NAMA : Herbeth Augustinus Napitupulu
NIM : 11321047
KELAS : D3TI02

LAPORAN PRAKTIKUM TEKNET

1. **Routeconfig.cs**

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;
using System.Web.Routing;

namespace W12S2_11321047_HerbethNapitupulu
{
    public class RouteConfig
    {
        public static void RegisterRoutes(RouteCollection routes)
        {
            routes.IgnoreRoute("{resource}.axd/{*pathInfo}");
            routes.MapRoute(
            name: "Login",
            url: "{controller}/{action}/{id}",
            defaults: new
            {
                controller = "Login",
                action = "Index",
                id = UrlParameter.Optional
            }
            );
            routes.MapRoute(
            name: "Home1",
            url: "{controller}/{action}/{id}",
            defaults: new
            {
                controller = "Home",
                action =
            "Dashboard",
                id = UrlParameter.Optional
            }
            );
            routes.MapRoute(
            name: "Home2",
            url: "{controller}/{action}/{id}",
            defaults: new
            {
                controller = "Home",
                action = "BuyForm",
                id = UrlParameter.Optional
            }
            );
        }
    }
}
```

Penjelasan :

Kode di atas merupakan bagian dari konfigurasi routing pada sebuah aplikasi web yang dibangun dengan menggunakan teknologi ASP.NET MVC. Routing adalah proses pemetaan URL yang diterima oleh server ke sebuah aksi (action) pada sebuah controller yang akan menangani permintaan tersebut.

Pada kode tersebut, terdapat sebuah kelas RouteConfig dengan sebuah method RegisterRoutes yang akan mendaftarkan routing pada aplikasi. Method ini menerima parameter RouteCollection yang akan menampung semua daftar routing.

Baris pertama pada method tersebut adalah routes.IgnoreRoute yang akan mengabaikan permintaan yang mengandung path "/{resource}.axd/{*pathInfo}".

Kemudian, terdapat tiga buah routes.MapRoute, yang masing-masing akan menambahkan konfigurasi routing dengan nama yang berbeda.

Pada routing pertama, terdapat nama "Login" dan URL pattern "{controller}/{action}/{id}". Hal ini akan membuat permintaan dengan URL seperti "http://localhost/Login/Index" akan diarahkan ke aksi "Index" pada controller "Login".

Pada routing kedua, terdapat nama "Home1" dan URL pattern yang sama seperti routing pertama. Namun, controller yang dituju adalah "Home" dan action yang dituju adalah "Dashboard".

Pada routing ketiga, terdapat nama "Home2" dan URL pattern yang sama dengan routing sebelumnya. Controller yang dituju tetap "Home", namun action yang dituju adalah "BuyForm".

Kedua routing dengan nama "Home1" dan "Home2" sebenarnya memiliki URL pattern yang sama, sehingga pada kenyataannya hanya routing yang pertama kali ditemukan oleh server yang akan dijalankan.

2. **HomeController.cs**

```
using W12S2_11321047_HerbethNapitupulu.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace W12S2_11321047_HerbethNapitupulu.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Dashboard()
        {
            Books[] array =
            {
                new Books {
                    Id = 1,
                    BookName = "Hands-on Machine Learning with Scikit-Learn,
Keras &TensorFlow",
                    BookDescription = "Through a series of recent breakthroughs,
deep learning has boosted the entire field of machine learning. Now,
evenprogrammers who know close to nothing about this technology can use simple,
efficient tools toimplement programs capable of learning from data. This
practical book shows you how.",
```

