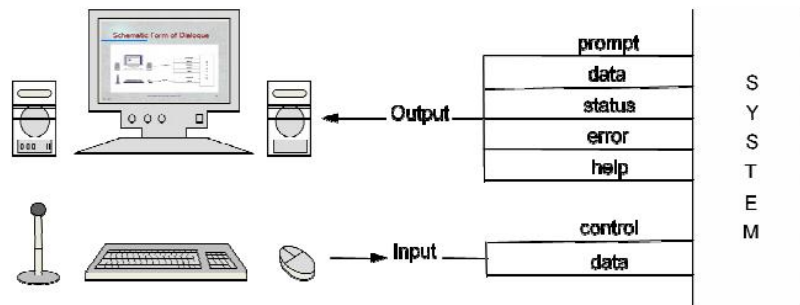
### 1. Mengidentifikasi Rancangan User Interface

Metafora adalah istilah yang digunakan secara kiasan untuk menjelaskan sesuatu namun tidak dapat diartikan secara harfiah. Dalam interaksi manusia dengan komputer ada 2 jenis pendekatan metafora, yaitu:

#### 1. Metafora Dialog;

Komunikasi yang terjadi antara manusia dengan komputer dianggap semacam dialog. Tidak ada pembicaraan secara nyata namun manusia memberikan pesan ke komputer kemudian komputer merespon pesan tersebut. Komputer merespon dengan berbagai cara untuk kemudian mendorong manusia merespon kembali, dan demikian seterusnya.

Bentuk skema dialog antara manusia dengan komputer digambarkan sebagai berikut.



Gambar 1.2 Skema Bentuk Dialog Manusia dengan Komputer

Tabel 1.4 berikut adalah tipe *message* yang terdapat pada metafora dialog.

Tabel 1.4 Tipe Message pada Dialog

Output	Prompt	Meminta inputan dari <i>user</i>
	Data	Data dari aplikasi yang diminta oleh <i>user</i>
	Status	Pemberitahuan mengenai kondisi tertentu
	Error	Proses yang tidak bisa dilanjutkan
	Help	Informasi tambahan untuk <i>user</i>
Input	Control	Mengarahkan <i>user</i> untuk melanjutkan dialog
	Data	Data yang diberikan oleh <i>user</i>

## 2. Metafora Manipulasi Langsung;

Interface memberi kesan *user* memanipulasi objek-objek pada layar dengan menggunakan *mouse*. Manipulasi yang dilakukan antara lain:

- *Drag dan drop icon*
- Mengecilkan atau memperlebar *window*
- Menekan *button*

## 2. Rancangan User Interface Diidentifikasi sesuai Kebutuhan

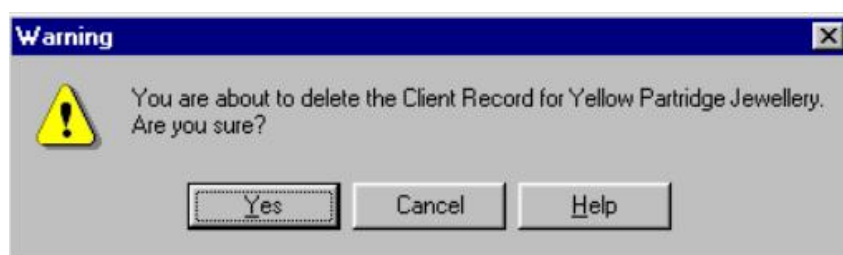
Ada beberapa *point* penting yang harus diperhatikan dalam merancang sebuah dialog yang baik, antara lain:

### 1. Konsistensi (*consistency*);

Rancangan dialog yang konsisten membantu pengguna untuk mempelajari aplikasi. Sebaiknya sebuah organisasi memiliki standar rancangan dialog yang konsisten. Misalkan tombol F2 digunakan untuk menyimpan dan sebagainya. Dapat juga digunakan panduan gaya dialog yang diterapkan oleh Microsoft atau Apple sebagai referensi.

### 2. Dukungan pengguna yang tepat;

Rancangan dialog mampu memberikan pesan kesalahan serta peringatan dengan tepat. Jika pengguna melakukan kesalahan maka dialog tersebut harus mampu membantu dalam mengatasi hal tersebut. Sebaiknya hindari *content* tersembunyi khususnya untuk rancangan berbasis web. Pesan kesalahan yang diberikan harus informatif dan tidak samar. Gunakan istilah yang umum sehingga pengguna tahu pada bagian apa terjadi kesalahan. Gunakan pesan untuk mencegah terjadinya kesalahan pengguna namun jangan berlebihan sehingga cenderung membingungkan.



Gambar 1.3 Contoh Rancangan Dialog Peringatan

### 3. Umpan balik yang memadai,

Pengguna cenderung mengharapkan respon ketika mereka menekan atau mengklik tombol tertentu. Jika mereka tidak mendapatkan balasan, maka pengguna cenderung untuk mencoba lagi atau

menekan tombol yang lain. Kadang-kadang penekanan tombol tersebut mengakibatkan hasil yang tidak terduga. Gunakan umpan balik sebaik mungkin untuk menghindari adanya tindakan yang berlebihan dari pengguna.

4. Inputan dari *user* minimal;

Cobalah untuk merancang sistem sehingga pengguna tidak harus menekan tombol atau mengklik *mouse* yang tidak perlu. Hal tersebut dapat dilakukan dengan cara:

- Gunakan kode atau singkatan
- Biarkan pengguna memilih dari daftar, bukan memasukkan
- Biarkan pengguna memperbaiki nilai yang salah dari pada melakukan pengetikan ulang
- Berikan informasi secara otomatis kepada pengguna
- Gunakan *default*

**3. Komponen User Interface Dialog Diidentifikasi sesuai Konteks Rancangan Proses**

Keterampilan yang diperlukan dalam mengidentifikasi rancangan *User Interface* adalah:

- a. Mengidentifikasi kebutuhan rancangan *user interface*
- b. Mengidentifikasi komponen *user interface* yang akan digunakan
- c. Mengatur urutan akses komponen *user interface*
- d. Membuat simulasi *mock-up* menggunakan *interface mockup tool* dan *software engineering tool*.

