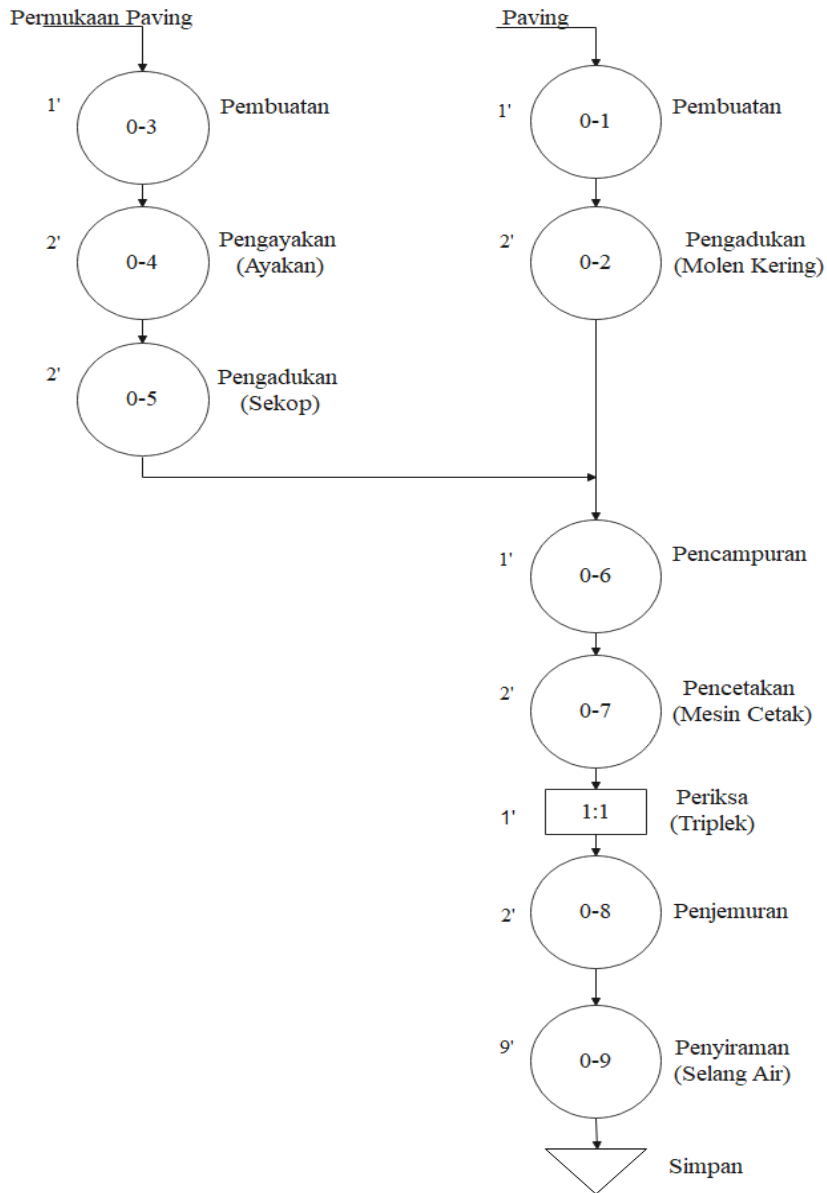
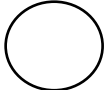

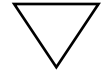


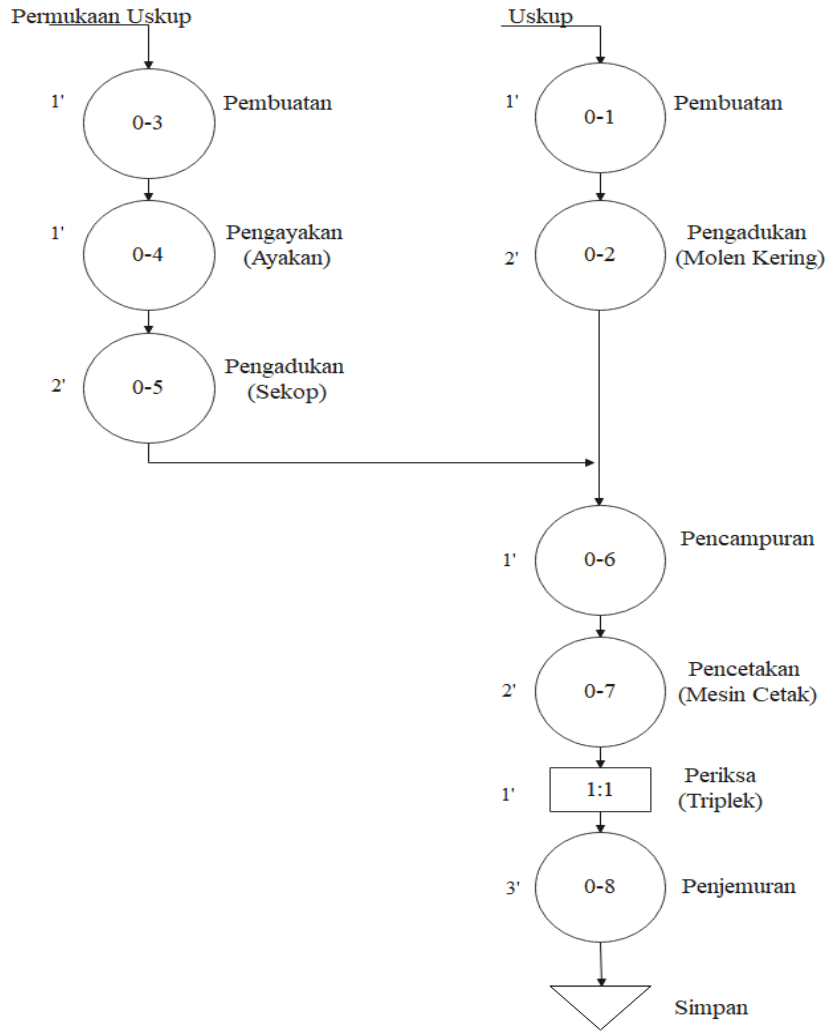
LAMPIRAN

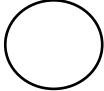


Lampiran 1 OPC Paving Block



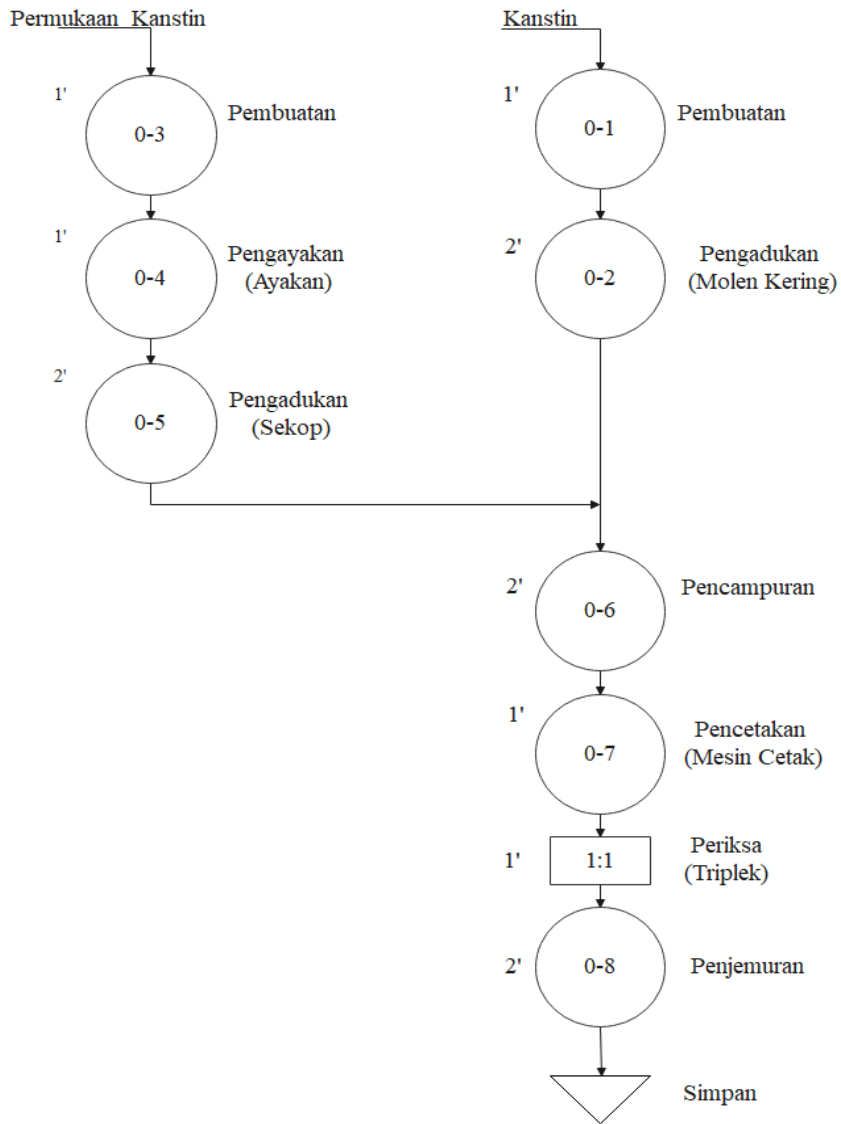
Ringkasan		
	Jumlah	Waktu
	9	24
	1	1
	1	
Total	11	25

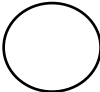

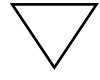
Lampiran 2 OPC Uskup



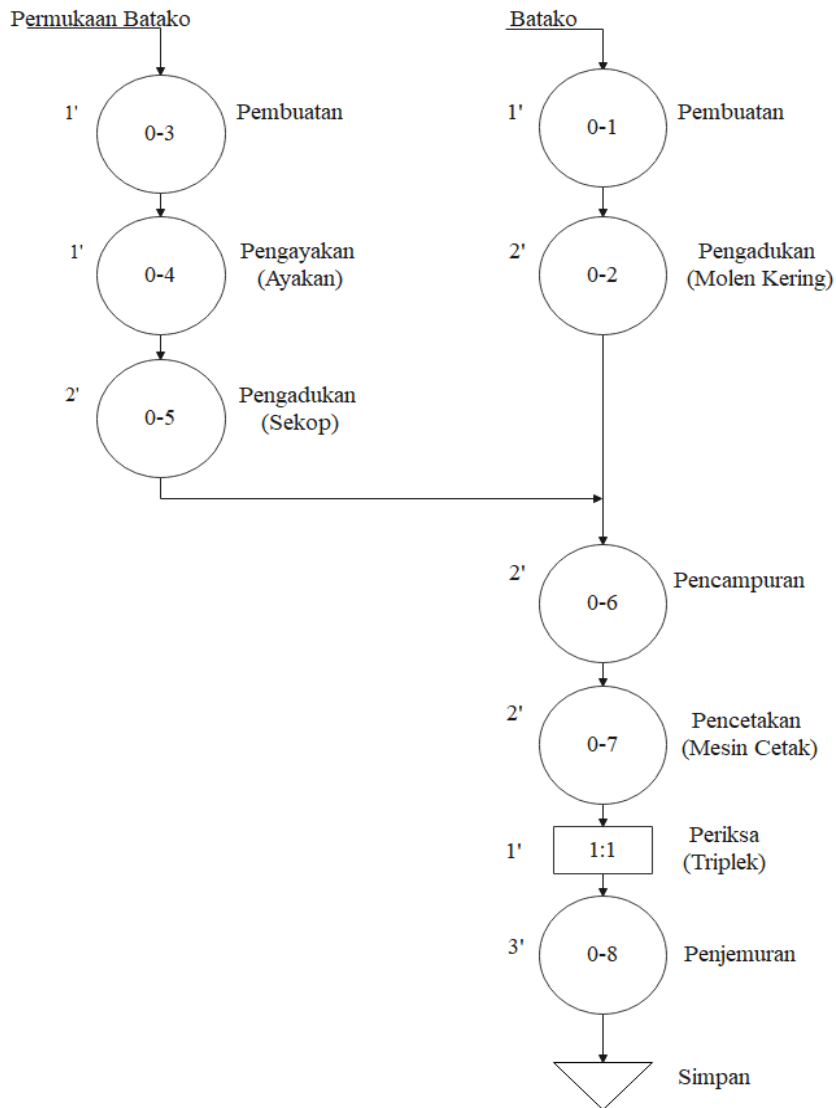
Ringkasan		
	Jumlah	Waktu
	8	14
	1	1
	1	
Total	10	15

