4. IMPLEMENTASI SISTEM

4.1 Fungsi Login

Fungsi *login* berfungsi bagi *user* yang ingin melakukan *login* ke halaman *administrator*.

```
<?php
if(isset($_POST["login"]))
{
  $username = $_POST["username"];
  $password = $_POST["password"];
        $query = "select * from admin where username='$username' and
        password='$password'";
  $result = mysql_query($query);
  if($row = mysql_fetch_array($result))
  {
    $_SESSION["username"] = $row["username"];
    $_SESSION["name"] = $row["name"];
    ?><script>alert('Welcome, <?php echo $row["name"]; ?>');
</script>
<?php
  ?><script>document.location.href='collection.php';</script><?php
  }
  else
  {
        ?><script>alert('Wrong Username or Password');</script><?php
  }
}
?>
```

Segmen 4.1 Fungsi *Administrator Login*

Pada fungsi *login*, sistem akan melakukan pengecekan terhadap *input user* pada form *username* dan *password*. Apabila *username* dan *password* yang di-*input* oleh *user* benar dan sesuai dengan *database*, maka akan muncul pesan selamat datang dan halaman akan berpindah menuju collection.php. Tetapi jika *username*

atau *password* tidak benar, maka akan muncul pesan yang menyatakan bahwa *username* atau *password* salah.

4.2 Fungsi *Check Login*

Apabila *user* mencoba mengakses halaman *administrator* tanpa melakukan *login*, atau sesi *login*-nya telah habis, maka akan muncul *error*, dan sistem akan membuka halaman *login administrator*, untuk lebih jelasnya dapat dilihat pada Segment 4.2

```
<?php
  require_once("../db-connect.php");
  if(!isset($_SESSION["username"]))
  {
    ?><script>alert('Please Login First!');</script><?php
    ?><script>document.location.href='index.php';</script><?php
    die(0);
  }
?>
```

Segmen 4.2 Fungsi *Check Login*

4.3 Fungsi *Save*

Fungsi *save* adalah fungsi yang digunakan ketika *administrator* ingin menyimpan data seperti misalnya *collection*, atau *category*. Sebelum tersimpan, sistem akan memastikan apakah nama yang digunakan sudah ada di dalam *database* atau belum. Bila nama sudah digunakan, maka *administrator* harus menggunakan nama lain untuk dapat menyimpan data yang baru. Untuk lebih jelasnya dapat dilihat pada Segmen 4.3, dengan contoh melakukan penyimpanan saat *add new collection*.

```
if(isset($_POST["save"]))
{
  $error = false;
  $collection_name = $_POST["collection_name"];
  $year = $_POST["year"];
  $location = $_POST["location"];
  $description = $_POST["description"];
  $dataIdCategory = $_POST["id_category"];
```

```
$query = "select * from collection where collection_name='$collection_name'
and id_collection <> '$id_collection'";
$result = mysql_query($query);
  if(mysql_num_rows($result) > 0)
  {
    $error = true
    ?><script>alert('Collection name already exists');</script><?php
  }
  if($error == false)
  {
                $query = "insert into collection
                (collection_name,year,location,description) values
                ('$collection_name','$year','$location','$description')";
    mysql_query($query);
    $id_collection = mysql_insert_id();
    for($i=0; $i<count($dataIdCategory); $i++)
    {
      $id_category = $dataIdCategory[$i];
                        $query = "insert into collection_category
                (id_category,id_collection) values ('$id_category','$id_collection')";
      mysql_query($query);
    }
    ?><script>alert('Save success');</script><?php
    ?><script>document.location.href='collection.php';</script>
                <?php
  }
}
```

Segmen 4.3 Fungsi *Save*

Dalam fungsi *save*, ada satu variabel yang bernama *error* untuk menyimpan tipe *boolean*. Secara default nilai *error* adalah *false*. Kemudian sistem akan mengambil data-data yang di-*inputkan* oleh *administrator*, seperti *collection_name*, *year*, dan data lainnya. Apabila *collection_name* yang di-*inputkan* telah ada dalam *database*, maka variabel *error* akan diubah menjadi *true*, lalu muncul peringatan yang memberitahu bahwa *collection name* sudah digunakan. Apabila nilai dari variabel *error* tidak berubah, atau *false*, maka sistem akan langsung men-*query*

data-data ke dalam *database* dan kemudian akan muncul pernyataan bahwa *save* berhasil.