Sikap kerja yang diperlukan dalam mengidentifikasi rancangan *User Interface* adalah:

1. Menunjukkan hasil identifikasi rancangan *user interface* secara cermat, tepat, teliti dan sesuai kebutuhan

2. Menyajikan hasil identifikasi rancangan *user interface* secara cermat, tepat, teliti dan sesuai konteks rancangan proses. sesuai kebutuhan
3. Menunjukkan urutan dari akses komponen *user interface* dialog secara cermat, tepat, dan teliti sesuai kebutuhan
4. Menyajikan hasil pembuatan simulasi (*mock-up*) dari aplikasi yang akan dikembangkan secara cermat, tepat, dan teliti sesuai kebutuhan.

Beberapa cara dilakukan dalam merancang dan mengimplementasi elemen pada tampilan sehingga dapat mendukung interaksi dengan pengguna. Pilihan rancangan yang diterapkan dipengaruhi oleh:

- Sifat yang dimiliki pengguna
- Jenis Pengguna
- Jumlah pelatihan yang diterima pengguna
- Frekuensi Penggunaan
- *Hardware* dan arsitektur Perangkat lunak

Semua pendekatan dilakukan dengan beberapa langkah, yaitu:

- Mengumpulkan kebutuhan (*requirement gathering*)
- Merancang tampilan (*design of interface*)
- Melakukan evaluasi tampilan (*interface evaluation*)

Pada pendekatan formal, hal yang harus diperhatikan adalah:

1. Pendekatan struktur;

Model pendekatan ini berkaitan dengan pendekatan terstruktur yang lazim digunakan dalam merancang dan menganalisa pada era tahun 1980 sampai dengan 1990-an. Model siklus hidup digunakan sebagai langkah atau tahapan. Pada bentuk ini memungkinkan adanya kegiatan yang dilakukan secara paralel.

Keuntungan:

- Tahapan dan langkah-langkah dilakukan secara lebih terinci sehingga memudahkan dalam manajemen proyek

- Menyediakan standar dan dokumentasi sehingga meningkatkan komunikasi
- Spesifikasi lebih komprehensif sehingga memudahkan untuk menghasilkan sistem dengan mutu yang baik

Pendekatan ini fokus pada pemahaman tugas dan mengalokasikan tugas antara pengguna dengan sistem. Contoh pendekatan ini adalah STUDIO (*Terstruktur User-interface Desain untuk Optimisation Interface*). Teknik yang digunakan pada STUDIO adalah:

- task hierarchy diagrams*
- knowledge representation grammars*
- task allocation charts*
- statecharts*

2. Pendekatan ethographic;

Pendekatan ini berakar dari pendekatan *etnografi* dalam sosiologi dan antropologi. Peneliti berusaha untuk terlibat dalam situasi untuk kemudian belajar dari situasi tersebut. Hanya dengan cara ini situasi dapat dipahami dengan baik. Pendekatan ini lebih memperhatikan kualitatif dari pada kuantitatif.

3. Pendekatan berdasarkan skenario;

Pendekatan ini cenderung informal daripada pendekatan terstruktur dan *etnografi*. Pendekatan ini menggunakan skenario sebagai alat dalam merancang antar muka dan mengevaluasi antarmuka. Skenario dapat berupa:

- *textual descriptions*
- *storyboards*
- *prototypes*
- *video mock-ups*

#### **4. Urutan dari Akses Komponen User Interface Dialog Dijelaskan**

Komponen antar muka grafis adalah bentuk bentuk tampilan yang dapat dilihat dan umumnya bersifat tetap yang selalu digunakan pada sebuah program aplikasi komputer. Komponen antar muka grafis bertujuan untuk memenuhi salah satu kriteria interaksi antara manusia dan komputer yakni agar mendapat perhatian pengguna kepada program aplikasi yang dibuat, tentunya sebuah program aplikasi haruslah mempunyai tampilan yang menarik perhatian pengguna.

Tujuan perancangan antarmuka pada dasarnya adalah untuk mendapatkan satu kriteria yang sangat penting dalam pengoperasian sebuah program aplikasi, yakni aspek ramah dengan pengguna (user friendly).

##### **Komponen Dasar Antarmuka Grafis**

Komponen antar muka bekerja pada mode teks ataupun mode grafis. Dari sekian banyak antar muka grafis yang dipakai pada paket-paket program aplikasi, sering ditemukan bentuk-bentuk yang selalu tetap. Komponen itulah yang disebut dengan komponen dasar antar muka grafis.

1. Spin Box, adalah salah satu komponen antar muka grafis yang digunakan untuk mengatur suatu nilai peubah, biasanya bertipe numerik, dengan menambahkan atau mengurangkankannya dengan suatu nilai tertentu. Dalam spin box nilai maksimum dan minimum peubah harus dinyatakan dengan jelas, supaya tombol ini tidak memutar (spining) terus menerus.
2. List Box, digunakan untuk menampilkan sejumlah pilihan yang tersedia yang dapat dipilih oleh pengguna program. Setelah pengguna memilih sekian banyak pilihan yang tersedia, pilihan terpilih akan disimpan sebagai nilai dari sebuah peubah. Combo box mirip dengan

list box, tetapi memiliki perbedaan dalam penampilannya. Pada list box beberapa pilihan dapat dilihat langsung oleh pengguna, sementara pada combo box pilihan-pilihan yang ada tidak akan terlihat kecuali pengguna menekan tombol anak panah yang melekat pada tombol combo box itu.