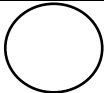


Lampiran 3 OPC Kanstin



Ringkasan		
	Jumlah	Waktu
	8	13
	1	1
	1	
Total	10	14

Lampiran 4 OPC Batako

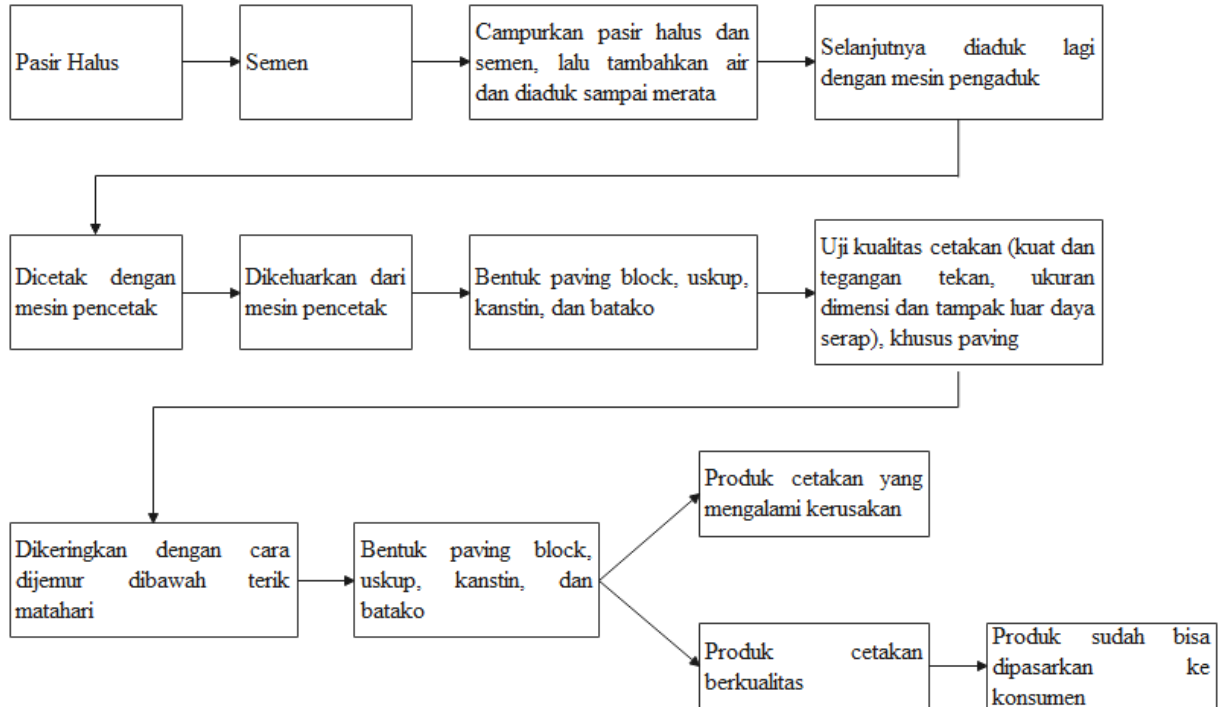


Ringkasan		
	Jumlah	Waktu
	8	15
	1	1
	1	
Total	9	16

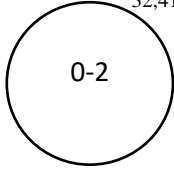
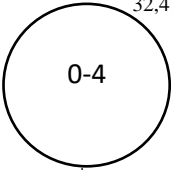
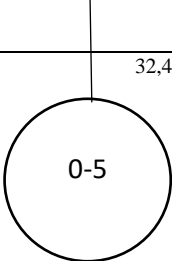
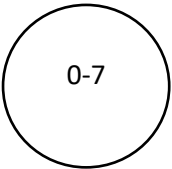
Lampiran 5 Mesin Yang di Gunakan



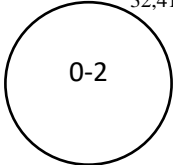
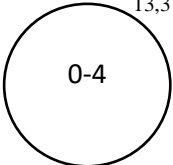
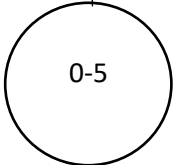
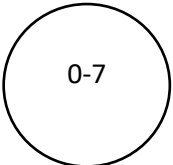
Lampiran 6 Alur Produksi



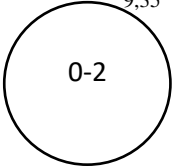
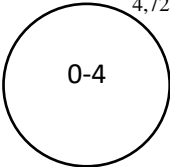
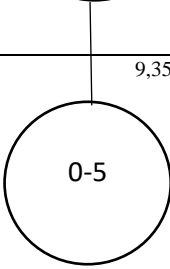
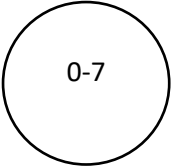
Lampiran 7 MPPC Paving Block

Mesin	Komponen Paving Block				
	1	2	3	N. hitung	N. aktual
Mesin I Molen kering				32,41	33
Mesin II Sekop				32,41	33
Mesin III Pengayakan				32,41	33
Mesin IV Mesin cetak				32,41	33

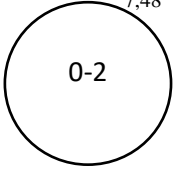
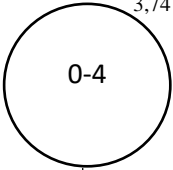
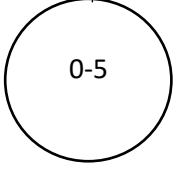
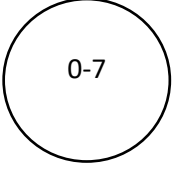
Lampiran 8 MPPC Uskup

Mesin	Komponen Uskup				
	1	2	3	N. hitung	N. aktual
Mesin I Molen kering	32,41 			32,41	33
Mesin II Sekop		13,37 		13,37	14
Mesin III Pengayakan		32,41 		32,41	33
Mesin IV Mesin cetak			33,402 	33,402	34

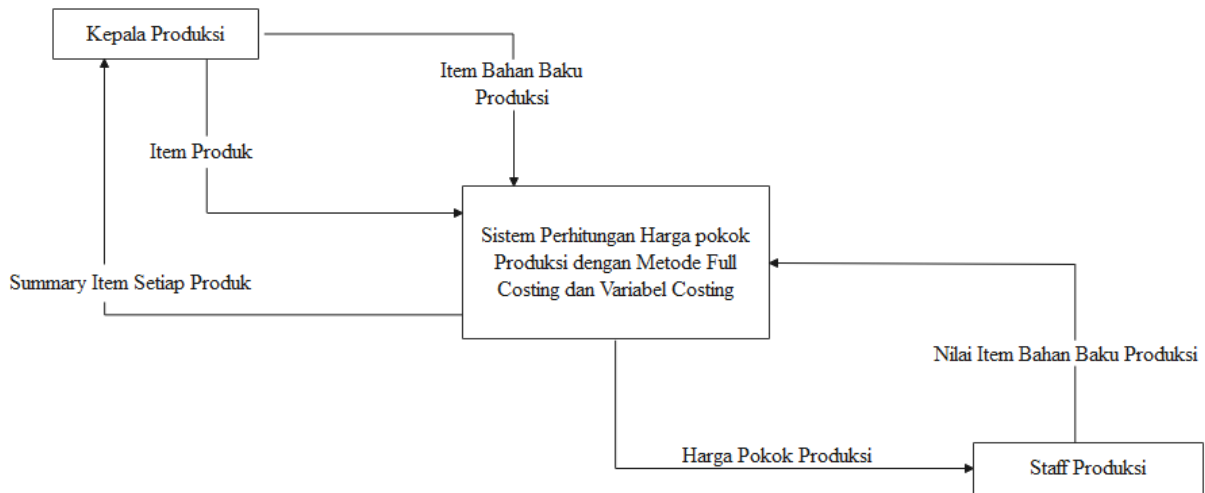
Lampiran 9 MPPC Kanstin

Mesin	Komponen Kanstin				
	1	2	3	N. hitung	N. aktual
Mesin I Molen kering				9,35	10
Mesin II Sekop				4,72	5
Mesin III Pengayakan				9,35	10
Mesin IV Mesin cetak				4,72	5

Lampiran 10 MPPC Batako

Mesin	Komponen Batako				
	1	2	3	N. hitung	N. aktual
Mesin I Molen kering				7,48	8
Mesin II Sekop				3,74	4
Mesin III Pengayakan				7,84	8
Mesin IV Mesin cetak				7,84	8

Lampiran 11 DFD Untuk Aplikasi Perhitungan Harga Pokok Produksi Yang Berbasis Web



Penjelasan DFD :

Summary Item Setiap Produk berhubungan dengan kepala produksi. Kepala produksi sendiri dibagi menjadi dua yaitu Item produk dan bahan baku produksi dimana masing-masing setiap item juga melakukan sistem perhitungan harga pokok produksi dengan menggunakan metode full costing dan variable costing, harga pokok produksi sendiri dibagi menjadi staff produksi dimana staff produksi tersebut juga memiliki nilai item bahan baku produksi setelah itu kembali lagi ke sistem perhitungan harga pokok produksi dengan metode full costing dan variabel costing.

