

BAB III

METODE PENELITIAN DAN PERANCANGAN KARYA

3.1 Metode Penelitian

Pada bagian ini akan dijelaskan berbagai macam metode yang dilakukan dalam menyelesaikan Kerja Praktik. Adapun karya yang dibuat dalam Kerja Praktik ini adalah “Pembuatan Website Keyna Galeri Ploso Jombang” yaitu dengan beberapa tahapan sebagai berikut:

1. Survey Lapangan atau Observasi, kegiatan ini ditujukan untuk mengamati kegiatan seni dan multimedia apa saja yang dilakukan di “Keyna Galeri”.
2. Studi Kepustakaan, dilakukan untuk mendapatkan landasan teori yang sesuai dari tema dan permasalahan yang diangkat dan digunakan sebagai referensi untuk pelaksanaan rencana pengembangan sistem.
3. Analisa Permasalahan, dilakukan untuk mengetahui kebutuhan perusahaan dan juga sebagai proses pembandingan dengan website galeri lain pada umumnya. Seperti pengevaluasian pada desain, menu, konten dan kelengkapan informasi sebagai penunjang website.
4. Desain, setelah melakukan analisa dan evaluasi, tahap mendesain dilakukan dengan membuat rancangan berupa storyboard yang didalamnya terdapat ide, gambaran dan konsep website serta analisa warna yang sesuai dengan tema “Pembuatan Website Keyna Galeri Ploso Jombang”

5. Implementasi (Building and Coding), pada tahapan ini dilakukan implementasi dari perancangan dan desain yang telah dilakukan. Sehingga pada tahap ini akan menghasilkan Website jadi yang telah terkomposisi.
6. Pengujian (Testing), pengujian dilakukan setelah website dibangun. Bertujuan untuk menguji tingkat kelancaran saat user membuka website Keyna Galeri. Hal ini dilakukan untuk memastikan website yang dihasilkan dapat berjalan dengan baik.

3.2 Metode Perancangan Karya

3.2.1 Ide Pembuatan Website

Bila seorang mengunjungi sebuah situs web dan mengamati baik dari segi desain, grafis, komposisi warna atau tata letak, sekilas terlihat memiliki kesamaan dengan website lain. Terkadang baru tersadar setelah memperhatikan cukup lama, kadang dengan langsung melihatnya sebentar saja kita sudah dapat menebak bahwa sebuah web tersebut mirip dengan situs web lain.

Seperti dalam bidang lain, bidang desain web juga memiliki apa yang disebut *intertekstualitas*. Intertekstualitas mempercayai apa yang kita buat sekarang adalah proses adaptasi, pengaruh, aspirasi, atau inspirasi dari karya-karya yang telah ada. Jadi tidak ada karya yang betul-betul asli, tapi mengadaptasi dari karya terdahulu yang terus-menerus diturunkan sejalan dengan peradaban manusia. *Intertekstualitas* pada sebuah karya bisa berasal dari melihat, mendengar, atau meniru dari sebuah karya yang telah ada.

Dalam bukunya yang berjudul *Image-Music-Text*, Roland Barthes (1997: 146)

mengemukakan:

“A text is... a multidimensional space in which a variety of writings, none of them original, blend and clash. The text is a tissue of quotations... The writer can only imitate a gesture that is always anterior, never original. His only power is to mix writings, to counter the ones with the others, in such a way as never to rest on any one of them.”

Intertekstualitas ternyata tak hanya terdapat dalam karya-karya sastra, seperti puisi atau novel, tapi juga dalam bidang-bidang lain seperti bidang IT. Sehingga dapat disimpulkan bahwa ide dalam “Pembuatan Website Keyna Galeri Ploso Jombang” dilatarbelakangi oleh semakin berkembangnya website sebagai media promosi dan pengenalan suatu instansi perusahaan maupun usaha. Sedangkan dalam pembuatan desain dan sistemnya, memiliki sedikit kesamaan dengan desain website pada umumnya. Namun, website Keyna Galeri tetap memiliki ciri khas dan karakteristik yang berbeda dari website galeri lain baik dari segi warna, tampilan dan fasilitas yang tersedia.