3. Check Box, komponen ini digunakan untuk memilih satu atau lebih pilihan dari jumlah pilihan yang tersedia. Komponen ini cocok dipakai apabila pemrogram lebih memberi kebebasan kepada pengguna untuk membuat lebih dari satu pilihan.
4. Tombol radio, berbeda dengan check box, tombol radio hanya memungkinkan pengguna untuk memilih satu pilihan diantara beberapa pilihan yang tersedia. Contoh nyatanya adalah pemilihan jenis kelamin.
5. Tombol tekan, biasanya digunakan untuk mengaktifkan suatu aktivitas apabila tombol tersebut ditekan menggunakan mouse (klik mouse). Jenis tombol ini disebut tombol tekan karena tombol ini seolah-olah masuk kedalam layar komputer, seperti halnya anda menekan sebuah tombol. Teknik penampilan tombol tekan pada layar perlu diperhatikan, sehingga penampilan tombol tekan tersebut terkesan merupakan tombol 3 dimensi yang ketika tombol tersebut dipilih, terlihat seperti menekan tombol yang sebenarnya.

Berikut ini adalah hal-hal yang perlu diperhatikan dalam pembuatan antar muka berbasis form filling:

1. Proteksi tampilan. Adanya pembatasan agar pengguna tidak dapat mengakses semua tampilan yang ada di layar monitor.
2. Batasan medan tampilan. Medan data dapat ditentukan mempunyai panjang yang tetap atau berubah, menggunakan format bebas atau tertentu.



3. Isi medan. Pengguna biasanya mempunyai sejumlah gambaran tentang isi medan yang diperbolehkan. Petunjuk pengisian dapat juga ditampilkan sebagai bagian dari tampilan
4. Medan opsional. Beberapa medan isian dapat bersifat opsional. Medan opsional dapat dinyatakan secara tekstual atau menggunakan aturan tertentu, seperti penggunaan warna berintensitas rendah, warna tampilan yang berbeda, dan lain-lain.
5. Default. Apakah dalam medan isian dimungkinkan adanya nilai default. Jika ya, tentukan tempatnya, apakah pada bagian yang tidak dapat diakses pengguna ataukah pada bagian pemasukan data.
6. Bantuan. Adanya bantuan (help) yang menunjukkan cara pengisian borang, tetapi harus terpisah dari bentuk dasar borang. Jika terdapat fasilitas bantuan, pengguna perlu diberitahu cara mengakses.
7. Medan penghentian. Masukan data dalam medan dapat diakhiri dengan jalan menekan tombol Enter atau Return atau mengisi karakter terakhir dengan karakter tertentu atau dengan cara berpindah ke medan lain.
8. Navigasi. Kursor dapat digerakan di sekeliling layar menggunakan tombol Tab untuk urutan yang tetap, atau dapat pula digerakkan secara bebas menggunakan mouse.
9. Pembetulan kesalahan. Pengguna dapat membetulkan kesalahan dengan menggunakan tombol BackSpace dengan menindahi (overwrite) isian lama, dengan jalan membersihkan dan mengisi kembali medan tersebut, dan lain-lain.
10. Penyelesaian. Perlu diperhatikan cara yang digunakan untuk memberitahu pengguna bahwa seluruh proses pengisian telah selesai.

## 5. Simulasi (Mock-up) dari Aplikasi yang akan Dikembangkan Dibuat

Mock-up adalah sebuah bentuk realistis dari sebuah karya digital, bertujuan untuk pemanis atau menjadi contoh nyata bagaimana sebuah karya digital bekerja, sebelum benar – benar dicetak, biasanya untuk klien atau calon klien dan bertujuan untuk presentasi sebuah karya. Perancang seharusnya membuat dokumentasi akan bentuk-bentuk tampilan yang akan diimplementasikan. Peranti bantu sederhana yang dapat digunakan misalnya adalah lembar kerja tampilan/LKT (screen design work sheet).

Peranti bantu sederhana yang dapat digunakan misalnya adalah lembar kerja tampilan/LKT (screen design work sheet). Pada LKT, disajikan empat bagian:

- Nomor lembar kerja
- Bagian tampilan
- Bagian navigasi
- Bagian keterangan

No. sheet: 5

NAVIGASI:

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

PEMISALAN DATA PRIBADI

Nama:

Alamat:

Kota:

Tgl. Lahir:

Kelamin: ☒ Laki-laki ☐ Perempuan

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

Simpan Batal Selesai

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.

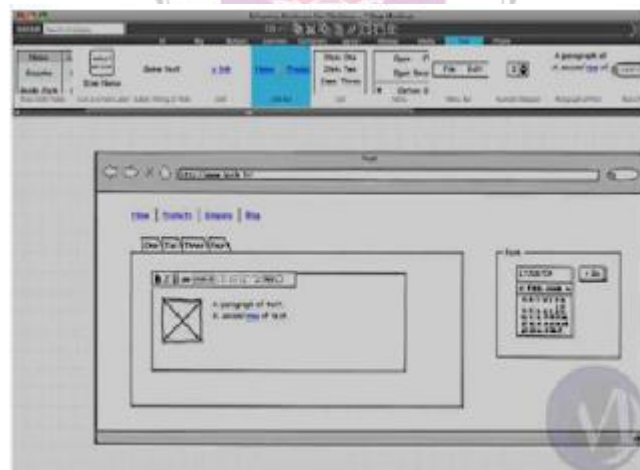
Gambar 1.4 Contoh Screen Design Worksheet

Dalam membuat sketsa user interface diperlukan waktu yang cepat sebagai tahap awal desain untuk mengeksplorasi banyak alternatif, untuk memungkinkan komunikasi dalam tim desain, dan untuk menyampaikan

kepada klien seperti apa produk itu. User interface dapat dibuat dengan kertas dan pensil, pengolah kata, atau perangkat lunak presentasi slide-show (seperti Microsoft PowerPoint atau Apple Keynote). Perancang yang ahli juga telah membuat prototip antarmuka pengguna dengan alat konstruksi multimedia, seperti Macromedia Director, Flash MX Dreamweaver, visio, balsamiq, iplotz, wireframe dan lain-lain. Program ini dapat dengan cepat menghasilkan program animasi atau bahkan interaktif dan didistribusikan melalui Web. Berikut ini beberapa contoh interface mockup tool.