Lampiran 12 Coding Aplikasi Perhitungan Harga Pokok Produksi Berbasis Web

a. Controllers

1. AccountController.cs

```
using AutoMapper;
using CMS.BLL.User.Logics;
using CMS.BLL.User.Models;
using CMS.Website.Models.Account;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Security.Cryptography;
using System.Text;
using System.Web;
using System.Web.Mvc;
using System.Web.Security;
using System.Web.SessionState;
using WebMatrix.WebData;
```

```
namespace CMS.Website.Controllers.Account
```

```
{
    public class AccountController : Controller
    {
        private MasterUserBLL _masterUserBLL = new MasterUserBLL();

        [AllowAnonymous]
        public ActionResult Login(string returnUrl)
        {
            ViewBag.ReturnUrl = returnUrl;
            ViewBag.Error = "";
            return View();
        }

        [HttpPost]
        [AllowAnonymous]
        [ValidateAntiForgeryToken]
        public ActionResult Login(MasterUserVM model, string returnUrl)
        {
            var EncryptPassword = ToSHA256(model.Password);
            //var getUserResult = _masterUserBLL.Get(x => x.Username.ToLower() ==
            model.Username.ToLower() && x.IsActive == true && x.IsDelete == false);
            var getUserResult = _masterUserBLL.Get(x => x.Username.ToLower() ==
            model.Username.ToLower() && x.IsDelete == false);
            model = (getUserResult.Status.IsSuccess) ? Mapper.Map<MasterUserDto,
            MasterUserVM>(getUserResult.Data) : new MasterUserVM();
```

```

//if (ModelState.IsValid && WebSecurity.Login(model.Username, model.Password))
//{
//  return RedirectToLocal(returnUrl);
//}

if (model != null)
{
  // Redirect so the next request can see the user as authenticated
  if (model.IsActive == false)
  {
    ViewBag.Error = "User tidak aktif";
    return View();
  }
  if (model.Password != EncryptPassword)
  {
    if (Url.IsLocalUrl(returnUrl) && returnUrl.Length > 1 && returnUrl.StartsWith("/")
    && !returnUrl.StartsWith("//") && !returnUrl.StartsWith(@"\"))
    {
      ViewBag.Error = "Username & Password Tidak Sesuai";
      return Redirect(returnUrl);
    }
    else
    {
      ViewBag.Error = "Username & Password Tidak Sesuai";
      return View();
    }
  }
  else
  {
    FormsAuthentication.SetAuthCookie(model.Username, true);
    return RedirectToAction("Index", "HPP");
  }
}
ViewBag.Error = "Username & Password Tidak Sesuai";
return View();
}

public ActionResult Logout()
{
  FormsAuthentication.SignOut();
  return RedirectToAction("Login", "Account");
}

public string ToSHA256(string value)
{
  SHA256 sha256 = SHA256.Create();

```

```

byte[] hashData = sha256.ComputeHash(Encoding.Default.GetBytes(value));
StringBuilder returnValue = new StringBuilder();

for (int i = 0; i < hashData.Length; i++)
{
    returnValue.Append(hashData[i].ToString());
}

return returnValue.ToString();
}
}
}

```

2. HPPController.cs

```

using AutoMapper;
using CMS.BLL.HPP.Logics;
using CMS.BLL.HPP.Models;
using CMS.DataHelpers;
using CMS.Models.Http;
using CMS.Website.Models.HPP;
using DataTables.Mvc;
using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.Mvc;

namespace CMS.Website.Controllers.HPP
{
    public class HPPController : CoreController
    {
        private ProdukBLL _ProdukBLL = new ProdukBLL();

        // GET: User
        public ActionResult Index()
        {
            ViewBag.Menu = Enums>NamaMenu.Produk.ToString();
            ViewBag.Header = Enums.GetDescription(Enums>NamaMenu.Produk);
            ViewBag.User = GetCurrentUser();
            ViewBag.Width = 0;
            ViewBag.Height = 0;
            ViewBag.PathFile = "";
            return View();
        }
    }
}

```

```

    public ActionResult GetData([ModelBinder(typeof(DataTablesBinder))]
IDataTablesRequest requestModel)
    {
        IEnumerable<ProdukVM> list = null;
        string inputSearch = requestModel.Search.Value;

        var getAllResult = _ProdukBLL.GetAllByPaging(
            requestModel.Start,
            requestModel.Length,
            x => x.OrderBy(y => y.Kode),
            inputSearch);

        if (getAllResult.Status.IsSuccess)
            list = Mapper.Map<IEnumerable<ProdukDto>,
IEnumerable<ProdukVM>>(getAllResult.Data);
        var totalCount = _ProdukBLL.Count(inputSearch);

        var data = list.Select(x => new
        {
            ID = x.ID,
            Kode = x.Kode,
            Nama = x.Nama,
            ProdukDetails = x.ProdukDetails
        }).ToList();

        int recordsTotal = totalCount;
        int recordsFiltered = totalCount;
        int startRow = requestModel.Start;

        return Json(new { data, recordsTotal, recordsFiltered, startRow },
JsonRequestBehavior.AllowGet);
    }

    [HttpGet]
    public JsonResult GetById(Guid ID)
    {
        ProdukVM model = null;

        var getAreaDampakResult = _ProdukBLL.Get(x => x.ID == ID, "ProdukDetails");
        model = (getAreaDampakResult.Status.IsSuccess && ID != Guid.Empty) ? Mapper
            .Map<ProdukDto, ProdukVM>(getAreaDampakResult.Data) : new ProdukVM();

        return Json(new { data = model, status = getAreaDampakResult.Status },
JsonRequestBehavior.AllowGet);
    }

```

```

[HttpPost]
public ActionResult Edit(ProdukVM model)
{
    //var model = JsonConvert.DeserializeObject<ProdukVM>(data);

    bool success = true;
    List<string> errors = new List<string>();
    try
    {
        if (!ModelState.IsValid)
        {
            success = false;
            foreach (var kvp in ModelState)
            {
                if (kvp.Value != null)
                {
                    foreach (var error in kvp.Value.Errors)
                    {
                        errors.Add(error.ErrorMessage);
                    }
                }
            }
        }
        else
        {
            if (model.ID != Guid.Empty)
            {
                #region Region Update Data
                model.UpdatedBy = GetCurrentUser();
                model.UpdatedDate = DateTime.Now;

                //var updateResult = _ProdukBLL.Update(Mapper.Map<ProdukVM,
                ProdukDto>(model));

                foreach(var dt in model.ProdukDetails)
                {
                    var updateResultDetail =
                    _ProdukBLL.UpdateDetail(Mapper.Map<ProdukDetailVM, ProdukDetailDto>(dt));
                    success = updateResultDetail.IsSuccess;
                    errors = updateResultDetail.Errors;
                }

                //success = updateResult.IsSuccess;
                //errors = updateResult.Errors;

                #endregion
            }
        }
    }
}

```



```

    }
    else
    {
        #region Region Insert Data
        model.ID = Guid.NewGuid();
        model.CreatedBy = GetCurrentUser();
        model.CreatedDate = DateTime.Now;

        var insertResult = _ProdukBLL.Insert(Mapper.Map<ProdukVM,
ProdukDto>(model));

        success = insertResult.Status.IsSuccess;
        errors = insertResult.Status.Errors;
        #endregion
    }
}
}
catch (Exception ex)
{
    return PrintError(new string[] { ex.Message });
}
return Json(new ResponseFlag(success, errors));
}

[HttpPost]
public ActionResult Delete(Guid ID)
{
    bool success = true;
    List<string> errors = new List<string>();
    try
    {
        if (ID != Guid.Empty)
        {
            #region Region Update Data
            var updateResult = _ProdukBLL.Delete(ID);

            success = updateResult.IsSuccess;
            errors = updateResult.Errors;

            #endregion
        }
    }
    catch (Exception ex)
    {
        return PrintError(new string[] { ex.Message });
    }
}

```

```

        return Json(new ResponseFlag(success, errors));
    }
}
}

```

3. ItemController.cs

```

using AutoMapper;
using CMS.Models.Http;
using DataTables.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web.Mvc;
using System.Security.Cryptography;
using System.Text;
using CMS.DataHelpers;
using CMS.BLL.Item.Logics;
using CMS.Website.Models.Item;
using CMS.BLL.Item.Models;

namespace CMS.Website.Controllers.Item
{
    public class ItemController : CoreController
    {
        private ItemBLL _ItemBLL = new ItemBLL();

        // GET: User
        public ActionResult Index()
        {
            ViewBag.Menu = Enums>NamaMenu.Item.ToString();
            ViewBag.Header = Enums.GetDescription(Enums>NamaMenu.Item);
            ViewBag.User = GetCurrentUser();
            ViewBag.Width = 0;
            ViewBag.Height = 0;
            ViewBag.PathFile = "";
            return View();
        }

        public ActionResult GetData([ModelBinder(typeof(DataTablesBinder))]
IDataTablesRequest requestModel)
        {
            IEnumerable<ItemVM> list = null;
            string inputSearch = requestModel.Search.Value;

            var getAllResult = _ItemBLL.GetAllByPaging(

```

```

        requestModel.Start,
        requestModel.Length,
        x => x.OrderBy(y => y.NoUrut),
        inputSearch);

    if (getAllResult.Status.IsSuccess)
        list = Mapper.Map<IEnumerable<ItemDto>,
IEnumerable<ItemVM>>(getAllResult.Data);
    var totalCount = _ItemBLL.Count(inputSearch);

    var data = list.Select(x => new
    {
        ID = x.ID,
        Kode = x.Kode,
        Nama = x.Nama,
        Formula = x.Formula,
        IsActive = x.IsActive,
        NoUrut = x.NoUrut,
        Jenis = x.Jenis
    }).ToList();

    int recordsTotal = totalCount;
    int recordsFiltered = totalCount;
    int startRow = requestModel.Start;

    return Json(new { data, recordsTotal, recordsFiltered, startRow },
JsonRequestBehavior.AllowGet);
}

[HttpGet]
public JsonResult GetById(Guid ID)
{
    ItemVM model = null;

    var getAreaDampakResult = _ItemBLL.Get(x => x.ID == ID);
    model = (getAreaDampakResult.Status.IsSuccess && ID != Guid.Empty) ? Mapper
        .Map<ItemDto, ItemVM>(getAreaDampakResult.Data) : new ItemVM();

    return Json(new { data = model, status = getAreaDampakResult.Status },
JsonRequestBehavior.AllowGet);
}

[HttpPost]
public ActionResult Edit(ItemVM model)
{
    bool success = true;

```

```

List<string> errors = new List<string>();
try
{
    if (!ModelState.IsValid)
    {
        success = false;
        foreach (var kvp in ModelState)
        {
            if (kvp.Value != null)
            {
                foreach (var error in kvp.Value.Errors)
                {
                    errors.Add(error.ErrorMessage);
                }
            }
        }
    }
    else
    {
        if (model.ID != Guid.Empty)
        {
            #region Region Update Data
            model.UpdatedBy = GetCurrentUser();
            model.UpdatedDate = DateTime.Now;

            var updateResult = _ItemBLL.Update(Mapper.Map<ItemVM,
ItemDto>(model));

            success = updateResult.IsSuccess;
            errors = updateResult.Errors;

            #endregion
        }
        else
        {
            #region Region Insert Data
            model.ID = Guid.NewGuid();
            model.CreatedBy = GetCurrentUser();
            model.CreatedDate = DateTime.Now;

            var insertResult = _ItemBLL.Insert(Mapper.Map<ItemVM, ItemDto>(model));

            success = insertResult.Status.IsSuccess;
            errors = insertResult.Status.Errors;
            #endregion
        }
    }
}