4.4 Fungsi *Collection Category*

Fungsi ini adalah fungsi yang digunakan saat *administrator* ingin menambahkan kategori pada sebuah koleksi. Karena sebuah koleksi bisa memiliki lebih dari satu kategori, dan tidak boleh tidak memiliki kategori, maka sistem membuat beberapa pengecekan. Untuk lebih jelas dapat dilihat pada Segmen 4.4 di bawah.

```
<script>
$(function(){
  $("#table-data").dataTable();
  var jumlah_data = <?php echo count($dataIdCategory); ?>;
  addCategory = function(){
    if($("#id_category").val() == "")
    {
      alert('No category selected');
      return false;
    }
    for(i=0; i<jumlah_data; i++)
    {
    if($("#baris_"+i.toString()).length > 0)
    {
                    if($("#id_category_"+i.toString()).val() == $("#id_category").val())
      {
        alert('Category already added');
        return false;
      }
    }
    }
  id_category = $("#id_category").val();
  category_name = $("#id_category").find('option:selected').attr("data-name");
  tambah = '<tr id="baris_'+jumlah_data.toString()+'">';
```

```
        tambah += '<td><input type="hidden" readonly
        id="id_category_'+jumlah_data.toString()+'" name="id_category[]"
        value="'+id_category+'" >';

        tambah += '<input type="hidden" readonly
        id="category_name_'+jumlah_data.toString()+'" name="category_name[]"
        value="'+category_name+'" >'+category_name+'</td>';

        tambah += '<td><button class="btn btn-danger"
        onclick="deleteCategory('+jumlah_data.toString()+')" >X</button></td>';
  tambah += '</tr>';
  $("#table-detail").append(tambah);      jumlah_data++;
  }
  deleteCategory = function(i){
    $("#baris_"+i.toString()).remove();
    }
  cekSave = function(){
    for(i=0; i<jumlah_data; i++)
    {
      if($("#baris_"+i.toString()).length > 0)
      {
        return true;
      }
    }
    alert('Category cannot be empty');
    return false;
  }
})
</script>
```

Segmen 4.4 Fungsi *Collection Category*

Pada fungsi ini akan terdapat satu variabel yang berfungsi untuk menampung jumlah data yang akan dimasukkan. Apabila *category* kosong, maka akan keluar *error*, karena *collection* setidaknya harus memiliki satu *category*. Kemudian akan dilakukan pengecekan, apakah *category* yang ingin ditambahkan telah ditambahkan sebelumnya. Kalau *category* sudah ditambahkan, maka akan muncul *error* yang memberi tahu bahwa *category* sudah ditambahkan. Apabila

semua pengecekan tidak menemukan *error*, maka *category* akan ditambahkan ke dalam tabel yang telah disediakan.

4.5 Fungsi *Delete*

Fungsi *delete* digunakan saat *administrator* ingin menghapus data, seperti *collection* atau *category*. Fungsi lebih jelasnya dapat dilihat pada Segmen 4.5.

```
$id_collection = "";
$collection_name = "";
if(isset($_GET["id_collection"]))
{
  $id_collection = $_GET["id_collection"];
  $query = "select * from collection where id_collection = '$id_collection'";
  $result = mysql_query($query);
  if($row = mysql_fetch_array($result))
  {
  $collection_name = $row["collection_name"];
  $query = "delete from collection where id_collection='$id_collection' ";
  mysql_query($query);
  if(mysql_error() == "")
  {
    ?><script>alert('Delete success');</script><?php
    ?><script>document.location.href='collection.php';</script><?php
  }
  else
  {
    ?><script>alert('Data cannot be deleted');</script><?php
    ?><script>document.location.href='collection.php';</script><?php
  }
  }
          else
          {
    ?><script>alert('Data not found');</script><?php
    ?><script>document.location.href='collection.php';</script><?php
    die(0);
          }
}
```

```
else
  {
    ?><script>alert('Data not found');</script><?php
    ?><script>document.location.href='collection.php';</script><?php
    die(0);
  }
?>
```