#### 1. Balsamiq;

Balsamiq Mockup Tools merupakan aplikasi terbaik, untuk membuat sketsa gambar Blueprint sebuah blog ataupun website. Cara penggunaannya pun cukup mudah, hanya tinggal tarik dan letakan. Jika ingin menambahkan gambar tinggal drop and drag saja sesuai keinginan. Fitur lainnya pun dengan mudah digunakan.



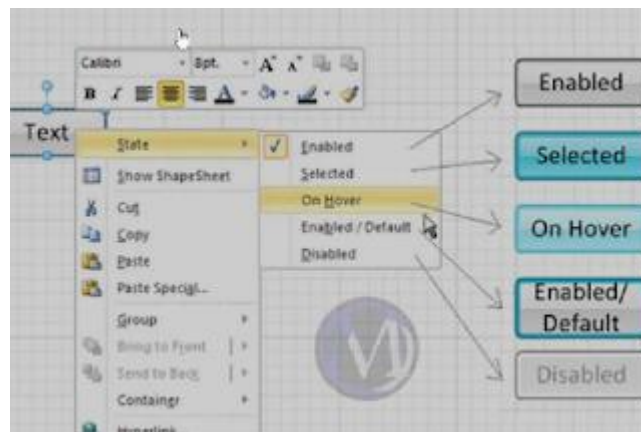
Gambar 1.5 Balsamiq Mock-up Tools

#### 2. Microsoft Visio;

Microsoft Visio ini sangatlah mudah untuk digunakan, fitur yang lengkap membuat kita tidak perlu lagi mencari gambar yang ingin kita gunakan dalam pembuatan mockup sebuah website ataupun blog.

Kelebihan Visio adalah kemampuan diagram teknis dari pada pembuatan mockup.

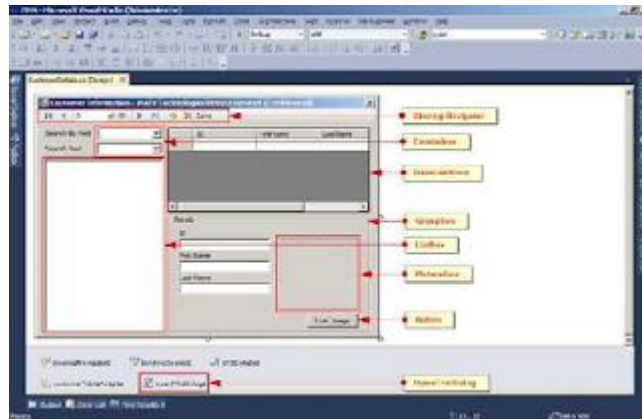
Microsoft Visual Studio dapat digunakan untuk mengembangkan aplikasi dalam native code (dalam bentuk bahasa mesin yang berjalan di atas Windows) ataupun managed code (dalam bentuk Microsoft Intermediate Language di atas .NET Framework). Selain itu, Visual Studio juga dapat digunakan untuk mengembangkan aplikasi Silverlight, aplikasi Windows Mobile (yang berjalan di atas .NET Compact Framework).



Gambar 1.6 Microsoft Visio Mockup Tools

### 3. Netbeans;

Netbeans adalah sebuah aplikasi Integrated Development Environment (IDE) yang berbasis Java dari Sun Microsystems yang berjalan di atas swing. Swing merupakan sebuah teknologi Java untuk pengembangan aplikasi dekstop yang dapat berjalan pada berbagai macam platform seperti windows, linux, Mac OS X dan Solaris. Sebuah IDE merupakan lingkup pemrograman yang di integrasikan ke dalam suatu aplikasi perangkat lunak yang menyediakan Graphic User Interface (GUI), suatu kode editor atau text, suatu compiler dan suatu debugger.



Gambar 1.7 Netbeans Visio Mockup Tools

## 2. Melakukan Implementasi Rancangan User Interface

Empat layer perangkat lunak yang tersedia untuk membangun user-interface seperti terlihat pada tabel 1.5 berikut.

Tabel 1.5 Software Layer

Software Layers	Visual Tools	Contoh
a. Windowing System	Resource Editor	Microsoft Win32/GDI+
		Apple Quartz
		XII Windowing System
b. GUI Toolkit	Interface Builder	Eclipse
		Borland Jbuilder
		Microsoft Visual Studio
c. Application Framework Specialized Language	Conceptual Building Tools	Macromedia Director
		Microsoft MFC
d. Application	Model-Based	Microsoft Access
	Building Tools	Sybase PowerDesigner

User Interface adalah tampilan yang bisa kita lihat ketika kita mengoperasikan suatu aplikasi, baik aplikasi Android, IOS dan aplikasi website. User interface lebih mengarah ke sisi tampilan mulai dari skema warna, besar kecil dari tulisan, gambar ilustrasi, tombol, label dan lain-lain.

User Interface memperhatikan layouting, penempatan suatu komponen. Dimana letak gambar, di mana letak text, apakah judul harus berukuran besar atau bahkan kecil, apakah suatu aplikasi cukup menggunakan 2 warna, 3 warna, 4 warna atau bahkan lebih dari 5 warna.

User experience adalah pengalaman pengguna. User experience design adalah proses dengan tujuan supaya aplikasi yang kita buat itu nyaman digunakan, mudah dipelajari, tidak membingungkan dan menyelesaikan masalah yang sedang dialami oleh pengguna. seorang user experience designer harus melakukan riset ke user, masalah apa yang dialami oleh user.

