

## LAMPIRAN

### 1. View Keseluran Tampilan aplikasi

```
<!DOCTYPE html>
<html>
<head>
  <meta charset="utf-8" />
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>@ViewBag.Title - My ASP.NET Application</title>
  @Styles.Render("~/Content/css")
  @Scripts.Render("~/bundles/modernizr")
</head>
<body>
  <div class="navbar navbar-inverse navbar-fixed-top">
    <div class="container">
      <div class="navbar-header">
        <button type="button" class="navbar-toggle" data-toggle="collapse" data-target=".navbar-collapse">
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
          <span class="icon-bar"></span>
        </button>
        @Html.ActionLink("Repository Software", "Index", "Home", new { area = "" }, new { @class = "navbar-brand" })
      </div>
      <div class="navbar-collapse collapse">
        <ul class="nav navbar-nav">
          @if (@Session["username"] != null)
          {
            if (@Session["idRole"].Equals("1"))
            {
              <li>@Html.ActionLink("User Account Management", "AllList", "mahasiswa")</li>
            }
          }
          <li>
            <a href="#" class="dropdown-toggle" data-toggle="dropdown">Software Management</a>
            <ul class="dropdown-menu">
              <li>@Html.ActionLink("Software Management", "Index", "softwares")</li>
              <li>@Html.ActionLink("Software Category Management", "Index", "softwareCategories")</li>
              <li>@Html.ActionLink("Request Software", "Index", "requestedSoftwares")</li>
              <li>@Html.ActionLink("My Downloaded Software", "MyDownload", "downloadedSoftwares", new { id = Session["idUser"] }, null
```

## 2. View Software Manajemen

```
<h2>Software Category Management</h2>

<p>
  @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
  <tr>
    <th>
      NameCategory
    </th>
    <th>
      Description
    </th></th>
  </tr>

  @foreach (var item in Model) {
    <tr>
      <td>
        @Html.DisplayFor(modelItem => item.nameCategory)
      </td>
      <td>
        @Html.DisplayFor(modelItem => item.description)
      </td>
      <td>
        @Html.ActionLink("Edit", "Edit", new { id=item.idCategory }) |
        @Html.ActionLink("Details", "Details", new { id=item.idCategory }) |
        @Html.ActionLink("Delete", "Delete", new { id=item.idCategory })
      </td>
    </tr>
  }
</table>
```

### 3. View Create Software

```
// POST: softwares/Create
// To protect from overposting attacks, please enable the specific properties you want to bind to, for
// more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult Create([Bind(Include = "idSoftware,nameSoftware,version,size,description,logo,url,idCategory")] software software, HttpPostedFileBase image1)
{
    if (ModelState.IsValid)
    {
        if (image1 != null)
        {
            //software.logo = new byte[image1.ContentLength];
            //image1.InputStream.Read(software.logo, 0, image1.ContentLength);
            var fileName = Path.GetFileName(image1.FileName);
            var paths = Path.Combine("~/Content/Image/Logo/" + fileName);
            image1.SaveAs(Server.MapPath(paths));
            software.logo = Url.Content("~/Content/Image/Logo/" + fileName);
        }
        else
        {
            var paths = Path.Combine("~/Content/Image/Logo/defaultLogo.png");
            image1.SaveAs(Server.MapPath(paths));
            software.logo = Url.Content("~/Content/Image/Logo/defaultLogo.png");
        }

        if (path != null)
        {
            var fileName = Path.GetFileName(path.FileName);
            var paths = Path.Combine("~/Content/Software" + fileName);
            path.SaveAs(paths);
            software.url = Url.Content("~/Content/Software" + fileName);
        }
        db.software.Add(software);
        db.SaveChanges();
    }
}
```

#### 4. View Home Login

```
@{
    Layout = null;
}
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - Repository Software</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    <div class="navbar navbar-inverse navbar-fixed-top">
        <div class="container">
            <div class="navbar-header">
                <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                    <span class="icon-bar"></span>
                </button>
                @Html.ActionLink("Repository Software", "Index", "Home",
new { area = "" }, new { @class = "navbar-brand" })
            </div>
            <div class="navbar-collapse collapse">
                <ul class="nav navbar-nav"></ul>
                <ul class="nav navbar-nav navbar-right">
                    <li>@Html.ActionLink("Register", "Activation", "Home",
routeValues: null, htmlAttributes: new { id = "registerLink" })</li>
                    <li>@Html.ActionLink("Log in", "LoginPage", "Home",
routeValues: null, htmlAttributes: new { id = "loginLink" })</li>
                </ul>
            </div>
        </div>
    </div>
    <div class="container body-content">
        <div class="jumbotron">
            <center></center>
            <p class=""><center>Repository Software</center></p>
            <p class=""><center>Universitas Muhammadiyah
Yogyakarta</center></p>
        </div>
        <div class="col-md-12">
            @Html.Action("LastSoftware", "softwares")
        </div>
    </div>
    <hr />
    <footer>
        <p>&copy; @DateTime.Now.Year - Repository Software UMY</p>
    </footer>

    @Scripts.Render("~/bundles/jquery")
```

```
@Scripts.Render("~/bundles/bootstrap")
</body>
</html>
```