```
                    BookPrice = 200000,
                    Stock = 100,
                    BookRating = 5,
                    BookImage = "book 1.jpg",
},
new Books {
                    Id = 2,
                    BookName = "Practical Deep Learning for Cloud, Mobile &
Edge",
                    BookDescription = "Whether you're a software engineer
aspiring to enter the world of deep learning, a veteran data scientist, or a
hobbyist with asimple dream of making the next viral AI app, you might have
wondered where to begin. This step_by-step guide teaches you how to build
practical deep learning applications for the cloud,mobile, browsers, and edge
devices using a hands-on approach.",
                    BookPrice = 390000,
                    Stock = 80,
                    BookRating = 4,
                    BookImage = "book 2.png",
},
new Books
            {
                    Id = 3,
                    BookName = "Programming PyTorch for Deep Learning",
                    BookDescription = "Take the next steps toward mastering deep
learning, the machine learning method that's transforming the world around us by
thesecond.In this practical book, you'll get up to speed on key ideas using
Facebook's open sourcePyTorch framework and gain the latest skills you need to
create your very own neural networks.",
                    BookPrice = 600000,
                    Stock = 5,
                    BookRating = 3,
                    BookImage = "book 3.jpg",
},
new Books
{
                    Id = 4,
                    BookName = "Building Machine Learning Pipelines",
                    BookDescription = "Companies are spending billions on machine
learning projects, but it's money wasted if the models can't be deployed
effectively.Inthis practical guide,Hannes Hapke and Catherine Nelson walk you
through the steps of automatinga machine learning pipeline using the TensorFlow
ecosystem.You'll learn the techniques and tools that will cut deployment time
from days to minutes, so that you can focus on developing new models rather than
maintaining legacy systems.",
                    BookPrice = 600000,
                    Stock = 0,
                    BookRating = 3,
                    BookImage = "book 4.jpg",
            },
        };
            return View(array);
        }
        [HttpGet]
        public ActionResult BuyForm()
        {
            return View();
        }
        [HttpPost]
        public ActionResult
BuyForm(W12S2_11321047_HerbethNapitupulu.Models.BuyForm buyForm)
        {
            if (ModelState.IsValid)
```

```
            {
                return View("Thankyou", buyForm);
            }
            return View();
        }
    }
}
```

Penjelasan :

Class tersebut bernama HomeController dan merupakan turunan dari class Controller.

Method Dashboard digunakan untuk menampilkan daftar buku beserta informasi seperti nama buku, deskripsi, harga, stok, rating, dan gambar buku. Informasi buku disimpan dalam array Books[] array. Method ini mengembalikan sebuah ActionResult yang nantinya akan di-render dalam view.

Method BuyForm memiliki dua implementasi. Implementasi pertama berfungsi untuk menampilkan form pembelian buku, dan implementasi kedua berfungsi untuk memproses inputan pembelian yang dikirim dari form tersebut. Implementasi pertama menggunakan metode HttpGet, sementara implementasi kedua menggunakan metode HttpPost.

Jika inputan valid, method BuyForm yang menggunakan metode HttpPost akan meredirect pengguna ke halaman "Thankyou" dan menampilkan objek buyForm. Jika inputan tidak valid, maka method BuyForm yang menggunakan metode HttpPost akan kembali menampilkan form pembelian.

3. LoginController.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace W12S2_11321047_HerbethNapitupulu.Controllers
{
    public class LoginController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }
        [HttpPost]
        public ActionResult Index(string email, string password)
        {
            if (email == "admin@gmail.com" && password == "password")
            {
                return RedirectToAction("Dashboard", "Home");
            }
            else
            {
                return View();
            }
        }
    }
}
```

Penjelasan :

Class tersebut bernama LoginController dan merupakan turunan dari class Controller.

Method Index memiliki dua implementasi. Implementasi pertama berfungsi untuk menampilkan halaman login, sementara implementasi kedua berfungsi untuk memproses inputan email dan password yang dikirim dari form login yang menggunakan metode HttpPost.

Jika email dan password yang dimasukkan oleh pengguna sesuai dengan yang telah ditentukan (dalam kasus ini adalah "admin@gmail.com" dan "password"), maka method Index yang menggunakan metode HttpPost akan meredirect pengguna ke halaman dashboard di dalam HomeController dengan menggunakan RedirectToAction. Jika email atau password tidak sesuai, maka method Index yang menggunakan metode HttpPost akan kembali menampilkan halaman login.