```

```

    }
}
catch (Exception ex)
{
    return PrintError(new string[] { ex.Message });
}
return Json(new ResponseFlag(success, errors));
}

[HttpPost]
public ActionResult Delete(Guid ID)
{
    bool success = true;
    List<string> errors = new List<string>();
    try
    {
        if (ID != Guid.Empty)
        {
            #region Region Update Data
            var updateResult = _ItemBLL.Delete(ID);

            success = updateResult.IsSuccess;
            errors = updateResult.Errors;

            #endregion
        }
    }
    catch (Exception ex)
    {
        return PrintError(new string[] { ex.Message });
    }
    return Json(new ResponseFlag(success, errors));
}
}
}

```

4. UserController.cs

```

using AutoMapper;
using CMS.BLL.User.Logics;
using CMS.BLL.User.Models;
using CMS.Models.Http;
using CMS.Website.Models.User;
using DataTables.Mvc;
using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Web;
using System.Web.Mvc;
using System.Security.Cryptography;
using System.Text;
using CMS.DataHelpers;

namespace CMS.Website.Controllers.User
{
    public class UserController : CoreController
    {
        private MasterUserBLL _masterUserBLL = new MasterUserBLL();

        // GET: User
        public ActionResult Index()
        {
            ViewBag.Menu = Enums>NamaMenu.User.ToString();
            ViewBag.Header = Enums.GetDescription(Enums>NamaMenu.User);
            ViewBag.User = GetCurrentUser();
            ViewBag.Width = 0;
            ViewBag.Height = 0;
            ViewBag.PathFile = "";
            return View();
        }

        public ActionResult GetData([ModelBinder(typeof(DataTablesBinder))]
IDataTablesRequest requestModel)
        {
            IEnumerable<MasterUserVM> list = null;
            string inputSearch = requestModel.Search.Value;

            var getAllResult = _masterUserBLL.GetAllByPaging(
                requestModel.Start,
                requestModel.Length,
                x => x.OrderByDescending(y => y.CreatedDate),
                inputSearch);

            if (getAllResult.Status.IsSuccess)
                list = Mapper.Map<IEnumerable<MasterUserDto>,
IEnumerable<MasterUserVM>>(getAllResult.Data);
            var totalCount = _masterUserBLL.Count(inputSearch);

            var data = list.Select(x => new
            {
                ID = x.ID,
                Username = x.Username,
                IsActive = x.IsActive
            }

```

```

    }).ToList();

    int recordsTotal = totalCount;
    int recordsFiltered = totalCount;
    int startRow = requestModel.Start;

    return Json(new { data, recordsTotal, recordsFiltered, startRow },
    JsonRequestBehavior.AllowGet);
}

[HttpGet]
public JsonResult GetById(Guid ID)
{
    MasterUserVM model = null;

    var getAreaDampakResult = _masterUserBLL.Get(x => x.ID == ID);
    model = (getAreaDampakResult.Status.IsSuccess && ID != Guid.Empty) ? Mapper
        .Map<MasterUserDto, MasterUserVM>(getAreaDampakResult.Data) : new
    MasterUserVM();

    return Json(new { data = model, status = getAreaDampakResult.Status },
    JsonRequestBehavior.AllowGet);
}

[HttpPost]
public ActionResult Edit(MasterUserVM model)
{
    bool success = true;
    List<string> errors = new List<string>();
    try
    {
        if (!ModelState.IsValid)
        {
            success = false;
            foreach (var kvp in ModelState)
            {
                if (kvp.Value != null)
                {
                    foreach (var error in kvp.Value.Errors)
                    {
                        errors.Add(error.ErrorMessage);
                    }
                }
            }
        }
    }
    else

```

```

    {
        if (model.ID != Guid.Empty)
        {
            #region Region Update Data
            model.Password = ToSHA256(model.Password);
            model.UpdatedBy = GetCurrentUser();
            model.UpdatedDate = DateTime.Now;

            var updateResult = _masterUserBLL.Update(Mapper.Map<MasterUserVM,
MasterUserDto >(model));

            success = updateResult.IsSuccess;
            errors = updateResult.Errors;

            #endregion
        }
        else
        {
            #region Region Insert Data
            model.ID = Guid.NewGuid();
            model.Password = ToSHA256(model.Password);
            model.CreatedBy = GetCurrentUser();
            model.CreatedDate = DateTime.Now;

            var insertResult = _masterUserBLL.Insert(Mapper.Map<MasterUserVM,
MasterUserDto>(model));

            success = insertResult.Status.IsSuccess;
            errors = insertResult.Status.Errors;
            #endregion
        }
    }
}
catch (Exception ex)
{
    return PrintError(new string[] { ex.Message });
}
return Json(new ResponseFlag(success, errors));
}

[HttpPost]
public ActionResult Delete(Guid ID)
{
    bool success = true;
    List<string> errors = new List<string>();
    try

```



```

    {
        if (ID != Guid.Empty)
        {
            #region Region Update Data
            var updateResult = _masterUserBLL.Delete(ID);

            success = updateResult.IsSuccess;
            errors = updateResult.Errors;

            #endregion
        }
    }
    catch (Exception ex)
    {
        return PrintError(new string[] { ex.Message });
    }
    return Json(new ResponseFlag(success, errors));
}

public string ToSHA256(string value)
{
    SHA256 sha256 = SHA256.Create();

    byte[] hashData = sha256.ComputeHash(Encoding.Default.GetBytes(value));
    StringBuilder returnValue = new StringBuilder();

    for (int i = 0; i < hashData.Length; i++)
    {
        returnValue.Append(hashData[i].ToString());
    }

    return returnValue.ToString();
}
}
}

```

5. CoreController.cs

```

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

namespace CMS.Website.Controllers
{
    public class CoreController : Controller

```

```

{
protected virtual ActionResult PrintError(IEnumerable<string> errors)
{
    return View("Error", errors);
}
protected virtual ActionResult RedirectToLocal(string returnUrl)
{
    if (Url.IsLocalUrl(returnUrl))
    {
        return Redirect(returnUrl);
    }
    return RedirectToAction("Index", "Home");
}
protected virtual void AddCookie(string key, string value, DateTime expires)
{
    try
    {
        HttpCookie cookie = null;
        if (Request.Cookies.AllKeys.Contains(key))
        {
            cookie = Request.Cookies[key];
        }
        else
        {
            cookie = new HttpCookie(key);
        }
        cookie.Value = value;
        cookie.Expires = expires;
        Response.Cookies.Add(cookie);
    }
    catch { }
}
protected virtual object GetCookie(string key)
{
    if (Request.Cookies.AllKeys.Contains(key))
        return Request.Cookies[key];
    return null;
}
protected virtual string GetCurrentUser()
{
    try
    {
        return User.Identity.IsAuthenticated ? User.Identity.Name : "Anonymous";
        //return "SYSTEM";
    }
    catch

```

```

        {
            return string.Empty;
        }
    }
}
}

```

6. HomeController.cs

```

using Newtonsoft.Json;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Net.Http.Headers;
using System.Threading.Tasks;
using System.Web;
using System.Web.Mvc;

namespace CMS.Website.Controllers
{
    public class HomeController : Controller
    {
        public ActionResult Index()
        {
            return View();
        }

        public ActionResult About()
        {
            ViewBag.Message = "Your application description page.";

            return View();
        }

        public ActionResult Contact()
        {
            ViewBag.Message = "Your contact page.";

            return View();
        }

        string Baseurl = "http://www.pgncms.test/";
        public async Task<ActionResult> GetData()
        {
            DividendInformationEditVM data = new DividendInformationEditVM();

```

```

using (var client = new HttpClient())
{
    client.DefaultRequestHeaders.Authorization = new
AuthenticationHeaderValue("Bearer",

"eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJ1bmlxdWVfmFtZSI6ImFkbWluIiwibmJmIjoxNjA4Mzc2Njg5LCJleHAiOjE2MDgzODAyODksImhhdCI6MTYwODM3NjY4OSwiaXNzIjoicGduY21zIiwiaXVkiOicGduY21zIn0.-PyoHqQ1m_Ql1VIJbI62sw9OmpDHVSfn6HXRgDQ56NA");
    client.BaseAddress = new Uri(Baseurl);
    client.DefaultRequestHeaders.Clear();
    client.DefaultRequestHeaders.Accept.Add(new
MediaTypeWithQualityHeaderValue("application/json"));

    HttpResponseMessage Res = await client.PostAsync("api/stock/", null);

    if (Res.IsSuccessStatusCode)
    {
        var DataResponse = Res.Content.ReadAsStringAsync().Result;
        data =
JsonConvert.DeserializeObject<DividendInformationEditVM>(DataResponse);

    }
    return View(data);
}

public class DividendInformationEditVM
{
    public DividendInformationEditVM()
    {
        Message = new List<string>();
    }
    public string Status { get; set; }
    public List<string> Message { get; set; }
    public Data Data { get; set; }
}

public class Data
{
    public MenuVM Menu { get; set; }
    public List<DividendInformationVM> DividendInformation { get; set; }
}

public class MenuVM
{

```

```
public Guid ID { get; set; }
public string PathFotoPreview { get; set; }
public string NamaMenu { get; set; }
public string NamaSideMenu { get; set; }
public string Title { get; set; }
public string Konten { get; set; }
public string Subtitle1 { get; set; }
public string Subtitle2 { get; set; }
public string Subtitle3 { get; set; }
public string Subtitle4 { get; set; }
public string Url1 { get; set; }
public string Url2 { get; set; }
public string Url3 { get; set; }
public string Url4 { get; set; }
public Guid? Slider1ID { get; set; }
public Guid? Slider2ID { get; set; }
public Guid? Slider3ID { get; set; }
public string Foto1 { get; set; }
public string PathFoto1 { get; set; }
public string Foto2 { get; set; }
public string PathFoto2 { get; set; }
public string CaptionFoto2 { get; set; }
public string CaptionFoto1 { get; set; }
public string Faq1 { get; set; }
public string JawabanFaq1 { get; set; }
public string Faq2 { get; set; }
public string JawabanFaq2 { get; set; }
public string Faq3 { get; set; }
public string JawabanFaq3 { get; set; }
public string Faq4 { get; set; }
public string JawabanFaq4 { get; set; }
public string Faq5 { get; set; }
public string JawabanFaq5 { get; set; }
public string Keterangan1 { get; set; }
public string Keterangan2 { get; set; }
public string Contact1 { get; set; }
public string KeteranganContact1 { get; set; }
public string Contact2 { get; set; }
public string KeteranganContact2 { get; set; }
public string AlertType1 { get; set; }
public string AlertType2 { get; set; }
public string AlertType3 { get; set; }
public string AlertType4 { get; set; }
public string AlertType5 { get; set; }
public bool? IsActive { get; set; }
public DateTime? CreatedDate { get; set; }
```

```

    public string CreatedBy { get; set; }
    public DateTime? UpdatedDate { get; set; }
    public string UpdatedBy { get; set; }
    public bool? IsDelete { get; set; }
    public string SliderFoto1 { get; set; }
    public string SliderPathfoto1 { get; set; }
    public decimal? SliderFilesize1 { get; set; }
    public string SliderFoto2 { get; set; }
    public string SliderPathfoto2 { get; set; }
    public decimal? SliderFilesize2 { get; set; }
    public string SliderFoto3 { get; set; }
    public string SliderPathfoto3 { get; set; }
    public decimal? SliderFilesize3 { get; set; }
}

public class DividendInformationVM
{
    public Guid ID { get; set; }
    public Guid MenuID { get; set; }
    public int Year { get; set; }
    public DateTime? ExDividend { get; set; }
    public DateTime? RecordingDate { get; set; }
    public DateTime? Declare { get; set; }
    public DateTime? Payable { get; set; }
    public DateTime? CumDividend { get; set; }
    public decimal? DividendPershare { get; set; }
    public bool? IsActive { get; set; }
    public DateTime? CreatedDate { get; set; }
    public string CreatedBy { get; set; }
    public DateTime? UpdatedDate { get; set; }
    public string UpdatedBy { get; set; }
    public bool? IsDelete { get; set; }
}
}
}