## **2.1 Menu Program Sesuai dengan Rancangan Program Diterapkan**

Kriteria yang penting dari sebuah UI adalah tampilan yang menarik

- Desainer UI: harus punya jiwa seni, selera pengguna dan dapat meyakinkan programmer bahwa rancangannya dapat diimplementasikan dengan tools
- Desainer harus membuat dokumentasi rancangannya. 4 cara dokumentasi:
  - a. Sketsa pada kertas
  - b. Penggunaan piranti prototype GUI
  - c. Penjelasan tekstual
  - d. Menggunakan CASE

Pendekatan yang dapat dilakukan untuk perancangan:

### **1. User-Centered Design Approach**

Fokus pada tujuan dan tasks pengguna → melibatkan pengguna dalam perancangan UI. Digunakan pada special purpose software. Tidak difokuskan pada teknologi, programmer dan prioritas organisasi.

Software engineering dititikberatkan pada analisis user/interface dan Penekanan pada dokumentasi IMK, bukan coding.

## 2. User Design Approach

Pengguna yang menentukan “wajah” UI yang diinginkan digunakan pada general purpose software, kadang-kadang memberatkan programmer dalam implementasinya. Customization untuk setting UI oleh pengguna.

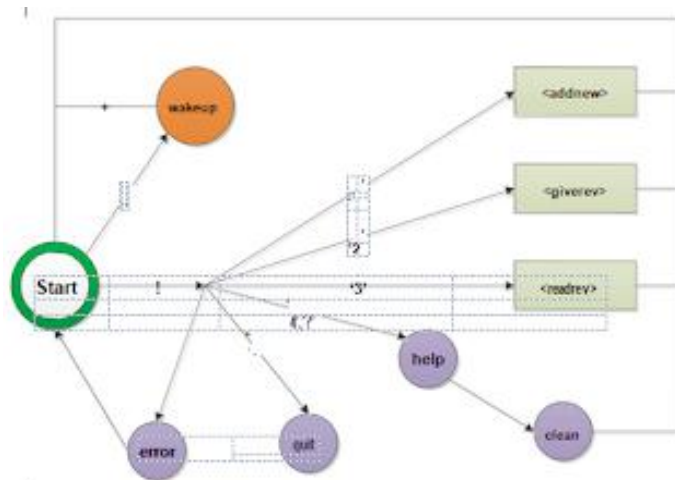
### 2.2 Penempatan User Interface Dialog Diatur secara Sekuensial

Dua pendekatan yang digunakan untuk menjelaskan urutan akses komponen dari user interface yaitu diagram transisi dan statechart.

#### 1. Diagram Transisi

Umumnya sebuah diagram transisi memiliki sekumpulan nodes yang merepresentasikan status sistem dan sekumpulan hubungan antara nodes yang merepresentasikan transisi yang mungkin. Setiap hubungan diberi label dengan aktivitas user yang memilih hubungan tersebut dan respon computer yang mungkin ada. Diagram transisi yang sederhana dalam Gambar 4 merepresentasikan sejumlah sistem pemilihan menu untuk tampilan sebuah restoran yang menunjukkan apa yang terjadi ketika pemakai memilih sejumlah pilihan:

- a. Menambah sebuah restoran ke daftar.
- b. Menyediakan sebuah tampilan dari sebuah restoran.
- c. Membaca sebuah tampilan.
- d. Memperoleh bantuan
- e. Keluar, atau karakter lainnya (pesan kesalahan).



Gambar 1.8 Contoh Diagram Transisi

Diagram transisi diterjemahkan langsung ke dalam finite-state automata, yang dipelajari dengan baik dalam ilmu komputer. Beberapa properti dapat diverifikasi secara otomatis, seperti reachability (apakah ada kemungkinan jalan untuk mencapai semua state) Atau keaktifan (apakah ada jalan keluar dari semua state?). Namun, sifat yang berpusat pada pengguna seperti visibilitas status sistem sangat sulit untuk diungkapkan dan diverifikasi dalam hal automata.

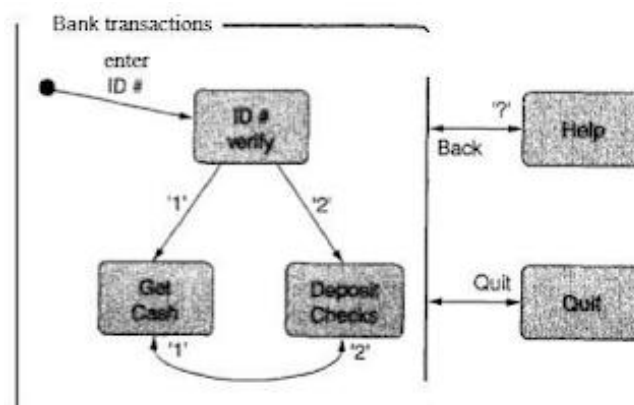
Sayangnya, diagram transisi tidak sesuai dengan sistem yang semakin berkembang, dan terlalu banyak transisi yang dapat menyebabkan tampilan yang rumit. Peningkatannya dapat dilakukan dengan menggantikan sebuah node dari transisi status dengan sebuah tampilan cetakan untuk memberikan tampilan bahwa ada perpindahan melalui tampilan dan kotak dialog. Penjelasan semacam ini berguna dalam perancangan dan pelatihan. Perancangan untuk interface dengan ratusan kotak dialog, atau website dengan ratusan tampilan, lebih mudah untuk dipelajari ketika digunakan.



## 2. Statecharts

Walaupun diagram transisi efektif untuk alur atau aksi dan untuk menelusuri jejak dari status, semua itu dapat dengan cepat menjadi luas dan memusingkan. Modularitas memungkinkan jika nodes ada bersama subgraph, tapi strategi ini bekerja dengan baik hanya secara berurutan. Diagram transisi juga menjadi memusingkan ketika setiap node harus menunjukkan hubungan ke sebuah status bantuan, melompat ke status sebelumnya, dan sebuah status keluar. Konkurensi dan sinkronisasi kurang dapat direpresentasikan oleh diagram transisi, walaupun beberapa variasi seperti petri-nets dapat membantu. Alternative yang dapat mengatasi masalah ini adalah statecharts, yang memiliki beberapa kebaikan untuk menspesifikasikan interface. Karena sebuah fitur pengelompokan ditawarkan melalui lingkaran yang bersarang, transisi yang berulang dapat difaktorkan ke luar lingkaran.