3.2.2 Konsep Pembuatan Website

Website “Keyna Galeri” ini dilengkapi dengan berbagai menu-menu yang kompleks dan dinamis, seperti menu penampilan gambar/foto dengan teknik javascript dalam mode pop-up (overlay) disertai komentar dari pengunjung yang telah diintegrasikan dengan gambar atau foto yang bersangkutan sehingga penampilan gambar lebih menarik. Khusus untuk menu fotografi, penampilan gambar disertai pembacaan meta-data pada file foto, karena informasi tersebut berguna bagi para foto-

grafer yang membuka website “Keyna Galeri”. Penampilan video juga menggunakan JavaScript, yang mana bisa ditampilkan pada mode biasa maupun mode pop-up (overlay). Konsep video overlay ini merupakan konsep baru yang hanya diterapkan pada website “Keyna Galeri”.

3.2.3 Langkah Pembuatan Website

Pembuatan website “Keyna Galeri” yang dinamis dimulai dari konsep-konsep awal hingga memasuki proses coding dan desain. Untuk proses codingnya, akan dijelaskan tentang desain proses penampilan gambar atau video dengan teknik overlay Lightbox JavaScript.

Lightbox javascript adalah teknik JavaScript yang digunakan untuk menampilkan gambar besar dengan menggunakan modal dialog. Efeknya telah mendapatkan popularitas luas karena gaya yang sederhana dan dengan implementasi yang mudah.

Konsep sederhananya terdapat pada pengiriman nilai atribut HREF dari sebuah tautan menuju attribute SRC pada gambar atau video saat tautan tersebut di klik.



```
<a href="gambar1.jpg">Contoh</a>
↓

```

Gambar 3.1 Memindahkan nilai HREF ke dalam attribute SRC gambar di handle dengan menggunakan jQuery **.attr()**:

```
// Saat sebuah tautan (link) diklik...
```

```
$('#a').click(function() {
```

```
// Set nilai src pada gambar menjadi nilai href dari tautan yang diklik
```

```
$('#img').attr('src', this.href);
```

```
return false;
```

```
});
```

Disini, pada saat *thumbnail* atau gambar contoh yang ada di klik maka akan muncul gambar overlay yang besar, penampilannya dengan menggunakan selektor JQuery:

```
$('#a:has(img)')
```

Selector tersebut digunakan untuk menyeleksi semua elemen `<a>` (tautan/link) yang memiliki elemen `` (gambar) didalamnya. Selector yang lebih kompleks dengan menggunakan selector atribut untuk mendeteksi ekstensi akhir file pada URL (tautan/link) seperti berikut :

```
$('#a[href$=".jpg"], a[href$=".png"], a[href$=".gif"], [href$=".bmp"]')
```

Selector diatas hanya menyeleksi semua elemen `<a>` yang memiliki attribute HREF dengan nilai akhir .JPG, .PNG, .GIF dan .BMP yang merupakan URL gambar.

Selector JQuery sendiri memiliki elemen-elemen yang hampir sama persis seperti CSS.

Penyeleksian elemen di JQuery menggunakan kode:

`$('#container')`

Dalam proses penyeleksian JQuery mendapatkan bantuan dari **sizzle** yaitu mesin selector javaScript dengan orientasi CSS. Berikut merupakan aturan-aturan penulisan selector dalam CSS yang juga digunakan dalam website “Keyna Galeri”:

Tabel 3.1 Umum

Selektor	Contoh	Keterangan
*	<code>\$('*')</code>	Seleksi semua elemen
#id	<code>\$('#container')</code>	Seleksi semua elemen yang memiliki id container
.class	<code>\$('.clear')</code>	Seleksi semua elemen yang memiliki kelas clear
nama tag	<code>\$('img')</code>	Seleksi semua elemen

Tabel 3.2 Selektor Atribut

Selektor	Contoh	Keterangan
[atribut]	\$('[title]')	Seleksi semua elemen yang memiliki atribut title
[atribut="nilai"]	\$('[alt="image"]')	Seleksi semua elemen yang memiliki atribut alt dengan nilai image
x[atribut="nilai"]	\$('a[href="#"]')	Seleksi semua elemen <a> yang memiliki atribut href dengan nilai #
[atribut*="nilai"]	\$('a[href=*".com"]')	Seleksi semua elemen <a> yang memiliki atribut href dengan nilai yang mengandung kata .com
[atribut~="nilai"]	\$('[name~="image"]')	Seleksi semua elemen dengan atribut name yang mengandung nilai image berdiri sendiri (terpisah dengan spasi). Ini berarti bahwa selektor ini akan menyeleksi elemen <input name="image noah" /> tapi tidak