```

b. Models

1. Account

MasterUserMapperVM.cs

```
using AutoMapper;
using CMS.BLL.User.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
```

```
namespace CMS.Website.Models.Account
{
    public class MasterUserMapperVM : Profile
    {
        public MasterUserMapperVM()
        {
            CreateMap<MasterUserDto, MasterUserVM>().ReverseMap();
        }
    }
}
```

MasterUserVM.cs

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

```
namespace CMS.Website.Models.Account
{
    public class MasterUserVM
    {
        public Guid ID { get; set; }
        public string Username { get; set; }
        public string Password { get; set; }
        public bool IsActive { get; set; }
        public DateTime? CreatedDate { get; set; }
        public string CreatedBy { get; set; }
        public DateTime? UpdatedDate { get; set; }
        public string UpdatedBy { get; set; }
    }
}
```

2. HPP

- ProdukDetailVM.cs

```
using System;
```

```
using System.Collections.Generic;
```

```

using System.Linq;
using System.Web;

namespace CMS.Website.Models.HPP
{
    public class ProdukDetailVM
    {
        public Guid ID { get; set; }
        public Guid ProdukID { get; set; }
        public Guid ItemID { get; set; }
        public string Kode { get; set; }
        public string Nama { get; set; }
        public string NoUrut { get; set; }
        public string Formula { get; set; }
        public decimal? NilaiItem { get; set; }
        public bool IsActive { get; set; }
        public bool IsDelete { get; set; }
        public DateTime? CreatedDate { get; set; }
        public string CreatedBy { get; set; }
        public DateTime? UpdatedDate { get; set; }
        public string UpdatedBy { get; set; }
        public string Jenis { get; set; }
        public bool IsFullCosting { get; set; }
    }
}

```

- ProdukMapperVM.cs

```

using AutoMapper;
using CMS.BLL.HPP.Models;
using System;
using System.Collections.Generic;

```



```

using System.Linq;
using System.Web;
namespace CMS.Website.Models.HPP
{
    public class ProdukMapperVM : Profile
    {
        public ProdukMapperVM()
        {
            CreateMap<ProdukDto, ProdukVM>().ReverseMap();
        }
    }
}

```

- ProdukVM.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
namespace CMS.Website.Models.HPP
{
    public class ProdukVM
    {
        public ProdukVM()
        {
            ProdukDetails = new List<ProdukDetailVM>();
        }
        public Guid ID { get; set; }
        public string Kode { get; set; }
        public string Nama { get; set; }
        public bool IsActive { get; set; }
        public bool IsDelete { get; set; }
    }
}

```

```

    public DateTime? CreatedDate { get; set; }
    public string CreatedBy { get; set; }
    public DateTime? UpdatedDate { get; set; }
    public string UpdatedBy { get; set; }
    public List<ProdukDetailVM> ProdukDetails { get; set; }
}
}

```

3. Http

HttpResponseResult.cs

```

using System;

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

namespace CMS.Models.Http
{
    public class ResponseData<T>
    {
        public ResponseData() { }
        public ResponseData(T data, ResponseFlag flag)
        {
            this.Data = data;
            this.Flag = flag;
        }
        public T Data { get; set; }
        public ResponseFlag Flag { get; set; }
    }
    public class ResponseFlag
    {
        public ResponseFlag() { }

```

```

public ResponseFlag(bool success, IEnumerable<string> errors, int id = 0)
{
    this.Success = success;
    this.Errors = errors;
    this.Id = id;
}
public bool Success { get; set; }
public IEnumerable<string> Errors { get; set; }
public int Id { get; set; }
}
}

```

4. Item

ItemMapperVM.cs

```

using AutoMapper;
using CMS.BLL.Item.Models;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

```

```

namespace CMS.Website.Models.Item

```

```

{
    public class ItemMapperVM : Profile
    {
        public ItemMapperVM()
        {
            CreateMap<ItemDto, ItemVM>().ReverseMap();
        }
    }
}
}

```

```

ItemVM.cs
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;

namespace CMS.Website.Models.Item
{
    public class ItemVM
    {
        public Guid ID { get; set; }
        public string Kode { get; set; }
        public string Nama { get; set; }
        public string NoUrut { get; set; }
        public string Formula { get; set; }
        public bool IsActive { get; set; }
        public bool IsDelete { get; set; }
        public DateTime? CreatedDate { get; set; }
        public string CreatedBy { get; set; }
        public DateTime? UpdatedDate { get; set; }
        public string UpdatedBy { get; set; }
        public string Jenis { get; set; }
        public bool IsFullCosting { get; set; }
    }
}

```

5. User

```

MasterUserMapperVM.cs
using AutoMapper;
using CMS.BLL.User.Models;
using System;
using System.Collections.Generic;
using System.Linq;

```

```

using System.Web;

namespace CMS.Website.Models.User
{
    public class MasterUserMapperVM : Profile
    {
        public MasterUserMapperVM()
        {
            CreateMap<MasterUserDto, MasterUserVM>().ReverseMap();
        }
    }
}

```

MasterUserVM.cs

```

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

```

```

namespace CMS.Website.Models.User
{
    public class MasterUserVM
    {
        public Guid ID { get; set; }
        [Required(ErrorMessage = "Username Tidak Boleh Kosong")]
        public string Username { get; set; }
        [Required(ErrorMessage = "Password Tidak Boleh Kosong")]
        public string Password { get; set; }
        public bool IsActive { get; set; }
        public DateTime? CreatedDate { get; set; }
        public string CreatedBy { get; set; }
        public DateTime? UpdatedDate { get; set; }
        public string UpdatedBy { get; set; }
        public bool IsDelete { get; set; }
    }
}

```

6. AccountViewModels.cs

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;

```

```

namespace CMS.Website.Models
{
    public class ExternalLoginConfirmationViewModel
    {
        [Required]
        [Display(Name = "Email")]
    }
}

```

```
    public string Email { get; set; }
}

public class ExternalLoginListViewModel
{
    public string returnUrl { get; set; }
}

public class SendCodeViewModel
{
    public string SelectedProvider { get; set; }
    public ICollection<System.Web.Mvc.SelectListItem> Providers { get; set; }
    public string returnUrl { get; set; }
    public bool RememberMe { get; set; }
}

public class VerifyCodeViewModel
{
    [Required]
    public string Provider { get; set; }

    [Required]
    [Display(Name = "Code")]
    public string Code { get; set; }
    public string returnUrl { get; set; }

    [Display(Name = "Remember this browser?")]
    public bool RememberBrowser { get; set; }

    public bool RememberMe { get; set; }
}

public class ForgotViewModel
{
    [Required]
    [Display(Name = "Email")]
    public string Email { get; set; }
}

public class LoginViewModel
{
    [Required]
    [Display(Name = "Email")]
    [EmailAddress]
    public string Email { get; set; }
```

```

    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [Display(Name = "Remember me?")]
    public bool RememberMe { get; set; }
}

public class RegisterViewModel
{
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    public string Email { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.",
MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]
    [Compare("Password", ErrorMessage = "The password and confirmation password do not
match.")]
    public string ConfirmPassword { get; set; }
}

public class ResetPasswordViewModel
{
    [Required]
    [EmailAddress]
    [Display(Name = "Email")]
    public string Email { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.",
MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "Password")]
    public string Password { get; set; }
    [DataType(DataType.Password)]
    [Display(Name = "Confirm password")]

```

```
[Compare("Password", ErrorMessage = "The password and confirmation password do not match.")]
```

```
public string ConfirmPassword { get; set; }
```

```
public string Code { get; set; }
```

```
}
```

```
public class ForgotPasswordViewModel
```

```
{
```

```
    [Required]
```

```
    [EmailAddress]
```

```
    [Display(Name = "Email")]
```

```
    public string Email { get; set; }
```

```
}
```

```
}
```

7. IdentityModels.cs

```
using System.Data.Entity;
```

```
using System.Security.Claims;
```

```
using System.Threading.Tasks;
```

```
using Microsoft.AspNet.Identity;
```

```
using Microsoft.AspNet.Identity.EntityFramework;
```

```
namespace CMS.Website.Models
```

```
{
```

```
    // You can add profile data for the user by adding more properties to your ApplicationUser class, please visit http://go.microsoft.com/fwlink/?LinkID=317594 to learn more.
```

```
    public class ApplicationUser : IdentityUser
```

```
    {
```

```
        public async Task<ClaimsIdentity>
```

```
GenerateUserIdentityAsync(UserManager<ApplicationUser> manager)
```

```
    {
```

```
        // Note the authenticationType must match the one defined in CookieAuthenticationOptions.AuthenticationType
```

```
        var userIdentity = await manager.CreateIdentityAsync(this, DefaultAuthenticationTypes.ApplicationCookie);
```

```
        // Add custom user claims here
```

```
        return userIdentity;
```

```
    }
```

```
}
```

```
public class ApplicationDbContext : IdentityDbContext<ApplicationUser>
```

```
{
```

```
    public ApplicationDbContext()
```

```
        : base("DefaultConnection", throwIfV1Schema: false)
```

```
    {
```

```
    }
```

```
}
```



```

        public static ApplicationDbContext Create()
        {
            return new ApplicationDbContext();
        }
    }
}

```

8. ManagerViewModels.cs

```

using System.Collections.Generic;
using System.ComponentModel.DataAnnotations;
using Microsoft.AspNet.Identity;
using Microsoft.Owin.Security;

```

```

namespace CMS.Website.Models

```

```

{
    public class IndexViewModel
    {
        public bool HasPassword { get; set; }
        public IList<UserLoginInfo> Logins { get; set; }
        public string PhoneNumber { get; set; }
        public bool TwoFactor { get; set; }
        public bool BrowserRemembered { get; set; }
    }

    public class ManageLoginsViewModel
    {
        public IList<UserLoginInfo> CurrentLogins { get; set; }
        public IList<AuthenticationDescription> OtherLogins { get; set; }
    }

    public class FactorViewModel
    {
        public string Purpose { get; set; }
    }

    public class SetPasswordViewModel
    {
        [Required]
        [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.",
        MinimumLength = 6)]
        [DataType(DataType.Password)]
        [Display(Name = "New password")]
        public string NewPassword { get; set; }

        [DataType(DataType.Password)]
        [Display(Name = "Confirm new password")]