Pada implementasi kedua, nilai email dan password diperoleh dari parameter string email dan string password yang diterima oleh method tersebut.

4. Account.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace W12S2_11321047_HerbethNapitupulu.Models
{
    public class Account
    {
        [Required(ErrorMessage = "Please enter your email")]
        [RegularExpression(".+\\@.+\\..+", ErrorMessage = "Please enter a valid
email address")]
        public string Email { get; set; }
        [Required(ErrorMessage = "Please enter your password")]
        [RegularExpression("^(?=.*[a-z])(?=.*[A-Z])(?=.*\\d)(?=.*[^\\da-zA-
Z]).{8,15}$",
        ErrorMessage = "Please enter a valid password")]
        public string Password { get; set; }
    }
}
```

Penjelasan :

Kode di atas adalah sebuah definisi kelas C# yang bernama "Account". Kelas ini memiliki dua properti publik yaitu "Email" dan "Password" yang masing-masing memiliki atribut validasi.

Atribut [Required] digunakan untuk menandakan bahwa kedua properti tersebut harus diisi dengan nilai, sehingga jika nilai yang diberikan adalah null atau string kosong ("") maka akan terjadi kesalahan validasi dengan pesan kesalahan yang tercantum dalam atribut.

Atribut [RegularExpression] digunakan untuk memvalidasi nilai dari properti "Email" dan "Password" dengan membandingkan nilai tersebut dengan pola regex yang telah ditentukan dalam atribut. Pola regex pada properti "Email" digunakan untuk memastikan bahwa nilai yang diberikan adalah alamat

email yang valid, sedangkan pola regex pada properti "Password" digunakan untuk memastikan bahwa nilai yang diberikan adalah password yang memenuhi kriteria keamanan tertentu.

Jika nilai yang diberikan tidak sesuai dengan pola regex yang telah ditentukan, maka akan terjadi kesalahan validasi dengan pesan kesalahan yang tercantum dalam atribut [RegularExpression].

5. Book.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace W12S2_11321047_HerbethNapitupulu.Models
{
    public class Books
    {
        public int Id { get; set; }
        public string BookName;
        public string GetBookName(int id)
        {
            return BookName;
        }
        public void SetBookName(string bookName)
        {
            BookName = bookName;
        }
        public string BookDescription { get; set; }
        public double BookPrice { get; set; }
        public int Stock { get; set; }
        public int BookRating { get; set; }
        public string BookImage { get; set; }
    }
}
```

Penjelasan :

Kode di atas adalah sebuah definisi kelas C# yang bernama "Books". Kelas ini memiliki beberapa properti dan metode yang digunakan untuk merepresentasikan buku dalam sebuah aplikasi.

Properti yang dimiliki oleh kelas ini antara lain adalah:

Id: sebuah bilangan bulat yang merupakan identitas unik dari sebuah buku.

BookName: sebuah string yang merupakan nama dari buku.

BookDescription: sebuah string yang berisi deskripsi singkat tentang buku.

BookPrice: sebuah bilangan pecahan yang merupakan harga dari buku.

Stock: sebuah bilangan bulat yang menunjukkan jumlah stok buku yang tersedia.

BookRating: sebuah bilangan bulat yang menunjukkan rating buku.

BookImage: sebuah string yang berisi path gambar dari buku.

Kelas ini juga memiliki dua metode yaitu GetBookName() dan SetBookName() yang digunakan untuk mengambil dan mengatur nilai dari properti BookName.

Namun, perlu diperhatikan bahwa properti BookName seharusnya memiliki akses modifier "public" pada deklarasinya, sehingga dapat diakses dari luar kelas. Dalam kode di atas, akses modifier untuk properti BookName tidak ditentukan, sehingga secara default memiliki akses modifier "private", dan tidak dapat diakses dari luar kelas.