Segmen 4.5 Fungsi *Delete*

4.6 Fungsi *Image Collection*

Fungsi ini akan menampilkan semua *image* pada *database* sesuai dengan id *collection* yang dipilih. Data-data *image collection* akan ditampilkan dalam tabel. Fungsi lengkapnya dapat dilihat pada Segment 4.6.

```
<thead>
<tr>
  <th>Image ID</th>
  <th>Image Name</th>
  <th>Image Preview</th>
  <th></th>
</tr>
</thead>
<tbody>
<?php
$query = "select * ".
        "from image_collection ic ".
        "inner join collection c on (ic.id_collection = c.id_collection) ".
            "WHERE c.id_collection=".$_GET["id_collection"];
  $result = mysql_query($query);
  while($row = mysql_fetch_array($result))
  {
?>
<tr>
<td><?php echo $row["id_image"]; ?></td>
```

```
<td><?php echo $row["image_name"]; ?></td>
<td>
<?php
If(file_exists("../uploadedimage/".$row["image_name"]) && $row["image_name"]!="")
{
 ?>
<img style="max-width:100px;max-height:100px" src="../uploadedimage/<?php echo $row["image_name"]; ?>" />
<?php
}
?>
</td>
<td>
<a href="image_collection_delete.php?id_image=<?php echo $row["id_image"]; ?>" class="btn btn-danger" onclick="return confirm('Delete this data ?')">Delete</a>
</td>
</tr>
<?php
}
?>
</tbody>
```

Segmen 4.6 Fungsi *Image Collection*

4.7 Fungsi *Image Upload*

Fungsi ini berfungsi mengambil gambar yang ingin di-*upload*, dan gambar akan di-*copy* ke *folder* atau lokasi yang telah ditentukan sistem sehingga bisa diakses oleh *database*. Sistem akan secara otomatis memberi ekstensi pada *file*, yaitu jpeg, tanpa harus dinamai oleh *administrator*.

```
$warning="";
$id=$_GET["id_collection"];
If ($_POST)
{
$lFileName="";
```

```
$nama=mysql_real_escape_string($_POST["nama"]);
if (isset($_FILES["gambar"]))
{
$name = $_FILES["gambar"]["name"];          $ext = end((explode(".", $name))); # extra () to prevent notice
$fname=$nama;          $set="../uploaded/image/".$fname.".".$ext;
$lFileName=$fname.".".$ext;
$ttl=0;
if (file_exists($set))
{
        $warning="Name already used";
}
else {
        move_uploaded_file($_FILES["gambar"]["tmp_name"],$set);
        if ($ext=="")
        {
                $lFileName="";
        }
}
$ttl++;
}
if ($warning=="")
{
        $q="INSERT INTO image_collection (id_collection,image_name) VALUES ('$id','$lFileName') ";
        mysql_query($q);
  header("location:image_collection.php?id_collection=".$_GET["id_collection"]);
}
}
?>
```

Segmen 4.7 Fungsi *Image Upload*

Sistem mampu mengubah nama *image* yang di-*upload* sesuai dengan apa yang *administrator* isi pada kolom nama. Apabila nama yang di-*inputkan* sudah digunakan, maka akan muncul *error* dan meminta *administrator* untuk menggunakan nama lain.

4.8 Fungsi Play *Video*

Dalam memutar file *video*, *video* akan di putar dalam modal. Sistem juga menggunakan *javascript* bernama videosub.js agar dapat memutar *video* dan *subtitle* secara bersamaan. Fungsi lengkap dapat dilihat pada Segment 4.8.

```
<script src="../js/videosub.js"></script>
<script>
$(function(){
        $("#table-data").dataTable();
        playVideo = function(id,video,subtitle){
        $("#myModalLabel").html(video);
document.getElementById("videoPlayerSrc").src = "../uploadedvideo/"+video;
document.getElementById("videoPlayerSub").src = "../uploadedvideo/"+subtitle;
document.getElementById("videoPlayer").load();
document.getElementById("videoPlayer").play();
  loadVideoSub(window);
  $("#myModal").modal("show");
}
closeVideo = function(){
document.getElementById("videoPlayer").pause();
document.getElementById("videoPlayerSrc").src = null;
}
})
</script>
```