```

```

    [Compare("NewPassword", ErrorMessage = "The new password and confirmation
password do not match.")]
    public string ConfirmPassword { get; set; }
}

public class ChangePasswordViewModel
{
    [Required]
    [DataType(DataType.Password)]
    [Display(Name = "Current password")]
    public string OldPassword { get; set; }

    [Required]
    [StringLength(100, ErrorMessage = "The {0} must be at least {2} characters long.",
MinimumLength = 6)]
    [DataType(DataType.Password)]
    [Display(Name = "New password")]
    public string NewPassword { get; set; }

    [DataType(DataType.Password)]
    [Display(Name = "Confirm new password")]
    [Compare("NewPassword", ErrorMessage = "The new password and confirmation
password do not match.")]
    public string ConfirmPassword { get; set; }
}

public class AddPhoneNumberViewModel
{
    [Required]
    [Phone]
    [Display(Name = "Phone Number")]
    public string Number { get; set; }
}

public class VerifyPhoneNumberViewModel
{
    [Required]
    [Display(Name = "Code")]
    public string Code { get; set; }

    [Required]
    [Phone]
    [Display(Name = "Phone Number")]
    public string PhoneNumber { get; set; }
}

```

```

public class ConfigureTwoFactorViewModel
{
    public string SelectedProvider { get; set; }
    public ICollection<System.Web.Mvc.SelectListItem> Providers { get; set; }
}

```

9. TokenValidationHandler.cs

```

using Microsoft.IdentityModel.Tokens;
using System;
using System.Collections.Generic;
using System.IdentityModel.Tokens.Jwt;
using System.Linq;
using System.Net;
using System.Net.Http;
using System.Threading;
using System.Threading.Tasks;
using System.Web;

namespace CMS.Website.Models
{
    public class TokenValidationHandler : DelegatingHandler
    {
        private static bool TryRetrieveToken(HttpRequestMessage request, out string token)
        {
            token = null;
            IEnumerable<string> authzHeaders;
            if (!request.Headers.TryGetValues("Authorization", out authzHeaders) ||
authzHeaders.Count() > 1)
            {
                return false;
            }
            var bearerToken = authzHeaders.ElementAt(0);
            token = bearerToken.StartsWith("Bearer ") ? bearerToken.Substring(7) : bearerToken;
            return true;
        }

        protected override Task<HttpResponseMessage> SendAsync(HttpRequestMessage
request, CancellationToken cancellationToken)
        {
            HttpStatusCode statusCode;
            string token;
            //determine whether a jwt exists or not
            if (!TryRetrieveToken(request, out token))
            {
                statusCode = HttpStatusCode.Unauthorized;
            }
        }
    }
}

```

```

        //allow requests with no token - whether a action method needs an authentication can
        be set with the claimsauthorization attribute
        return base.SendAsync(request, cancellationTokens);
    }

    try
    {
        const string sec =
"612501821261602191181281320530512087624153158584415220717616290702301912262
181935";/"401b09eab3c013d4ca54922bb802bec8fd5318192b0a75f201d8b3727429090fb337
591abd3e44453b954555b7a0812e1081c39b740293f765eae731f5a65ed1";
        //var now = DateTime.UtcNow;
        var securityKey = new
Microsoft.IdentityModel.Tokens.SymmetricSecurityKey(System.Text.Encoding.Default.GetB
ytes(sec));

        SecurityToken securityToken;
        JwtSecurityTokenHandler handler = new JwtSecurityTokenHandler();
        TokenValidationParameters validationParameters = new
TokenValidationParameters()
        {
            ValidAudience = "pgncms",
            ValidIssuer = "pgncms",
            ValidateLifetime = true,
            ValidateIssuerSigningKey = true,
            LifetimeValidator = this.LifetimeValidator,
            IssuerSigningKey = securityKey
        };

        //extract and assign the user of the jwt
        Thread.CurrentPrincipal = handler.ValidateToken(token, validationParameters, out
securityToken);
        HttpContext.Current.User = handler.ValidateToken(token, validationParameters, out
securityToken);

        return base.SendAsync(request, cancellationTokens);
    }
    catch (SecurityTokenValidationException e)
    {
        statusCode = HttpStatusCode.Unauthorized;
    }
    catch (Exception ex)
    {
        statusCode = HttpStatusCode.InternalServerError;
    }
}

```

```

    //@josbol mentioned this issue in 2018

    //Since this solution pushes the request further down the pipeline, instead of immediately
returning, this
    //solution consumes more resources for each request, so this might not be the right
solution for everybody.
    //Alternatively one could manually add the missing headers before returning,
    //but I don't think that would be a good idea.

    //- this line would return correct CORS header.- credit to
    //return base.SendAsync(request, cancellationToken);

    return Task<HttpResponseMessage>.Factory.StartNew(() => new
HttpResponseMessage(statusCode) { });
}

public bool LifetimeValidator(DateTime? notBefore, DateTime? expires, SecurityToken
securityToken, TokenValidationParameters validationParameters)
{
    //if (expires != null)
    //{
    //    if (DateTime.UtcNow < expires) return true;
    //}
    //return false;
    return true;
}
}
}

```

c. Views

1. Account

```

@{
    Layout = null;
    ViewBag.Title = "HPP Harga Pokok Produksi";
}
<!DOCTYPE html>
<html lang="en">

<!-- begin::Head -->
<head>
    <meta charset="utf-8" />
    <title>@ViewBag.Title</title>
    <meta name="description" content="Latest updates and statistic charts">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1, shrink-to-fit=no">

```

```

<!--begin::Web font -->
<script src="https://ajax.googleapis.com/ajax/libs/webfont/1.6.16/webfont.js"></script>
<script>
  WebFont.load({
    google: {
      "families": ["Poppins:300,400,500,600,700", "Roboto:300,400,500,600,700"]
    },
    active: function () {
      sessionStorage.fonts = true;
    }
  });
</script>

<!--end::Web font -->
<!--begin::Base Styles -->
<link href="~/Content/vendors/base/vendors.bundle.css" rel="stylesheet"
type="text/css" />

<!--RTL version:<link href="~/Content/vendors/base/vendors.bundle.rtl.css"
rel="stylesheet" type="text/css" />-->
<link href="~/Content/demo/default/base/style.bundle.css" rel="stylesheet"
type="text/css" />

<!--RTL version:<link href="~/Content/demo/default/base/style.bundle.rtl.css"
rel="stylesheet" type="text/css" />-->
<!--end::Base Styles -->
@*<link rel="shortcut icon"
href="~/Content/demo/demo12/media/img/logo/favicon.ico" />*@
</head>

<!-- end::Head -->
<!-- begin::Body -->
<body class="m--skin- m--header--fixed m--header--fixed-mobile m--aside-left--enabled m--
aside-left--skin-dark m--aside-left--fixed m--aside-left--offcanvas m--footer--push m--aside--
offcanvas-default">

<!-- begin:: Page -->
<div class="m-grid m-grid--hor m-grid--root m-page">
  <div class="m-grid__item m-grid__item--fluid m-grid m-grid--desktop m-grid--ver-
desktop m-grid--hor-tablet-and-mobile m-login m-login--6" id="m_login">
    <div class="m-grid__item m-grid__item--order-tablet-and-mobile-2 m-grid m-
grid--hor m-login__aside " style="background-image: url(/Content/app/media/img/bg/bg-
1.jpg);">
      <div class="m-grid__item">
        <div class="m-login__logo">
          <a href="@Url.Action("Login", "Account")">

```

```

        
    </a>
</div>
</div>
<div class="m-grid__item m-grid__item--fluid m-grid m-grid--ver">
    <div class="m-grid__item m-grid__item--middle">
        <span class="m-login__title"> Harga Pokok Produksi </span>
        <span class="m-login__subtitle">PT. Citra Bumi Raya - Jombang</span>
    </div>
</div>
<div class="m-grid__item">
    <div class="m-login__info">
        <div class="m-login__section">
            <a href="#" class="m-link">Copyright &copy 2020 PT. Citra Bumi Raya.
All Rights Reserved.</a>
        </div>
        @*<div class="m-login__section">
            <a href="#" class="m-link">Privacy</a>
            <a href="#" class="m-link">Legal</a>
            <a href="#" class="m-link">Contact</a>
        </div>*@
    </div>
</div>
</div>
<div class="m-grid__item m-grid__item--fluid m-grid__item--order-tablet-and-
mobile-1 m-login__wrapper">

<!--begin::Head-->
<div class="m-login__head">
    @*<span>Don't have an account?</span>*@
    @*<a href="#" class="m-link m--font-danger">Sign Up</a>*@
</div>

<!--end::Head-->
<!--begin::Body-->
<div class="m-login__body">

    <!--begin::Signin-->
    <div class="m-login__signin">
        <div class="m-login__title">
            <h3>User Login</h3>
        </div>
        <!--begin::Form-->
            @using (Html.BeginForm("Login", "Account", new { returnUrl =
ViewBag.ReturnUrl }, FormMethod.Post, new { @class = "m-login__form m-form" })))

```

```

    {
        @Html.AntiForgeryToken()
        <span style="color:red">@ViewBag.Error</span>
        <div class="form-group m-form__group">
            <input class="form-control m-input" type="text"
placeholder="Username" id="username" name="username" autocomplete="off" required>
        </div>
        <div class="form-group m-form__group">
            <input class="form-control m-input m-login__form-input--last"
type="password" placeholder="Password" id="password" name="password" required>
        </div>

        <!--end::Form-->

        <!--begin::Action-->
        <div class="m-login__action">
            <button id="submit" type="submit" value="submit" class="btn btn-
primary">Sign In</button>
        </div>
        <!--end::Action-->
    }
</div>

<!--end::Signin-->
</div>

<!--end::Body-->
</div>
</div>
</div>

<!-- end:: Page -->
<!--begin::Base Scripts -->
<script src="~/Content/vendors/base/vendors.bundle.js"
type="text/javascript"></script>
<script src="~/Content/demo/default/base/scripts.bundle.js"
type="text/javascript"></script>
<!--end::Base Scripts -->
<!--begin::Page Snippets -->
<script src="~/Content/snippets/custom/pages/user/login6.js"
type="text/javascript"></script>
<!--end::Page Snippets -->
<script>
    //window.onload = function () { setTimeout(function () { ready(); }, 1000); };

    //function ready() {

```



```

// $(img).attr('src', function (i, src) {
//     return '/PGN' + $(this).attr('src');
// })

//}
</script>
</body>

```

```

<!-- end::Body -->
</html>

```

2. HPP

```

_ModalItem.cshtml
@using CMS.Website.Models.Item;
@model ItemVM
@Html.AntiForgeryToken()
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
<div id="viewData" class="modal fade" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Maintenance Item</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      @Html.HiddenFor(model => model.ID, new { htmlAttributes = new { @class =
"form-control" } })
      @Html.HiddenFor(model => model.CreatedBy, new { htmlAttributes = new {
@class = "form-control" } })
      @Html.HiddenFor(model => model.CreatedDate, new { htmlAttributes = new {
@class = "form-control" } })
      <div class="modal-body">
        <div class="m-portlet__body">
          <div class="row">
            <div class="col-md-12">
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-
label">Kode</label>
                <div class="col-10">
                  @Html.EditorFor(model => model.Kode, new { htmlAttributes =
new { @class = "form-control", @autocomplete = "off" } })
                  @Html.ValidationMessageFor(model => model.Kode, "", new {
@class = "text-danger" })
                </div>
              </div>
            </div>
          </div>
        </div>
      </div>
    </div>
  </div>