## 5. Hasil form register

### Registration Form

(Microsoft Student Development Network Academy Alliance)  
Universitas Muhammadiyah Yogyakarta

Name : [REDACTED]  
NIM / Dept : [REDACTED]  
Email : [REDACTED]  
Phone : [REDACTED]

Mengapa anda ingin mendapatkan MSDNAA?

- belajar pemograman

Teknologi apa yang ingin anda pelajari lebih dalam serta berikan penjelasan?

- visual studio

Saya bersungguh-sungguh untuk mempergunakan fasilitas MSDNAA ini untuk

1. meningkatkan kegiatan akademik saya.

2. mempergunakan fasilitas ini (hanya) untuk pribadi dan tidak didistribusikan ke pihak lain.

Bersama ini saya menyatakan bersedia mentaati semua peraturan yang terkait dengan program ini. Apabila saya melanggar peraturan yang ada maka saya bersedia untuk dikenakan sanksi Akademik dan memenuhi sanksi bersifat materil dari Microsoft/Universitas, yang muncul oleh adanya pelanggaran aturan tersebut.

Yogyakarta, 05/09/2017

[REDACTED]

## 6. View Software download details

```
@model SoftwareRepository.Models.software
@{
    ViewBag.Title = "SoftwareDownloadDetails";
}
<h2>SoftwareDownloadDetails</h2>

@using (Html.BeginForm("SoftwareDownloadDetails", "softwares",
    FormMethod.Post, new { @class = "form-horizontal login-container", id =
    "form" }))
{
    @Html.AntiForgeryToken()
    <div>

        <hr />
        @Html.HiddenFor(model => model.idSoftware)
        <dl class="dl-horizontal">
            <dt>
            </dt>
            <dd>
                
                @Html.HiddenFor(model => model.logo)
            </dd>
            <dt>
                Software Name
            </dt>
            <dd>
                @Html.DisplayFor(model => model.nameSoftware)
                @Html.HiddenFor(model => model.nameSoftware)
            </dd>
            <dt>
                Version
            </dt>
            <dd>
                @Html.DisplayFor(model => model.version)
                @Html.HiddenFor(model => model.version)
            </dd>
            <dt>
                @Html.DisplayNameFor(model => model.size)
            </dt>
            <dd>
                @Html.DisplayFor(model => model.size)
                @Html.HiddenFor(model => model.size) kb
            </dd>
            <dt>
                @Html.DisplayNameFor(model => model.description)
            </dt>
        </dl>
    </div>
}
```