6. BuyForm.cs

```
using System;
using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace W12S2_11321047_HerbethNapitupulu.Models
{
    public class BuyForm : Books
    {
        public int buyQuantity { get; set; }
        public double buyPrice
        {
            get
            {
                return BookPrice * buyQuantity;
            }
        }
        [Required(ErrorMessage = "Please upload your payment proof")]
        public string buyProof { get; set; }
        public bool Confirm { get; set; }
    }
}
```

Penjelasan :

Kode tersebut merupakan definisi kelas C# yang bernama "BuyForm". Kelas ini merupakan turunan dari kelas "Books", yang berarti kelas "BuyForm" memiliki semua properti yang dimiliki oleh kelas "Books".

Selain properti-properti yang diwarisi dari kelas "Books", kelas "BuyForm" juga memiliki properti tambahan berikut:

buyQuantity: sebuah bilangan bulat yang menyimpan jumlah buku yang ingin dibeli oleh pembeli.

buyPrice: sebuah bilangan pecahan yang menyimpan total harga pembelian buku berdasarkan jumlah buku yang ingin dibeli dan harga per buku yang terdapat dalam properti BookPrice. Properti ini hanya memiliki akses pembacaan (getter) saja, dan tidak memiliki akses penulisan (setter).

buyProof: sebuah string yang menyimpan path atau lokasi file dari bukti pembayaran yang diunggah oleh pembeli. Properti ini memiliki atribut validasi [Required] yang menandakan bahwa nilai dari properti ini harus diisi atau diunggah oleh pembeli.

Confirm: sebuah boolean yang menyimpan informasi apakah pembelian sudah dikonfirmasi atau belum oleh pihak penjual.

Dengan adanya properti tambahan tersebut, kita dapat membuat objek "BuyForm" yang merepresentasikan formulir pembelian buku oleh pembeli, dan juga melakukan validasi pada properti buyProof untuk memastikan bahwa pembeli sudah mengunggah bukti pembayaran sebelum melanjutkan proses pembelian.

7. BuyForm.cshtml

```
@model W12S2_11321047_HerbethNapitupulu.Models.BuyForm
@{
    ViewData["Title"] = "BuyForm";
    Layout = "~/Views/LayoutTemplate.cshtml";
}
@section Header{
    <h3>Form Pembelian Buku</h3>
    <p>Informasi Buku yang Anda Beli (Nomor Rekening Pembayaran : 123456789)</p>
}
<html>
<head>
    <title>
        @ViewData["Title"]
    </title>
</head>
<body>
    @using (Html.BeginForm("BuyForm", "Home", FormMethod.Post, new { enctype =
"multipart/form_data" }))
    {
        <p>
            Judul Buku : @Html.TextBoxFor(x =>
x.@BookName)@Html.ValidationMessageFor(x =>
x.@BookName)
        </p>
        <p>
            Jumlah Buku yang Anda Beli : @Html.TextBoxFor(x => x.buyQuantity, new
{
type = "number"
})
        </p>
        <p>
            Jumlah yang Anda Bayar : @Html.TextBoxFor(x => x.buyPrice, new
{
type = "number"
})
        </p>
        <p>
            Upload bukti pembayaran :
            @Html.TextBoxFor(x => x.buyProof, new { type = "file" })
            <img src="~/images/upload.png" alt="upload" style="width: 20px;
height: 20px;" />
            @Html.ValidationMessageFor(x => x.buyProof)
        </p>
        <p>
            Silahkan Konfirmasi Pengiriman Buku :
            @Html.DropDownListFor(x => x.Confirm, new[]{
            new SelectListItem(){Text = "Pengiriman buku dilakukan dengan kurir",
Value =
           bool.TrueString},
```

```
            new SelectListItem(){Text = "Pengiriman buku dilakukan secara COD",
Value =
           bool.TrueString}
            }, "Silahkan pilih pilihan anda")
        </p>
        <input type="submit" value="Kirim Pengajuan" />
    }
</body>
</html>
@section Footer{
    @Html.Partial("PartialViewHome")
}
```

Penjelasan :

Kode tersebut merupakan sebuah file view pada aplikasi web yang menampilkan form pembelian buku.