```

```

<div class="form-group m-form__group row">
  <label for="example-text-input" class="col-2 col-form-label">Nama
Item</label>
  <div class="col-10">
    @Html.EditorFor(model => model.Nama, new { htmlAttributes =
new { @class = "form-control", @autocomplete = "off" } })
    @Html.ValidationMessageFor(model => model.Nama, "", new {
@class = "text-danger" })
  </div>
</div>
<div class="form-group m-form__group row">
  <label for="example-text-input" class="col-2 col-form-label">Jenis
Biaya</label>
  <div class="col-10">
    @*@Html.EditorFor(model => model.Jenis, new { htmlAttributes =
new { @class = "form-control", @autocomplete = "off" } })*@
    <select class="form-control m-input" id="Jenis" name="Jenis">
      <option value="Biaya Bahan Baku">Biaya Bahan Baku</option>
      <option value="Biaya Tenaga Kerja Langsung">Biaya Tenaga
Kerja Langsung</option>
      <option value="Biaya Permesinan">Biaya Permesinan</option>
      <option value="Biaya Overhead">Biaya Overhead</option>
      <option value="Biaya Lainnya">Biaya Lainnya</option>
    </select>
  </div>
</div>
<div class="form-group m-form__group row">
  <label for="example-text-input" class="col-2 col-form-label">No.
Urut Tampilan</label>
  <div class="col-10">
    @Html.EditorFor(model => model.NoUrut, new { htmlAttributes =
new { @class = "form-control", @autocomplete = "off" } })
    @Html.ValidationMessageFor(model => model.NoUrut, "", new {
@class = "text-danger" })
  </div>
</div>
<div class="form-group m-form__group row" hidden>
  <label for="example-text-input" class="col-2 col-form-
label">Formula</label>
  <div class="col-10">
    @Html.EditorFor(model => model.Formula, new { htmlAttributes =
new { @class = "form-control", @autocomplete = "off" } })
    @Html.ValidationMessageFor(model => model.Formula, "", new {
@class = "text-danger" })
  </div>
</div>

```

```

        <div class="form-group m-form__group row">
            <label for="example-text-input" class="col-2 col-form-
label">Aktif</label>
            <div class="col-10">
                <span class="m-switch m-switch--outline m-switch--icon m-switch-
-succes">
                    <label>
                        <input id="IsActive" name="IsActive" type="checkbox"
checked="checked" value="true">
                            <span></span>
                        </label>
                    </span>
                </div>
            </div>
        </div>
    </div>
    </div>
    </div>
    </div>
    <div class="modal-footer">
        <button id="btnSimpan" type="button" class="btn btn-
primary">Submit</button>
        <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
    </div>
</div>
</div>
</div>
</div>

```

_TableItem.cshtml

```

<div class="row">
    <div class="col-md-12">
        <div class="panel panel-primary list-panel" id="list-panel">
            <div class="panel-heading list-panel-heading">
                <h1 class="panel-title list-panel-title">@ViewBag.Title</h1>
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="col-md-12">
                        <button id="btnTambah" type="reset" class="btn btn-succes"> + Tambah
Item</button>
                    </div>
                    <div class="col-md-12">
                        <table id="DataTable" class="table table-striped table-bordered"></table>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

    </div>
  </div>
</div>

```

3. Item

_General.cshtml

```

<script>
$(document).ready(function () {
    $("#ID").val(0);
    dataTable = $('#DataTable').DataTable({
        serverSide: true,
        processing: true,
        responsive: true,
        ajax: {
            type: "POST",
            url: "@Url.Action("GetData", "Item")",
            dataSrc: function (json) {
                number = json.startRow;
                return json.data;
            }
        },
        columns: [
            {
                title: "No",
                data: null
            },
            {
                title: "No Urut Tampilan",
                data: "NoUrut",
            },
            {
                title: "Kode",
                data: "Kode",
            },
            {
                title: "Nama Item",
                data: "Nama",
            },
            {
                title: "Jenis",
                data: "Jenis",
            },
            // {
            //     title: "Formula",
            //     data: "Formula",
            // },
            {

```

```

        title: "Status",
        data: "IsActive",
        render: function (val, type, row, meta)
        {
            var dataVal = "Tidak Aktif";
            if (val == true) {
                dataVal = "Aktif";
            }
            return dataVal;
        }
    },
    {
        title: "Action",
        data: "ID",
        render: function (val, type, row, meta) {
            var name = ""+row>Nama+"";
            var dataVal = " <button type='button' class='btn btn-primary btn-sm'
value="" + val +"" onclick='View(this.value, true)'> <i class='la la-align-justify'></i>
</button> "+
                " <button type='button' class='btn btn-info btn-sm' value="" + val
+ "" onclick='View(this.value, false)'> <i class='la la-edit'></i> </button> " +
                " <button type='button' class='btn btn-danger btn-sm' value="" +
val + "" onclick='DeleteConfirm(this.value,"+name+" )> <i class='la la-bitbucket-square'></i> </button> ";
            return dataVal;
        },
    },
],
lengthMenu: [[10, 25, 50, 100], [10, 25, 50, 100]],
});

dataTable.on("draw.dt", function () {
    dataTable.column(0, { search: 'applied', order: 'applied' }).nodes().each(function
(cell, i) {
        i = i + number + 1
        cell.innerHTML = i;
    });
}).draw();

$("#btnTambah").click(function ()
{
    $("#form0").closest('form').find("input[type=text], textarea").val("");
    $("#ID").val('00000000-0000-0000-0000-000000000000');
    $("input").prop("disabled", false);
    $("#btnSimpan").removeAttr("hidden");
    $("#IsActive").attr('checked', 'checked');

```

```

        $("#viewData").modal("show");
    });

    $("#btnBatal").click(function () {
        window.location.href = "@Url.Action("Index")";
    });

    $("#btnSimpan").click(function () {
        $("#form0").submit();
    });
});

function View(id, isDisabled)
{
    $.ajax({
        type: "GET",
        url: "@Url.Action("GetById", "Item)",
        data: {ID:id},
        success: function (res) {
            if (res.status.IsSuccess == true) {
                $("input").prop("disabled", isDisabled);
                $("textarea").prop("disabled", isDisabled);
                $("select").prop("disabled", isDisabled);
                if (isDisabled == true) {
                    $("#btnSimpan").hide();
                }
                else {
                    $("#btnSimpan").show();
                }
                SetValue(res.data);
            }
        }
    });

    $("#viewData").modal("show");
}

function DeleteConfirm(id) {
    bootbox.confirm({
        title: 'Confirm',
        message: "Apakah Anda Yakin Hapus Data Ini ?",
        buttons: {
            confirm: {
                label: 'Yes',
                className: 'btn-success'
            },

```



```

    }
    bootbox.dialog({
        title: 'Error',
        message: "Error while processing the request.\r\n" + errors,
        size: 'medium',
        buttons: {
            ok: {
                label: "OK",
                className: 'btn-danger'
            }
        }
    });
}
}

function OnFailure(data) {
    bootbox.alert("Error receiving data", function () {
        console.log(data);
    });
}

function SetValue(data)
{
    $("#ID").val(data.ID);
    $("#Kode").val(data.Kode);
    $("#Nama").val(data>Nama);
    $("#NoUrut").val(data.NoUrut);
    $("#Formula").val(data.Formula);
    //$("#Password").val(data.Password);
    $("#CreatedBy").val(data.CreatedBy);
    var createDate = ConvertJsonDate(data.CreatedDate);
    $("#CreatedDate").val(createDate);
    if (data.IsActive == true) {
        $("#IsActive").attr('checked', true);
    }
    else {
        $("#IsActive").attr('checked', false);
    }
}

function previewImage() {
    document.getElementById("image-preview").style.display = "block";
    var oFReader = new FileReader();
    oFReader.readAsDataURL(document.getElementById("image-source").files[0]);
}

```



```

        oFReader.onload = function (oFREvent) {
            document.getElementById("image-preview").src = oFREvent.target.result;
        };
    };
</script>

```

Index.cshtml

```

@using CMS.Website.Models.Item;
@model ItemVM
@{
    AjaxOptions ajaxOptions = new AjaxOptions
    {
        HttpMethod = "POST",
        OnSuccess = "OnSuccess",
        OnFailure = "OnFailure",
        OnBegin = "AjaxValidate",
        Url = Url.Action("Edit")
    };
}

<div class="m-content">
    <div class="row">
        <div class="col-md-12">

            <div class="m-portlet m-portlet--tab">
                <div class="m-portlet__head">
                    <div class="m-portlet__head-caption">
                        <div class="m-portlet__head-title">
                            <span class="m-portlet__head-icon m--hide">
                                <i class="la la-gear"></i>
                            </span>
                            <h3 class="m-portlet__head-text">
                                Maintenance Item
                            </h3>
                        </div>
                    </div>
                </div>
                <div class="m-portlet__body">
                    <div class="row">
                        <div class="col-md-12">
                            @Html.Partial("~/Views/Item/Component/_TableItem.cshtml")
                        </div>
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```

        </div>

    </div>
</div>
</div>

@using (Ajax.BeginForm(null, null, ajaxOptions, new { @class = "form-horizontal" }))
{
    @Html.Partial("~/Views/Item/Component/_ModalItem.cshtml")
}
@Html.Partial("~/Views/Item/Script/_General.cshtml")

```

4. Shared

```

_Layout.cshtml
@using CMS.DataHelpers;
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <title>HPP CMS</title>
    <meta name="description" content="Latest updates and statistic charts">
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-
scale=1, shrink-to-fit=no">

    <script src="~/Scripts/webfont.js"></script>
    <script>
        WebFont.load({
            google: {
                "families": ["Poppins:300,400,500,600,700", "Roboto:300,400,500,600,700"]
            },
            active: function () {
                sessionStorage.fonts = true;
            }
        });
    </script>

    <!--begin::Web font -->
    <link href="~/Content/vendors/base/vendors.bundle.css" rel="stylesheet" />
    <link href="~/Content/demo/default/base/style.bundle.css" rel="stylesheet" />
    <link
        href="~/Content/vendors/custom/datatables/datatables.bundle.css"
rel="stylesheet" />
    <link href="~/Content/loading.css" rel="stylesheet" />
    <style>
        .image-preview {
            width: 400px;
            height: 250px;