		akan menyeleksi elemen <code><input name="imagenoah" /></code> atau <code><input name="images" /></code>
<code>[atribut\$="nilai"]</code>	<code>\$('[href\$=".jpg"]')</code>	Seleksi semua elemen yang memiliki atribut href dengan nilai akhir berupa kata .jpg
<code>[atribut^="nilai"]</code>	<code>\$('img[src^="https://"]')</code>	Seleksi semua elemen yang memiliki atribut src dengan nilai awal berupa kata https://
<code>[atribut!="nilai"]</code>	<code>\$('input[type!="text"]')</code>	Seleksi semua elemen <code><input></code> yang bukan bertipe text

Tabel 3.3 Filterisasi, Keadaan & Pengurutan

Selektor	Contoh	Keterangan
<code>:first</code>	<code>\$('img:first')</code>	Seleksi semua elemen <code></code> yang pertama
<code>:last</code>	<code>\$('img:last')</code>	Seleksi semua elemen <code></code> yang terakhir
<code>:odd</code>	<code>\$('li:odd')</code>	Seleksi semua elemen <code></code> pada urutan ganjil (berarti urutan 1, 3, 5, ...)

:even	\$(li:even')	Seleksi semua elemen pada urutan genap (ini berarti urutan 2, 4, 6, ...)
:eq(n)	\$(li:eq(3))	Seleksi semua elemen ke empat (<i>start index</i> dimulai dari 0, nilai harus berupa angka positif)
:nth-child(n)	\$(li:nth-child(2))	Seleksi semua elemen ke dua (<i>start index</i> dimulai dari 1)
:only-child	\$(div img:only-child')	<p>Seleksi elemen yang merupakan anak elemen satu-satunya di dalam elemen induk. Ini berarti bahwa selektor ini akan menyeleksi elemen gambar yang berada pada kondisi seperti ini:</p> <pre><div> </div></pre> <p>namun tidak akan menyeleksi elemen gambar yang berada pada kon-</p>

		<p>disi ini:</p> <pre><div> </div></pre>
<pre>:parent</pre>	<pre>\$('td:parent')</pre>	<p>Seleksi elemen <code><td></code> yang merupakan elemen induk dari sesuatu, termasuk teks. Ini berarti bahwa selektor ini akan menyeleksi elemen <code><td></code> dengan keadaan seperti ini:</p> <pre><tr> <td>abc</td> <td></td> </tr></pre> <p>namun tidak akan menyeleksi elemen <code><td></code> dalam keadaan ini:</p> <pre><tr></pre>

		<pre><td></td> <td></td> </tr></pre>
:gt(n)	\$(li:gt(4))	Seleksi semua elemen pada urutan lebih besar dari 4 (5, 6, 7, ...)
:lt(n)	\$(li:lt(4))	Seleksi semua elemen pada urutan lebih kecil dari 4 (3, 2, 1)
:not()	\$(input:not(:last))	Seleksi semua elemen <input> selain pada urutan terakhir
:empty	\$(div:empty)	Seleksi semua elemen <div> yang kosong/tidak mempunyai anak elemen
:contains(teks)	\$(div:contains("@"))	Seleksi semua elemen yang mengandung teks "@" di dalamnya
:animated	\$(img:animated)	Seleksi semua elemen yang teranimasi (seleksi semua elemen yang sedang bergerak/bekerja oleh animasi JQuery)
:hidden	\$(img:hidden)	Seleksi semua gambar yang tidak tampak (dalam hal ini adalah elemen

		yang telah dikenai deklarasi <code>display:none</code> atau efek-efek JQuery penghilangan elemen seperti <code>.hide()</code> , <code>.slideUp()</code> dan <code>.fadeOut()</code>
<code>:visible</code>	<code>\$('img:visible')</code>	Seleksi semua elemen <code></code> yang tampak.
<code>:has(selector)</code>	<code>\$('li:has(ul)')</code>	Seleksi semua elemen <code></code> yang memiliki anak <code></code>