Penambahan lainnya dalam statecharts seperti konkurensi, kejadian dari eksternal, dan aksi dari pengguna direpresentasikan di dalam Statemaster, yang merupakan sebuah peralatan User Interface berdasarkan statecharts. Statecharts juga dapat diperluas dengan alur data dan spesifikasi batasan.



Gambar 1.9 Statechart Sistem yang Disederhanakan menggunakan Pengelompokan State

Urutan dalam melakukan perancangan perlu diperhatikan. Urutan kegiatan yang dilakukan diatur secara sequential adalah:

a. Pemilihan ragam dialog

Tergantung dari karakteristik pengguna, kemudian baru disiapkan tipe dialog yang diperlukan dan kemudian perkiraan kendala teknologi untuk implementasi nantinya juga perlu diperhatikan.

b. Desain struktur dialog

Melakukan analisis tugas dan penentuan model penggunaanya serta melibatkan pengguna untuk mendapatkan umpan balik pembuatan rancangan.

c. Desain format pesan

Perlu diperhatikan tata letak tampilan dan keterangan tekstual, dan dipertimbangkan efisiensi entry data bagi pengguna.

d. Desain penanganan kesalahan

Dipertimbangkan bagaimana validasi pemasukan data → ada mekanisme pengulangan pemasukan data atau tidak. Bagaimana dengan proteksi pengguna → ada peringatan atau tidak. Tidak ketinggalan perlu diperhatikan bagaimana pemulihan dari kesalahan → ada mekanisme untuk membatalkan serta penampilan pesan kesalahan.



Gambar 1.10 Contoh Dialog Pesan Kesalahan

e. Desain struktur data

Menentukan struktur data yg digunakan untuk menyajikan dan mendukung fungsionalitas. Menentukan komponen UI, memetakan

struktur data ke model pengguna yang dibuat dan diturunkan dari spesifikasi UI.

Organisasi elemen layar dapat dilihat dari beberapa sisi, yaitu:

1. Keseimbangan

Keseimbangan elemen dalam layar, apakah kiri ke kanan ataukah atas ke bawah. Kolom kiri ke kanan seharusnya dibuat sama, kedua kolom diusahakan memiliki elemen yang seimbang.



Gambar 1.11 Contoh Design Seimbang dan Tidak Stabil

2. Simetri

Dilihat dari jumlah element kiri dan kanan terhadap garis tengah. Perlu diperhatikan apakah kolom kiri dan kanan diproses secara sama ataukah terpisah.



Gambar 1.12 Contoh Simetri dan Unsimetri

3. Keteraturan

Selalu buat spasi standar dan konsisten secara horisontal maupun vertikal. Buat keteraturan apabila beberapa kolom diproses dengan cara yang sama. Pemrosesan lokasi dan ukuran untuk setiap objek.

4. Dapat diprediksi

Tempatkan objek pada lokasi yang mudah diprediksi pada layar. User lebih menyukai judul dan menu bar di layar atas. Layar harus diproses secara lengkap – objek tidak berada pada tempat yang diharapkan.

## 5. Keterurutan

Membantu mata bekerja secara wajar, karena mata tertarik pada:

- Elemen yang lebih terang dari elemen lain
- Elemen yang terpisah dari suatu kelompok
- Gambar sebelum teks
- Warna sebelum monokrom
- Warna variasi vs. warna konstan
- Daerah yang lebih gelap
- Elemen besar vs. elemen kecil
- Bentuk-bentuk yang tak biasa

## 6. Ekonomi

Gunakan beberapa style, font, warna, teknik display, dialog style, dan lain-lain.

## 7. Kesatuan

Buat item-item muncul sebagai satu kesatuan

- Gunakan bentuk, ukuran dan warna yang sama
- Berikan sedikit tempat kosong antar elemen

## 8. Proporsi

Buat pengelompokan data atau teks secara proporsional.

## 9. Kesederhanaan

Kurangi jumlah pengelompokan, gunakan sedikit kolom untuk menampilkan elemen dalam layar.

## 10. Pengelompokan

Kombinasikan elemen untuk meminimalkan jumlah objek layar dalam pembatasan yang tetap jelas. Pemrosesan biasanya didasarkan dari pengelompokan objek dialog.

Gunakan pengaturan visual untuk pengelompokan elemen layar. Pengelompokan dapat dilakukan dalam bentuk jajarkan elemen dalam group, rata spasi antar elemen dalam group atau dengan memberikan

pemisahan antar group. Pengelompokan dapat dilakukan untuk group elemendengan tambahkan secara hemat warna dan batas menambah kompleksitas.

a. Simple Grouping

Elemen yang sama ditata scr vertikal

Jarak vertikal antar objek kecil

b. Boxed Grouping

Kotak memberi tambahan kompleksitas suatu isian/form

Pengaturan spasial secara memadai

c. Background Grouping

Warna memberi tambahan kompleksitas visual

Pengaturan spasial secara memadai

### 2.3 Setting Aktif-Pasif Komponen User Interface Dialog Disesuaikan dengan Urutan Alur Proses

*Grafical User Interface* merupakan bentuk dari *even driven*. *Manager window* merespon setiap peristiwa (*even*) dan merubah kondisi dari objek yang dikehendaki pada sistem *window*. Pada *interface* yang cukup rumit seperti Word Processor, *user* dapat memilih beberapa aksi kemudian sistem merespon secara benar mana yang dipilih. Berikut ini adalah contoh *user interface even driven* yang dibangun.