Segmen 4.8 Fungsi *Play Video*

4.9 Fungsi *Video Upload*

Untuk meng*upload video*, *administrator* memerlukan file berkestensi mp4, dan file subtitle berekstensi srt bila ada. Metode untuk meng-*upload video* sama dengan meng-*upload image*, hanya saja ada tambahan *subtitle*. File *video* dan *subtitle* akan di-*copy* ke *folder* atau lokasi yang bisa diakses oleh *database* dan dinamai sesuai keinginan *administrator*. Untuk lebih jelasnya dapat dilihat pada Segment 4.9.

```
$warning="";
$id=$_GET["id_collection"];
if ($_POST)
{
  $lFileName="";
  $nama=mysql_real_escape_string($_POST["nama"]);
  if (isset($_FILES["video"]))
  {
    if($_FILES["video"]["tmp_name"] != "")
    {
      $name = $_FILES["video"]["name"];
      $name_srt="";
      if (isset($_FILES["subtitle"]))
      {
       $name_srt = $_FILES["subtitle"]["name"];
      }
      $ext = end((explode(".", $name))); # extra () to prevent notice
                                $ext2 = end((explode(".", $name_srt))); # extra () to prevent
                               notice
     if(strtolower($ext) != "mp4")
     {
       $warning="Video's Extention Must Be Mp4";
     }
     else if($name_srt!="" && strtolower($ext2) != "srt")
     {
       $warning="Subtitle's Extention Must Be SRT";
     }
     else
     {
       $fname=$nama;
       $set="uploadedvideo/".$fname.".".$ext;
       $set_srt="";
       if ($name_srt!="")
       {
        $set_srt="uploadedvideo/".$fname.".".$ext2;
```

```
}
        $lFileName=$fname.".".$ext;
        $lSubName="";
        if ($name_srt!="")
        {
          $lSubName=$fname.".".$ext2;
        }
        $ttl=0;
        if (file_exists($set))
        {
          $warning="Name Already Used";
        }
        else {
    move_uploaded_file($_FILES["video"]["tmp_name"],"../".$set);
           if ($name_srt!="")
           { move_uploaded_file($_FILES["subtitle"]["tmp_name"],"../".$set_srt);
           }                    if ($ext=="")
         {
            $lFileName="";
         }
       }
       $ttl++;
      }
      }
      else
      {
        $warning="Video or Subtitle Cannot Be Empty";
      }
    }
    if ($warning=="")
    {
$q="INSERT INTO video_collection (id_collection,video_name,subtitle_name)
VALUES ('$id','$lFileName','$lSubName') ";
```

```
        mysql_query($q);
  header("location:video_collection.php?id_collection=".$_GET["id_collection
"]);
 }
```

Segmen 4.9 Fungsi *Video Upload*

4.10 Fungsi *Edit Camera Position*

Fungsi *edit camera position* adalah fungsi yang digunakan saat *administrator* ingin mengubah posisi kamera pada 3d *object view*. Fungsi ini juga akan dipanggil ketika *administrator* pertama kali meng-*upload object* baru, sebelum akhirnya posisi kamera *object* akan disimpan ke dalam *database*. Variabel utama pada fungsi ini adalah camerax, cameray, dan cameraz yang akan menjadi posisi kamera terhadap 3d object. *Administrator* tidak bisa meng-*inputkan* nilai terhadap posisi kamera, tetapi *administrator* harus memanipulasi kamera dengan menggeser, *zoom in* ataupun *zoom out* dengan menggunakan *mouse*.

```
$warning="";
$id_object=$_GET["id_object"];
$query = "select * ".
   "from object_collection ic ".
   "inner join collection c on (ic.id_collection = c.id_collection) ".
   "WHERE ic.id_object=$id_object";
$result = mysql_query($query);
if($row = mysql_fetch_array($result))
{
  $object_name = $row["object_name"];
  $mtl_name = $row["mtl_name"];
  $camerax = $row["camerax"];
  $cameray = $row["cameray"];
  $cameraz = $row["cameraz"];
  $id_collection = $row["id_collection"];
}
else
{
  ?><script>alert('Object not found');</script><?php
```

```
  ?><script>document.location.href='collection.php';</script><?php
  die(0);
}
if(isset($_POST["save"]))
{
  $camerax = $_POST["camerax"];
  $cameray = $_POST["cameray"];
  $cameraz = $_POST["cameraz"];
  $query = "update object_collection set camerax='$camerax',cameray='$cameray',cameraz='$cameraz' where id_object='$id_object'";
  mysql_query($query);
  ?><script>alert('Posisi camera berhasil disimpan');</script><?php
  ?><script>document.location.href='object_collection.php?id_collection=<?php echo $id_collection; ?>';</script><?php
  die(0);
}
```