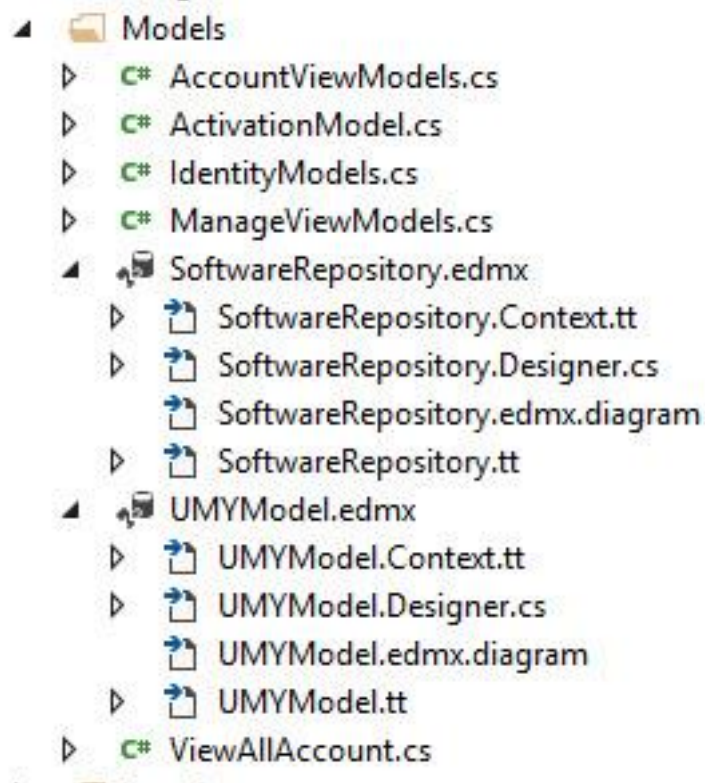
```

        </dt>
        <dd>
            @Html.DisplayFor(model => model.description)
            @Html.HiddenFor(model => model.description)
        </dd>
        <dt>
            @Html.DisplayNameFor(model =>
model.softwareCategory.nameCategory)
        </dt>
        <dd>
            @Html.DisplayFor(model =>
model.softwareCategory.nameCategory)
            @Html.HiddenFor(model =>
model.softwareCategory.nameCategory)
        </dd>
    </dl>
</div>
<p>
    <input type="submit" class="btn btn-default" value="Download" />
    @Html.ActionLink("Back to List", "SoftwareDownload")
</p>
}

```

## 7. Tampilan isi model

Terdapat dua *database* yang digunakan pada aplikasi



## 8. View tampilan *download software*

```
@model IEnumerable<SoftwareRepository.Models.downloadedSoftware>
@{
    ViewBag.Title = "Index";
}
<h2>Index</h2>
<p>
    @Html.ActionLink("Create New", "Create")
</p>
<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.software.nameSoftware)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.users.username)
        </th>
        <th></th>
    </tr>
    @foreach (var item in Model) {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.software.nameSoftware)
            </td>
            <td>
                @Html.DisplayFor(modelItem => item.users.username)
            </td>
            <td>
                @Html.ActionLink("Edit", "Edit", new { id=item.idDownload }) |
                @Html.ActionLink("Details", "Details", new { id=item.idDownload
            </td>
            @Html.ActionLink("Delete", "Delete", new { id=item.idDownload
        </td>
    </tr>
    }
</table>
```

## 9. Source code *softwarecontroller*

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using SoftwareRepository.Models;
using System.IO;
```



```

namespace SoftwareRepository.Controllers
{
    public class softwaresController : Controller
    {
        private SoftwareRepositoryEntities3 db = new
SoftwareRepositoryEntities3();

        // GET: softwares

        public ActionResult Index(string cari)
        {
            var software = db.software.Include(s => s.softwareCategory);
            if (!string.IsNullOrEmpty(cari))
            {
                return View(software.Where(s =>
s.nameSoftware.Contains(cari)));
            }
            return View(software.OrderByDescending(r => r.idSoftware));
        }

        public ActionResult PopularSoftware()
        {
            var popularSoftware = db.PopulerSoftware;
            return View(popularSoftware.ToList().OrderByDescending(p =>
p.PeopleDownloaded).Take(5));
        }

        // GET: softwares/Details/5
        public ActionResult Details(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            software software = db.software.Find(id);
            if (software == null)
            {
                return HttpNotFound();
            }
            return View(software);
        }

        // GET: softwares/Create
        public ActionResult Create()
        {
            ViewBag.idCategory = new SelectList(db.softwareCategory,
"idCategory", "nameCategory");
            return View();
        }

        // POST: softwares/Create
        // To protect from overposting attacks, please enable the specific
properties you want to bind to, for
        // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]

```

```

        [ValidateAntiForgeryToken]
        public ActionResult Create([Bind(Include =
        "idSoftware,nameSoftware,version,size,description,logo,url,idCategory")]
        software software,HttpPostedFileBase image1,HttpPostedFileBase path)
        {
            if (ModelState.IsValid)
            {
                if (image1 != null)
                {
                    //software.logo = new byte[image1.ContentLength];
                    //image1.InputStream.Read(software.logo, 0,
                    image1.ContentLength);
                    var fileName = Path.GetFileName(image1.FileName);
                    var paths = Path.Combine("~/Content/Image/Logo/" +
                    fileName);
                    image1.SaveAs(Server.MapPath(paths));
                    software.logo = Url.Content("~/Content/Image/Logo/"
                    + fileName);
                }
                else
                {
                    var paths =
                    Path.Combine("~/Content/Image/Logo/defaultLogo.png");
                    image1.SaveAs(Server.MapPath(paths));
                    software.logo =
                    Url.Content("~/Content/Image/Logo/defaultLogo.png");
                }

                if (path != null)
                {
                    var fileName = Path.GetFileName(path.FileName);
                    var paths = Path.Combine("~/Content/Software/" +
                    fileName);
                    //path.SaveAs(paths);
                    path.SaveAs(Server.MapPath(paths));
                    software.url = Url.Content("~/Content/Software/" +
                    fileName);
                }
                db.software.Add(software);
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.idCategory = new SelectList(db.softwareCategory,
            "idCategory", "nameCategory", software.idCategory);
            return View(software);
        }