Tabel 3.4 Elemen-Elemen Formulir

Selektor	Contoh	Keterangan
<code>:header</code>	<code>\$(':header')</code>	Seleksi semua elemen <i>heading</i> (dalam hal ini <code><h1></code> , <code><h2></code> , <code><h3></code> , ...)
<code>:input</code>	<code>\$(':input')</code>	Seleksi semua elemen formulir (dalam hal ini adalah elemen <code><input></code> , <code><textarea></code> , <code><select></code> , dan <code><button></code>)
<code>:button</code>	<code>\$(':button')</code>	Seleksi semua elemen <code><input></code> bertipe <code>button</code> atau elemen <code><button></code>
<code>:submit</code>	<code>\$(':submit')</code>	Seleksi semua elemen bertipe <code>submit</code>

:reset	\$(':reset')	Seleksi semua elemen bertipe reset
:text	\$(':text')	Seleksi semua elemen bertipe text
:radio	\$(':radio')	Seleksi semua elemen bertipe radio
:checkbox	\$(':checkbox')	Seleksi semua elemen bertipe checkbox
:image	\$(':image')	Seleksi semua elemen bertipe image
:file	\$(':file')	Seleksi semua elemen bertipe file
:password	\$(':password')	Seleksi semua elemen bertipe password
:selected	\$('option:selected')	Seleksi semua elemen <option> yang diseleksi
:checked	\$(':checkbox:checked')	Seleksi semua elemen <input type="checkbox"> yang dicek/centang
:disabled	\$(':disabled')	Seleksi semua elemen <i>disabled</i> (misal: <input disabled="disabled" />)
:enabled	\$(':enabled')	Lawan dari selektor :disabled
:focus	\$('input:text:focus')	Seleksi semua elemen bertipe text yang sedang terfokus

Tabel 3.5 Selektor Berganda

Selektor	Contoh	Keterangan
's1, s2, s3'	\$(div.item, #container, img')	Seleksi semua elemen <div class="item">, <div id="container"> dan elemen
's1 s2'	\$(#container img')	Seleksi semua elemen yang berada di dalam elemen <div id="container">
's2', 's1'	\$(img', '#container')	Sama dengan selektor \$(#container img'), yaitu akan menyeleksi semua elemen yang berada di dalam elemen <div id="container">. Hanya saja, mungkin dari segi performa agak berbeda.
[atribut1][atribut2]	\$(input[name="email"][rel="upload"]')	Seleksi semua elemen <input> yang memiliki atribut name="email" dan rel="upload"

Selanjutnya proses menampilkan tabir saat sebuah link gambar diklik dan menganimasikan gambar dengan efek [.fadeIn\(\)](#) saat gambar dimuat. Berikut kode penulisannya di CSS:

```
#overlay {  
    position:absolute;  
    top:0px;  
    right:0px;  
    bottom:0px;  
    left:0px;  
    z-index:998;  
    background-color:black;  
    opacity:0.8;  
    display:none;  
}  
img#large {  
    position:absolute;  
    top:30px;  
    left:30px;  
    z-index:999;  
    display:none;  
}
```

Serta berikut penulisan di JQuery nya:

```
// Sisipkan gambar dan overlay  
// dengan JQuery .append() untuk meminimalisasi markup
```

```
$('#body').append('<div id="overlay"></div><img id="large" />');  
  
var validURL = $('a[href$=".jpg"], a[href$=".png"], a[href$=".gif"],  
a[href$=".bmp"]');  
  
// Saat sebuah tautan gambar diklik  
validURL.click(function() {  
  
// Tampilkan overlay  
$('#overlay').show();  
  
// Set nilai atribut src pada gambar besar menjadi nilai atribut href dari tautan yang diklik  
$('#img#large').attr('src', this.href);  
  
return false;  
});  
  
// Tampilkan gambar besar dengan efek .fadeIn() saat telah termuat  
$('#img#large').load(function() {  
$(this).fadeIn();  
});  
  
// Hilangkan gambar dan overlay saat overlay diklik  
$('#overlay').click(function() {  
$(this).hide();  
$('#img#large').removeAttr('src').fadeOut();  
});
```


Berikutnya proses penyisipan judul dan deskripsi gambar pada attribut TITLE dalam tautan pada umumnya :

```

<a href="gambar1.jpg" title="Judul Gambar 1">
  
</a>

<a href="gambar2.jpg" title="Judul Gambar 2">
  
</a>

<a href="gambar3.jpg" title="Judul Gambar 3">
  
</a>