Segmen 4.10 Fungsi *Edit Camera Position*

4.11 Fungsi *Related Collection*

Fungsi ini akan menampilkan semua koleksi yang berhubungan dengan koleksi yang sedang dibuka oleh *user*. Hal yang akan merelasikan koleksi-koleksi tersebut adalah kategori yang diambil pada tabel *collection_category*. Fungsi lengkap dapat dilihat pada Segmen 4.11.

```
<div class="col-lg-12">
        <h3 class="page-header">Related Collections</h3>
</div>
<div class="slider-nav" style="width:90%;margin-left:5%">
<?php
$idC="";
$q="SELECT * ".
"FROM collection_category ".
"WHERE id_collection='$id'";
```

```
{
          if ($idC=="")
          {
    $idC=$row["id_category"];
  }
  else {
  $idC=$idC.",".$row["id_category"];
  }
}
$arrCetak=array();
$q="SELECT c.*,ic.* FROM ".
"collection c ".
"INNER JOIN collection_category c2 ON (c.id_collection=c2.id_collection) ".
"INNER JOIN image_collection ic ON (c.id_collection=ic.id_collection) ".
"WHERE c2.id_category IN (". $idC .") AND c2.id_collection<>$id";
$res=mysql_query($q);
while ($row=mysql_fetch_assoc($res)) {
          if (!in_array($row["id_collection"],$arrCetak))
          {
  ?>
  <div>
<a href="portfolio-item.php?id_collection=<?php echo $row["id_collection"];?>">
<img class="img-responsive img-hover img-related rltd" src="uploadedimage/<?php
echo $row["image_name"];?>" alt=""></a>
<label style="width:100%;text-align:center">
<?php
echo $row["collection_name"];
?>
</label>
</div>
<?php
 array_push($arrCetak,$row["id_collection"]);
} }
?>
</div>
</div>
```