        // GET: softwares/Edit/5
        public ActionResult Edit(int? id)
        {
            if (id == null)
            {
                return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
            }
            software software = db.software.Find(id);

```

```

        if (software == null)
        {
            return HttpNotFound();
        }
        ViewBag.idCategory = new SelectList(db.softwareCategory,
        "idCategory", "nameCategory", software.idCategory);
        return View(software);
    }

    // POST: softwares/Edit/5
    // To protect from overposting attacks, please enable the specific
properties you want to bind to, for
    // more details see http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Edit([Bind(Include =
        "idSoftware,nameSoftware,version,size,description,logo,url,idCategory")]
        software software, HttpPostedFileBase image1, HttpPostedFileBase path)
    {
        var db = new SoftwareRepositoryEntities3();
        if (image1 == null)
        {
            if (path == null)
            {
                //Go fetch the existing profile from the database
                var currentSoftware = db.software.FirstOrDefault(p =>
                p.idSoftware == software.idSoftware);
                //Update the database record with the values from your
model
                currentSoftware.nameSoftware = software.nameSoftware;
                currentSoftware.version = software.version;
                currentSoftware.size = software.size;
                currentSoftware.description = software.description;
                currentSoftware.idCategory = software.idCategory;
                //Commit to the database!
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            else
            {
                //Go fetch the existing profile from the database
                var currentSoftware = db.software.FirstOrDefault(p =>
                p.idSoftware == software.idSoftware);
                //Update the database record with the values from your
model
                currentSoftware.nameSoftware = software.nameSoftware;
                currentSoftware.version = software.version;
                currentSoftware.size = software.size;
                currentSoftware.description = software.description;
                currentSoftware.idCategory = software.idCategory;

                var fileName = Path.GetFileName(path.FileName);
                var paths = Path.Combine("~/\\Content\\Software" +
                fileName);
                path.SaveAs(paths);
            }
        }
    }

```

```

software.url = Url.Content("~/Content/Software" +
fileName);
currentSoftware.url = software.url;

//Commit to the database!
db.SaveChanges();
return RedirectToAction("Index");
}
}
else
{
if (path != null)
{
//Go fetch the existing profile from the database
var currentSoftware = db.software.FirstOrDefault(p =>
p.idSoftware == software.idSoftware);
//Update the database record with the values from your
model

currentSoftware.nameSoftware = software.nameSoftware;
currentSoftware.version = software.version;
currentSoftware.size = software.size;
currentSoftware.description = software.description;
currentSoftware.idCategory = software.idCategory;

var fileName = Path.GetFileName(image1.FileName);
var paths = Path.Combine("~/Content/Image/Logo/" +
fileName);
image1.SaveAs(paths);
software.logo = Url.Content("~/Content/Image/Logo/"
+ fileName);
currentSoftware.logo = software.logo;

var fileNames = Path.GetFileName(path.FileName);
var pathss = Path.Combine("~/Content/Software" +
fileName);
path.SaveAs(paths);
software.url = Url.Content("~/Content/Software" +
fileName);
currentSoftware.url = software.url;

//Commit to the database!
db.SaveChanges();
return RedirectToAction("Index");
}
else
{
var currentSoftware = db.software.FirstOrDefault(p =>
p.idSoftware == software.idSoftware);
//Update the database record with the values from your
model

currentSoftware.nameSoftware = software.nameSoftware;
currentSoftware.version = software.version;
currentSoftware.size = software.size;
currentSoftware.description = software.description;
currentSoftware.idCategory = software.idCategory;

```

```

        var fileName = Path.GetFileName(image1.FileName);
        var paths = Path.Combine("~/Content/Image/Logo/" +
fileName);
        image1.SaveAs(paths);
        software.logo = Url.Content("~/Content/Image/Logo/"
+ fileName);
        currentSoftware.logo = software.logo;