Gambar 1.13 Contoh Interface Even Driven

Pada kasus tersebut, *user* pertama kali memilih nama dari *client* pada *list box* yang diberi label *client*. Setelah *client* dipilih, maka *list* dari *campaign* dari *client* tersebut ditempatkan pada *listbox* yang diberi label *campaign*. Pada bagian ini jika belum *campaign* yang dipilih, maka *user* dapat mengklik tanda panah untuk menampilkan daftar *campaign* kemudian memilih *campaign* yang dimaksud. Setelah selesai memilih *campaign* maka klik *button check* maka program akan menampilkan total *cost* dari *campaign* tersebut. Jika nilainya negatif artinya *campaign* tersebut kelebihan *budged*. Hal lain yang diperhatikan jika belum ada *client* yang terpilih maka bisa men-*disable campaign listbox* dan *button check* sampai *client* dipilih. Bentuk *windows* ini disebut *dialog box windows GUI environments* yang mengkombinasikan antara dialog *user* dengan manipulasi *button* dan *listbox*.

Pada saat melakukan perancangan sistem, perancang umumnya menghadapi dua masalah penting yaitu:

- a. Bagaimana informasi dari user bisa disediakan untuk sistem komputer, misalnya pada saat input data
- b. Bagaimana informasi dari sistem komputer ditampilkan untuk user, apa hasil dari pemrosesan data

User interface yang baik harus menyatukan interaksi pengguna (user interaction) dan penyajian informasi (information presentation). Ada 5 tipe utama interaksi untuk user interaction:

1. Direct manipulation;

pengoperasian secara langsung: interaksi langsung dengan objek pada layar. Misalnya delete file dengan memasukkannya ke trash. Contoh: Video games.

- Kelebihan: Waktu pembelajaran user sangat singkat, feedback langsung diberikan pada tiap aksi sehingga kesalahan terdeteksi dan diperbaiki dengan cepat

- Kekurangan: Interface tipe ini rumit dan memerlukan banyak fasilitas pada sistem komputer, cocok untuk penggambaran secara visual untuk satu operasi atau objek
2. Menu selection;
- pilihan berbentuk menu: Memilih perintah dari daftar yang disediakan. Misalnya saat click kanan dan memilih aksi yang dikehendaki.
- Kelebihan: User tidak perlu ingat nama perintah. Pengetikan minimal. Kesalahan rendah.
  - Kekurangan: Tidak ada logika AND atau OR. Perlu ada struktur menu jika banyak pilihan. Menu dianggap lambat oleh expert user dibanding command language.
3. Form fill-in;
- pengisian form: Mengisi area-area pada form. Contoh: Stock control.
- Kelebihan: Masukan data yang sederhana. Mudah dipelajari
  - Kekurangan: Memerlukan banyak tempat di layar. Harus menyesuaikan dengan form manual dan kebiasaan user.
4. Command language;
- perintah tertulis: Menuliskan perintah yang sudah ditentukan pada program. Contoh: operating system.
- Kelebihan: Perintah diketikan langsung pada system. Misal UNIX, DOS command. Bisa diterapkan pada terminal yang murah. Kombinasi perintah bisa dilakukan. Misal copy file dan rename nama file.
  - Kekurangan: Perintah harus dipelajari dan diingat cara penggunaannya tidak cocok untuk user biasa. Kesalahan pakai perintah sering terjadi. Perlu ada sistem pemulihan kesalahan. Kemampuan mengetik perlu.

5. Natural language;

perintah dengan bahasa alami: Gunakan bahasa alami untuk mendapatkan hasil. Contoh: search engine di Internet.

- Kelebihan: Perintah dalam bentuk bahasa alami, dengan kosa kata yang terbatas (singkat) – misalnya kata kunci yang ditentukan untuk dicari oleh search engine. Ada kebebasan menggunakan kata-kata.
- Kekurangan: Tidak semua sistem cocok gunakan ini. Jika digunakan maka akan memerlukan banyak pengetikan.

## **2.4 Bentuk Style dari Komponen User Interface Ditentukan**

Sistem yang interaktif pasti menyediakan cara untuk menyajikan informasi untuk pengguna. Penyajian informasi bisa berupa penyajian langsung dari input yang diberikan (seperti teks pada word processing) atau disajikan dengan grafik.

Beberapa faktor berikut adalah hal yang perlu diperhatikan sebelum menentukan bentuk penyajian informasi:

- a. Apakah pengguna perlu informasi dengan ketepatan tinggi atau data yang saling berhubungan?
- b. Seberapa cepat nilai informasi berubah? Harus ada indikasi perubahan seketika?
- c. Apakah pengguna harus memberi respon pada perubahan?
- d. Apakah pengguna perlu melakukan perubahan pada informasi yang disajikan?
- e. Apakah informasi berupa teks atau numerik? Nilai relatif perlu atau tidak?

Informasi bisa bersifat statis atau dinamis ketika disajikan, masing-masing baik dengan karakteristik yang berbeda dan kebutuhan yang berbeda pula:



a. Static information:

- ✓ Ditetapkan saat awal sesi. Tidak berubah selama sesi berjalan.
- ✓ Bisa berupa informasi numeris atau teks Chart di MS-Excel
- ✓ Disajikan dengan jenis huruf khusus yang mudah dibaca atau diberi highlight dengan warna tertentu seperti pada Gambar 4 atau menggunakan icon yang mewakili

b. Dynamic information:

- ✓ Perubahan terjadi selama sesi berlangsung dan perubahan harus dikomunikasikan/ditunjukkan ke user
- ✓ Bisa berupa informasi numeris atau teks. Contoh: Defragmentation, scanning virus, download

Informasi dalam bentuk teks bersifat tepat dan berubah secara lambat sedangkan informasi dengan gambar/grafik mampu menjelaskan hubungan antar gambar, data bisa berubah dengan cepat. Seperti pada Gambar 2, informasi yang sama disajikan dengan dua cara yang berbeda. Jika yang dibutuhkan adalah hubungan antar data pada bulan-bulan tersebut, maka informasi dengan grafik memberikan informasi tentang hubungan tersebut lebih cepat dari pada informasi yang disajikan dengan teks dan numerik. Informasi dengan numerik dapat juga disajikan dengan cara digital atau analog dengan karakteristik sebagai berikut:

3. Digital presentation

- Singkat – hanya perlu sedikit tempat pada layar
- Ketepatan nilai ditunjukkan

4. Analogue presentation

- Nilai terlihat sambil lalu
- Untuk menunjukkan nilai relatif
- Mudah melihat data nilai yang berbeda

Penggunaan Warna pada desain Interface berfungsi untuk:

- Warna menambah dimensi ekstra pada suatu interface dan membantu user memahami struktur yang kompleks
- Bisa dipakai untuk mewarnai-terang (highlight) hal-hal khusus
- Kesalahan umum dalam penggunaan warna pada desain UI:
  - Menggunakan warna untuk mengkomunikasikan arti-- merah bisa jadi peringatan atau ada kesalahan
  - Terlalu banyak gunakan macam warna

Dalam menggunakan warna pada desain interface ada beberapa petunjuk yang dapat diikuti seperti berikut ini:

1. Hindari penggunaan terlalu banyak warna
2. Gunakan kode warna untuk mendukung operasi
3. Pengguna bisa kendalikan warna untuk kode
4. Desain monochrome kemudian tambahkan warna
5. Gunakan warna kode secara konsisten
6. Hindari pasangan warna yang tidak cocok/norak
7. Gunakan warna untuk menunjukkan perubahan status

User guidance meliputi semua fasilitas sistem untuk mendukung user termasuk on-line help, error messages, user manual. User guidance perlu disatukan dengan UI untuk bantu user saat membutuhkan informasi tentang sistem atau saat ada kesalahan. Help System dan sistem message (pesan kesalahan) adalah bentuk dari user guidance.

Error Messages sangat penting, karena error message yang buruk cenderung ditolak oleh user dan error message sebaiknya berpedoman pada faktor-faktor pada tabel 1.6 berikut.

Tabel 1.6 Faktor dalam Desain Error Message (Pesan Kesalahan)

Context	User guidance memberikan pesan sesuai konteks yang dialami oleh pengguna
Experience	Pesan yang panjang mengganggu bagi yang sudah biasa, tapi bagi pengguna baru pesan yang pendek membingungkan. User guidance perlu menjawab kebutuhan ini
Skill Level	Pesan dibuat sesuai dengan kemampuan pengguna dan pengalamannya
Style	Harus bersifat positif. Lebih baik bersifat aktif dari pada pasif. Harus sopan, tidak menghina atau bergurau
Culture	Jika mungkin, pesan disesuaikan dengan budaya. Mungkin kata atau warna yang digunakan disesuaikan dengan budaya setempat

## 2.5 Penerapan Simulasi Dijadikan suatu Proses yang Sesungguhnya

Simulasi diperlukan dengan beberapa alasan, antara lain:

1. Simulasi adalah satu-satunya cara yang dapat digunakan untuk mengatasi masalah, jika sistem nyata sulit diamati secara langsung. Contoh: Jalur penerbangan pesawat ruang angkasa atau satelit.
2. Solusi Analitik tidak bisa dikembangkan, karena sistem sangat kompleks.
3. Melengkapi urutan aksi yang mungkin terlewat/kurang.
4. Pengamatan sistem secara langsung tidak dimungkinkan, karena sangat mahal, memakan waktu yang terlalu lama dan khawatir akan merusak sistem yang sedang berjalan.

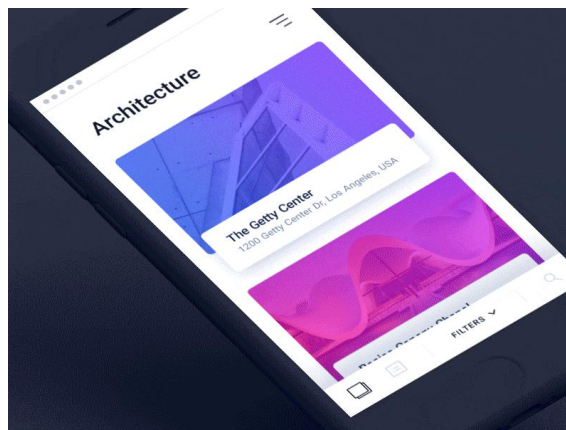
User Interface adalah sebuah tampilan yang digunakan pengguna untuk berinteraksi dengan komputer, tablet, smartphone maupun

perangkat elektronik lainnya melalui tampilan yang dapat dimengerti oleh pengguna. Setelah selesai melakukan persiapan, langkah berikutnya adalah membuat Wireframe, Mockup, dan Prototype.

Wireframe merupakan sebuah sketsa kasar tampilan aplikasi atau website. Biasanya digunakan untuk menyusun tata letak awal dalam suatu desain. Wireframe ini bisa dicoret-coret tangan atau menggunakan tools atau aplikasi khusus bikin wireframe. Biasanya wireframe dibuat dengan warna hitam putih yang lebih menekankan isi dari kontennya.

Mockup memberikan gambaran secara detail sebelum produk dibuat dengan menyampaikan aspek desain visual, termasuk gambar, warna, dan tipografi. Mockup dapat mengorganisir detail dari proyek, menerjemahkan ide ke dalam bahasa yang dapat dimengerti stakeholders, menyampaikan ide kepada anggota tim, implementasi desain, dan melihat perspektif user. Mockup bersifat statis dan tidak bisa dioperasikan.

Prototype adalah sebuah simulasi interaksi antara pengguna dan tampilan (interface) dimana user atau pengguna dapat melihat dan berinteraksi dengan user interface secara langsung. Mungkin tidak tampak persis seperti produk akhir, tetapi mendekati desain produk real-nya. Dengan ini harapannya kita mendapat masukan dari pengguna, bagian mana yang mudah dimengerti dan bagian mana yang susah mengerti atau bahkan bingung.



Gambar 1.14 Contoh Implementasi Prototype

Wireframe, mockup, dan prototype memiliki ciri khasnya masing-masing. Penggunaannya dapat disesuaikan dengan kebutuhan. Satu hal yang penting untuk diingat yaitu: buatlah wireframe, mockup, atau prototype dengan memikirkan user experience.