Segmen 4.11 Fungsi *Related Collection*

4.12 Fungsi Webgl

Fungsi webgl adalah sebuah script yang berfungsi untuk mengatur posisi kamera, render object, mengatur pencahayaan, mengontrol, dan memanggil file obj dan mtl untuk di tampilkan dalam canvas. Untuk lebih jelasnya dapat dilihat pada Segmen 4.12.

```
var lesson6 = {

nama_obj : '',

nama_mtl : '',

camera_x : 0,

camera_y : 0,

camera_z : 0,

scene: null,

camera: null,

renderer: null,

container: null,

controls: null,

clock: null,

stats: null,

init: function() { // Initialization

// create main scene

this.scene = new THREE.Scene();

this.scene.fog = new THREE.FogExp2(0xcce0ff, 0.0003);

var SCREEN_WIDTH = window.innerWidth,

SCREEN_HEIGHT = window.innerHeight;
```

```
// prepare camera
    var VIEW_ANGLE = 45;
  var ASPECT = SCREEN_WIDTH / SCREEN_HEIGHT;
  var NEAR = 1;
  var FAR = 2000;
    this.camera = new THREE.PerspectiveCamera( VIEW_ANGLE, ASPECT, NEAR,
FAR);
  this.scene.add(this.camera);
    this.camera.position.set(0, 100, 300);
    this.camera.lookAt(new THREE.Vector3(0,0,100));
    // prepare renderer
    this.renderer = new THREE.WebGLRenderer({ antialias:true });
    this.renderer.setSize(SCREEN_WIDTH, SCREEN_HEIGHT);
    this.renderer.setClearColor(this.scene.fog.color);
    this.renderer.shadowMapEnabled = true;
    this.renderer.shadowMapSoft = true;
    // prepare container
    this.container = document.createElement('div');
    document.body.appendChild(this.container);
    this.container.appendChild(this.renderer.domElement);
    // events
    THREEx.WindowResize(this.renderer, this.camera);
    // prepare controls (OrbitControls)
    this.controls = new THREE.OrbitControls(this.camera, this.renderer.domElement);
    this.controls.target = new THREE.Vector3(0, 0, 0);
    this.controls.maxDistance = 2000;
```

```
// prepare clock
    this.clock = new THREE.Clock();
    // prepare stats
    this.stats = new Stats();
    this.stats.domElement.style.position = 'absolute';
    this.stats.domElement.style.left = '50px';
    this.stats.domElement.style.bottom = '50px';
    this.stats.domElement.style.zIndex = 1;
    this.container.appendChild( this.stats.domElement );
    // add directional light
    var dLight = new THREE.DirectionalLight(0xffffff, 1.5);
    dLight.castShadow = true;
    dLight.position.set(0, -5000, 0);
    this.scene.add(dLight);
    var dLight2 = new THREE.DirectionalLight(0xffffff, 1.5);
    dLight2.castShadow = true;
    dLight2.position.set(0, 5000, 0);
    this.scene.add(dLight2);
    var dLight3 = new THREE.DirectionalLight(0xffffff, 1.5);
    dLight3.castShadow = true;
    dLight3.position.set(5000, 0, 0);
    this.scene.add(dLight3);
     var dLight4 = new THREE.DirectionalLight(0xffffff, 1.5);
    dLight4.castShadow = true;
    dLight4.position.set(-5000, 0, 0);
```

```
this.scene.add(dLight4);
    var dLight5 = new THREE.DirectionalLight(0xffffff, 1.5);
    dLight5.castShadow = true;
    dLight5.position.set(0, 0, -5000);
    this.scene.add(dLight5);
    var dLight6 = new THREE.DirectionalLight(0xffffff, 1.5);
    dLight6.castShadow = true;
    dLight6.position.set(0, 0, 5000);
    this.scene.add(dLight6);
   // load a model
    this.loadModel(); },
  loadModel: function() {
    // prepare loader and load the model
    var oLoader = new THREE.OBJMTLLoader();
    oLoader.load(this.nama_obj, this.nama_mtl, function(object) {
      object.position.x = 0;
      object.position.y = 0;
      object.position.z = 0;
      object.scale.set(1, 1, 1);
          lesson6.scene.add(object);
    });
  },
};
// Animate the scene
```

```
function animate() {
  requestAnimationFrame(animate);
  render();
  update();
}
// Update controls and stats
function update() {
  lesson6.controls.update(lesson6.clock.getDelta());
  lesson6.stats.update();
}
// Render the scene
function render() {
  if (lesson6.renderer) {
    lesson6.renderer.render(lesson6.scene, lesson6.camera);
  }
}
// Initialize lesson on page load
function initializeLesson(obj,mtl) {
  lesson6.nama_obj = obj;
  lesson6.nama_mtl = mtl;
  lesson6.init();
  animate();
  showCameraPosition();
   lesson6.controls.addEventListener( 'change', function(){
```

```
    showCameraPosition();

  } );

function initializeLesson2(obj,mtl,camerax,cameray,cameraz) {

  lesson6.nama_obj = obj;

  lesson6.nama_mtl = mtl;

  lesson6.init();

  lesson6.camera.position.x=camerax;

  lesson6.camera.position.y=cameray;

  lesson6.camera.position.z=cameraz;

  animate();

  showCameraPosition();

   lesson6.controls.addEventListener( 'change', function(){

   showCameraPosition();

   } );

function showCameraPosition(){

document.getElementById("camerax").value=lesson6.camera.position.x.toString
();

document.getElementById("cameray").value=lesson6.camera.position.y.toString
();

document.getElementById("cameraz").value=lesson6.camera.position.z.toString(
);

document.getElementById("camerapos").innerHTML                              =
lesson6.camera.position.x.toString() + ", " +lesson6.camera.position.y.toString()
+ ", " + lesson6.camera.position.z.toString();

function resetCameraPosition(){

   lesson6.controls.reset();

   animate();
```

Segmen 4.12 Fungsi Webgl