Pada bagian atas, terdapat definisi judul halaman dan layout yang digunakan.

Pada bagian @section Header, terdapat header dari form pembelian buku, yaitu "Form Pembelian Buku" dan informasi nomor rekening pembayaran.

Pada bagian @using (Html.BeginForm), terdapat form untuk memasukkan informasi pembelian buku. Terdapat beberapa field yang harus diisi, yaitu:

Judul Buku : field input untuk mengisi judul buku yang dibeli

Jumlah Buku yang Anda Beli : field input untuk mengisi jumlah buku yang dibeli

Jumlah yang Anda Bayar : field input untuk menampilkan jumlah yang harus dibayar

Upload bukti pembayaran : field input untuk upload file bukti pembayaran

Silahkan Konfirmasi Pengiriman Buku : dropdown untuk memilih opsi pengiriman buku, yaitu dilakukan dengan kurir atau secara COD

Pada bagian @section Footer, terdapat footer dari halaman yang memuat partial view "PartialViewHome".

Dashboard.cshtml

```
@using W12S2_11321047_HerbethNapitupulu.Models
@model Books[]
@{
    ViewBag.Title = "Dashboard";
    Layout = "~/Views/LayoutTemplate.cshtml";
}
@section Header{
    <h4>Selamat Datang di Koleksi Buku Toko Buku</h4>
}
<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1.0" />
    <title>@ViewBag.Title</title>
```

```
</head>
<body>
    <h3>Daftar Koleksi Buku di Toko Buku </h3>
    <table>
        <tr>
            <th>Book Picture</th>
            <th>Book Name</th>
            <th width="500px">Book Description</th>
            <th>Book Price</th>
            <th>Ketersediaan</th>
            <th>Book Rating</th>
            <th>BUY</th>
        </tr>
        @foreach (var item in Model)
        {
            <tr>
                <td><img src="~/Images/@item.BookImage" alt="@item.BookName"
width="100" /></td>
                <td>@item.BookName</td>
                <td width="500px">@item.BookDescription</td>
                <td>@item.BookPrice</td>
                <td>@item.Stock</td>
                <td>
                    @if (item.BookRating == 5)
                    {
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                    }
                    else if (item.BookRating == 4)
                    {
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                    }
                    else if (item.BookRating == 3)
                    {
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                        <img src="~/Images/star.jfif" alt="star" width="20" />
                    }
                </td>
                <td>
                    @Html.ActionLink("BUY", "BuyForm", new { id = item.Id })
                </td>
            </tr>
        }
    </table>
</body>
</html>
@section Footer{
    <h4>Copyright &copy; - Herbeth Napitupulu</h4>
}
```

Penjelasan :

Kode tersebut adalah sebuah file view untuk menampilkan daftar koleksi buku di sebuah toko buku. File ini menggunakan model Books[] yang berisi kumpulan data buku yang akan ditampilkan.

Bagian @section Header berisi judul halaman, sedangkan @section Footer berisi informasi hak cipta.

Kode HTML yang ada pada file view ini menampilkan daftar koleksi buku dalam sebuah tabel. Setiap baris tabel menampilkan gambar buku, nama buku, deskripsi buku, harga buku, ketersediaan stok buku, rating buku dalam bentuk bintang, dan tombol BUY yang mengarahkan pengguna ke form pembelian.

Setiap baris tabel dibuat menggunakan perulangan foreach yang mengulang data buku pada model Books[]. Untuk menampilkan gambar buku, file gambar disimpan dalam folder Images dan ditampilkan menggunakan tag img. Rating buku ditampilkan dalam bentuk gambar bintang sesuai dengan rating buku, dan tombol BUY yang mengarahkan pengguna ke form pembelian ditampilkan menggunakan Html.ActionLink.

Kode ini juga memperbolehkan pengguna untuk melihat file LayoutTemplate.cshtml untuk menampilkan layout secara umum pada halaman situs web.