```

Pada website “Keyna Galeri” penyisipan attribute title dengan membuat elemen baru sebagai container judul gambar, misalnya imgCap di JQuery:

```

// Sisipkan gambar, ruang judul dan overlay
// dengan JQuery .append() untuk meminimalisasi markup
$('body').append('<div id="overlay"></div><div id="imgCap"></div><img
id="large" />');

var validURL = $('a[href$=".jpg"]', a[href$=".png"], a[href$=".gif"],
a[href$=".bmp"]');

// Saat sebuah tautan gambar diklik...
validURL.click(function() {

```

```

// Tampilkan overlay
$( '#overlay' ).show();

// Set nilai atribut src pada gambar besar menjadi nilai atribut href dari tautan yang diklik
$( 'img#large' ).attr( 'src', this.href );

// Sisipkan nilai atribut title pada elemen #imgCap dengan .html()
$( '#imgCap' ).html( this.title );

return false;
});

// Tampilkan gambar besar dengan efek .fadeIn() saat telah termuat
$( 'img#large' ).load( function () {
$( this ).fadeIn();
});

// Hilangkan gambar dan overlay saat overlay diklik
$( '#overlay' ).click( function () {
$( this ).hide();
$( 'img#large' ).removeAttr( 'src' ).fadeOut();
});

```

Sampai disini proses pembuatan galeri gambar dengan teknik overlay Lightbox JQuery, selanjutnya merupakan proses perbaikan seperti memposisikan gambar, mengatur ukuran tabir pada saat overlay dengan mengkombinasikan CSS beserta JQuery .CSS untuk mengaturnya. Berikut pengaturannya:

Pada CSS

```
/* Lightbox */
```

```
#imgOverlay {  
    position:absolute;  
    top:0px;  
    right:0px;  
    bottom:0px;  
    left:0px;  
    z-index:9997;  
    background-color:black;  
    opacity:0.8;  
    display:none;  
}
```

```
#closeLightbox {  
    display:block;  
    font:bold 14px Arial,Sans-Serif;  
    color:white;  
    text-decoration:none;  
    position:absolute;  
    top:20px;  
    right:26px;  
    z-index:9999;  
    display:none;  
}
```

```
#imgWrapper {  
    position:absolute;  
    top:70px;  
    right:0px;
```

```
left:0px;
text-align:center;
z-index:9999;
}
img#imgShow {
margin:0px auto 70px;
display:none;
}
img#lightboxLoader {
display:block;
margin:30px auto 0px;
}
#imgCap {
position:absolute;
top:10px;
left:15px;
color:white;
z-index:9999;
}
```

Pada JQuery

```
$('body')
```

```
// Set scroll bar sumbu-Y menjadi scroll untuk menghindari efek loncatan yang jelek
```

```
.css('overflow-y', 'scroll')
```

```

// Sisipkan semua elemen lightbox yang diperlukan

.append('<div id="imgOverlay"></div><div id="imgWrapper"></div><a href="#"
id="closeLightbox" title="Tutup!"></a>');

// Set lebar overlay selebar jendela, dan tinggi overlay setinggi halaman
$('#imgOverlay').css({
    'width':$(window).width(),
    'height':$(document).height()
});

// Saat tautan yang mengandung thumbnail diklik...
$('a:has(img)').click(function() {

// Tampilkan overlay, pembungkus gambar dan tombol tutup
$('#imgOverlay, #imgWrapper, #closeLightbox').show();

// Sisipkan gambar pada elemen #imgWrapper dengan nilai src berupa nilai href dari tautan
yang diklik
// dan set nilai atribut alt gambar dari atribut title tautan yang diklik
$('#imgWrapper').append('');

// Sisipkan kontainer judul gambar dengan teks "Loading..." sebagai awalan
$('body').append('<div id="imgCap">Loading...</div>');

// Saat gambar besar telah selesai termuat...
$('#img#imgShow').load(function() {

```

```
// tampilkan dengan efek .fadeIn()

$(this).fadeIn(600);

// Ubah teks "Loading..." menjadi judul gambar (nilai atribut alt pada gambar besar)

$('#imgCap').html(this.alt);

});

return false;

});

// Saat tombol tutup diklik...

$('#a#closeLightbox').click(function() {

// Hilangkan diri sendiri

$(this).hide();

// Singkirkan gambar dan kontainer judul sisipan

$('#img#imgShow, #imgCap').remove();

// Hilangkan overlay dan pembungkus gambar dengan efek .fadeOut()

$('#imgOverlay, #imgWrapper').fadeOut();

return false;

});
```