        //Commit to the database!
        db.SaveChanges();
        return RedirectToAction("Index");
    }
    db.Entry(software).State = EntityState.Modified;
    db.SaveChanges();
    return RedirectToAction("Index");
}

ViewBag.idCategory = new SelectList(db.softwareCategory,
"idCategory", "nameCategory", software.idCategory);
return View(software);
}

// GET: softwares/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
    }
    software software = db.software.Find(id);
    if (software == null)
    {
        return HttpNotFound();
    }
    return View(software);
}

// POST: softwares/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int id)
{
    software software = db.software.Find(id);
    db.software.Remove(software);
    db.SaveChanges();
    return RedirectToAction("Index");
}

protected override void Dispose(bool disposing)
{
    if (disposing)
    {
        db.Dispose();
    }
    base.Dispose(disposing);
}

public ActionResult Download(string Url, int id)

```

```

    {
        checkData(id);

        string filename = "";
        software software = db.software.Find(id);
        string fullName = Path.Combine(Url);
        filename = Path.GetFileName(fullName);

        byte[] fileBytes = GetFile(fullName);
        return File(
            fileBytes,
            System.Net.Mime.MediaTypeNames.Application.Octet, filename);
    }

    byte[] GetFile(string s)
    {
        System.IO.FileStream fs =
            System.IO.File.OpenRead(Server.MapPath(s));
        byte[] data = new byte[fs.Length];
        int br = fs.Read(data, 0, data.Length);
        if (br != fs.Length)
            throw new System.IO.IOException(s);
        return data;
    }

    public ActionResult SoftwareDownload()
    {
        var software = db.software.Include(s => s.softwareCategory);
        return View(software.ToList());
    }

    public ActionResult SoftwareDownloadDetails(int? id)
    {
        if (id == null)
        {
            return new HttpStatusCodeResult(HttpStatusCode.BadRequest);
        }
        software software = db.software.Find(id);
        if (software == null)
        {
            return HttpNotFound();
        }
        return View(software);
    }

    //get software by category
    public ActionResult SoftwareByCategory(int? category)
    {
        //var software = db.software.Where(r=> r.idCategory ==
category);
        var result = from sf in db.software
                    where sf.idCategory == category
                    select sf;
        return View(result);
    }

```

```

    }

    public ActionResult NewSoftware()
    {
        DateTime date = DateTime.Today;

        DateTime FromDate = Convert.ToDateTime(date);
        DateTime ToDate = Convert.ToDateTime(date.AddMonths(-1));

        var result = from software in db.software
                     where software.uploadDate <= FromDate
                     where software.uploadDate > ToDate
                     select software;

        return View(result.OrderByDescending(p => p.idSoftware));
    }

    public ActionResult LastSoftware()
    {
        var software = db.software.OrderByDescending(x =>
x.idSoftware).Take(5);
        return PartialView(software);
    }

    public void checkData(int idSoftware)
    {
        int idUser = Convert.ToInt32(Session["idUser"]);

        var results = from md in db.downloadedSoftware
                     where md.idUser == idUser && md.idSoftware ==
idSoftware
                     select md;

        if (results.FirstOrDefault() == null)
        {
            downloadedSoftware ds = new downloadedSoftware();
            ds.idUser = Convert.ToInt32(Session["idUser"].ToString());
            ds.idSoftware = idSoftware;
            db.downloadedSoftware.Add(ds);
            db.SaveChanges();
        }
        else
        {
        }
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult Details([Bind(Include =
"idSoftware,nameSoftware,version,size,description,logo,url,idCategory")]
software software)
    {
        var results = from urls in db.software
                     where urls.idSoftware == software.idSoftware
                     select urls;

```

```

        var result = "";

        if (results.FirstOrDefault() != null)
        {
            result = results.FirstOrDefault().url.ToString();
        }
        else { }
        var downloadSoftware = Download(result, software.idSoftware);
        return downloadSoftware;
    }

    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult SoftwareDownloadDetails([Bind(Include =
    "idSoftware,nameSoftware,version,size,description,logo,url,idCategory")]
    software software)
    {
        var results = from urls in db.software
                      where urls.idSoftware == software.idSoftware
                      select urls;
        var result = "";

        if (results.FirstOrDefault() != null)
        {
            result = results.FirstOrDefault().url.ToString();
        }
        else { }
        var downloadSoftware = Download(result, software.idSoftware);
        return downloadSoftware;
    }
}
}
}

```