```

```

    }

    .table thead th {
        white-space: nowrap;
    }

    .no-wrap {
        white-space: nowrap;
    }
</style>
</head>
<body class="m-page--fluid m--skin- m-content--skin-light2 m-header--fixed m-header--
fixed-mobile m-aside-left--enabled m-aside-left--skin-dark m-aside-left--offcanvas m-
footer--push m-aside--offcanvas-default">
    <!-- begin:: Page -->
    <div class="loading-">Loading</div>
    <div class="m-grid m-grid--hor m-grid--root m-page">

        <!-- BEGIN: Header -->
        <header id="m_header" class="m-grid__item m-header " m-minimize-offset="200"
m-minimize-mobile-offset="200">
            <div class="m-container m-container--fluid m-container--full-height">
                <div class="m-stack m-stack--ver m-stack--desktop">

                    <!-- BEGIN: Brand -->
                    <div class="m-stack__item m-brand m-brand--skin-dark ">
                        <div class="m-stack m-stack--ver m-stack--general">
                            <div class="m-stack__item m-stack__item--middle m-brand__logo"
style="text-align:center">
                                <a href="javascript:;" class="m-brand__logo-wrapper">
                                    
                                </a>
                            </div>
                        </div>
                    </div>
                </div>

                <!-- END: Brand -->
                <div class="m-stack__item m-stack__item--fluid m-header-head"
id="m_header_nav">

                    <!-- BEGIN: Horizontal Menu -->
                    <button class="m-aside-header-menu-mobile-close m-aside-header-menu-
mobile-close--skin-dark " id="m_aside_header_menu_mobile_close_btn">
                        <i class="la la-close"></i>
                    </button>

```

```

<!-- END: Horizontal Menu -->
<!-- BEGIN: Topbar -->
<div id="m_header_topbar" class="m-topbar m-stack m-stack--ver m-
stack--general">
  <div class="m-stack__item m-topbar__nav-wrapper">
    <ul class="m-topbar__nav m-nav m-nav--inline">
      <span style="vertical-align:center; font-size:medium; text-
align:center">@User.Identity.Name</span>
      <li class="m-nav__item m-topbar__user-profile m-dropdown m-
dropdown--medium m-dropdown--arrow m-dropdown--align-right m-dropdown--mobile-
full-width m-dropdown--skin-light" m-dropdown-toggle="click">
        <a href="javascript:;" class="m-nav__link m-
dropdown__toggle">
          <span class="m-topbar__userpic">
            
          </span>
        </a>
        <div class="m-dropdown__wrapper">
          <span class="m-dropdown__arrow m-dropdown__arrow--right
m-dropdown__arrow--adjust"></span>
          <div class="m-dropdown__inner">
            <div class="m-dropdown__body">
              <div class="m-dropdown__content">
                <ul class="m-nav m-nav--skin-light">
                  <li class="m-nav__section m--hide">
                    <span class="m-nav__section-
text">Section</span>
                  </li>
                  <li class="m-nav__item">
                    <a href="@Url.Action("Logout",
"Account")">Logout</a>
                  </li>
                </ul>
              </div>
            </div>
          </div>
        </li>
      </ul>
    </div>
  </div>
<!-- END: Topbar -->
</div>

```

```

        </div>
    </div>
</header>

<!-- END: Header -->
<!-- begin::Body -->
<div class="m-grid__item m-grid__item--fluid m-grid m-grid--ver-desktop m-grid--
desktop m-body">

    <!-- BEGIN: Left Aside -->
    <button class="m-aside-left-close m-aside-left-close--skin-dark "
id="m_aside_left_close_btn">
        <i class="la la-close"></i>
    </button>
    <div id="m_aside_left" class="m-grid__item m-aside-left m-aside-left--skin-dark
">

        <!-- BEGIN: Aside Menu -->
        <div id="m_ver_menu" class="m-aside-menu m-aside-menu--skin-dark m-
aside-menu--submenu-skin-dark " m-menu-vertical="1" m-menu-scrollable="0" m-menu-
dropdown-timeout="500">
            <ul class="m-menu__nav ">
                <li class="m-menu__section m-menu__section--first">
                    <h4 class="m-menu__section-text">Menu</h4>
                    <i class="m-menu__section-icon flaticon-more-v2"></i>
                </li>
                <li class="m-menu__item @(ViewBag.Menu ==
Enums>NamaMenu.User.ToString() ? "m-menu__item--active" : "")" aria-
haspopup="true">
                    <a href="@Url.Action("index", "User")" class="m-menu__link ">
                        <span class="m-menu__item-here"></span>
                        <i class="m-menu__link-icon flaticon-lifebuoy"></i>
                        <span class="m-menu__link-text">Maintenance User</span>
                    </a>
                </li>
                <li class="m-menu__item @(ViewBag.Menu ==
Enums>NamaMenu.Item.ToString() ? "m-menu__item--active" : "")" aria-
haspopup="true">
                    <a href="@Url.Action("index", "Item")" class="m-menu__link ">
                        <span class="m-menu__item-here"></span>
                        <i class="m-menu__link-icon flaticon-lifebuoy"></i>
                        <span class="m-menu__link-text">Maintenance Item</span>
                    </a>
                </li>
            </ul>
        </div>
    </div>

```



```

<!-- end::Footer -->
<!-- end:: Page -->
<!-- begin::Scroll Top -->
<div id="m_scroll_top" class="m-scroll-top">
  <i class="la la-arrow-up"></i>
</div>

<!-- end::Scroll Top -->
<script src="~/Content/vendors/base/vendors.bundle.js"></script>
<script src="~/Content/demo/default/base/scripts.bundle.js"></script>
<script src="~/Content/demo/default/custom/crud/forms/widgets/bootstrap-datepicker.js" type="text/javascript"></script>
<script src="~/Content/demo/default/custom/crud/forms/widgets/bootstrap-datetimepicker.js" type="text/javascript"></script>
<script src="~/Content/demo/default/custom/crud/forms/widgets/bootstrap-timepicker.js" type="text/javascript"></script>
<script src="~/Content/vendors/custom/datatables/datatables.bundle.js"></script>
<script src="~/Content/vendors/custom/validate/jquery.unobtrusive-ajax.js"></script>
<script src="~/Content/vendors/custom/validate/jquery.validate.js"></script>
<script src="~/Content/vendors/custom/validate/jquery.validate.unobtrusive.js"></script>
<script src="~/Content/vendors/custom/bootbox/bootbox.js"></script>
<script src="~/Content/vendors/custom/bootbox/bootbox.all.js"></script>
<script src="~/Content/vendors/custom/bootbox/bootbox.locales.js"></script>
<link href="~/Content/demo/default/custom/crud/forms/widgets/summernote.css" rel="stylesheet">
<script src="~/Content/demo/default/custom/crud/forms/widgets/summernote.js"></script>
<script>

$( '#faqmodal' ).on( 'shown.bs.modal', function () {
  $( '.note-toolbar-wrapper' ).removeAttr( 'style' );
  $( '.note-toolbar' ).removeAttr( 'style' );
} )
function ConvertJsonDate( date ) {
  var dateString = date.substr( 6 );
  var currentTime = new Date( parseInt( dateString ) );
  var month = currentTime.getMonth() + 1;
  var day = currentTime.getDate();
  var year = currentTime.getFullYear();
  var hour = currentTime.getHours();
  var minute = currentTime.getMinutes();
  var second = currentTime.getSeconds();
  if ( month < 10 ) month = '0' + month;
  if ( day < 10 ) day = '0' + day;
  if ( hour < 10 ) hour = '0' + hour;
}

```

```

    if (minute < 10) minute = '0' + minute;
    if (second < 10) second = '0' + second;
    return year + "-" + month + "-" + day + " " + hour + ":" + minute + ":" + second;
}

```

```

function ConvertJsonDateOnly(date) {
    var dateString = date.substr(6);
    var currentTime = new Date(parseInt(dateString ));
    var month = currentTime.getMonth() + 1;
    var day = currentTime.getDate();
    var year = currentTime.getFullYear();
    if (month < 10) month = '0' + month;
    if (day < 10) day = '0' + day;
    return year + "-" + month + "-" + day;
}

```

```

function CreateGuid() {
    function _p8(s) {
        var p = (Math.random().toString(16)+"000000000").substr(2,8);
        return s ? "-" + p.substr(0,4) + "-" + p.substr(4,4) : p ;
    }
    return _p8() + _p8(true) + _p8(true) + _p8();
}

```

```

function FormatCreatedDate(date) {
    var d = new Date(date),
        month = " " + (d.getMonth() + 1),
        day = " " + d.getDate(),
        year = d.getFullYear();
    hour = d.getHours();
    minute = d.getMinutes();
    second = d.getSeconds();

    if (month < 10) month = '0' + month;
    if (day < 10) day = '0' + day;
    if (hour < 10) hour = '0' + hour;
    if (minute < 10) minute = '0' + minute;
    if (second < 10) second = '0' + second;

    return [year, month, day].join('-') + " " + [hour, minute, second].join(':');
}

```

```

function FormatDateDDMMYY(date) {
    var d = new Date(date),
        month = " " + (d.getMonth() + 1),
        day = " " + d.getDate(),

```



```

        year = d.getFullYear();

        if (month < 10) month = '0' + month;
        if (day < 10) day = '0' + day;

        return [day, month, year].join('/');
    }

function FormatDateYYMMDD(date) {
    var split = date.split("/");
    var d = new Date(split[2], split[1] - 1, split[0]);
        month = " + (d.getMonth() + 1),
        day = " + d.getDate(),
        year = d.getFullYear();

        if (month < 10) month = '0' + month;
        if (day < 10) day = '0' + day;

        return [year, month, day].join('-');
    }

var arrows;
if (mUtil.isRTL()) {
    arrows = {
        leftArrow: '<i class="la la-angle-right"></i>',
        rightArrow: '<i class="la la-angle-left"></i>'
    }
} else {
    arrows = {
        leftArrow: '<i class="la la-angle-left"></i>',
        rightArrow: '<i class="la la-angle-right"></i>'
    }
}

jQuery.validator.methods["date"] = function (value, element) { return true; }

$('.datepicker').datepicker({
    rtl: mUtil.isRTL(),
    todayBtn: "linked",
    clearBtn: true,
    todayHighlight: false,
    templates: arrows,
    format: 'dd/mm/yyyy',
    dateFormat: 'dd/mm/yyyy'
});
$('.datetimepicker').datetimepicker({

```

```

        todayHighlight: true,
        autoclose: true,
        pickerPosition: 'bottom-left',
        format: 'hh:ii dd M yyyy '
    });

    $('timepicker').timepicker({
    disableMousewheel: true,
    icons: {
        up: 'la la-angle-up',
        down: 'la la-angle-down'
    },
    showSeconds: true,
    showMeridian: false,
    defaultTime: new Date()
    }).on('changeTime.timepicker', function (e) {

    var hours = ('0' + e.time.hours).slice(-2);
    var minutes = ('0' + e.time.minutes).slice(-2);
    var seconds = ('0' + e.time.seconds).slice(-2);

    $(this).val(hours + ':' + minutes + ':' + seconds);
    });

    function convertTglJam(str) {
        //var str = "19:15 25 Jan 2021";
        var res = str.split(" ");
        var tgl = res[0].split("/");
        var tglx = [tgl[2], tgl[1], tgl[0]].join('-') + " " + res[1] + ":00";
        return tglx;
    }

    var _URL = window.URL || window.webkitURL;
    var IsDimension = true;

    var path = '@ViewBag.PathFile';
    function DownloadFile(val) {
        window.open(path+val);
    }

    function Currency(input) {
        var ThisVal = $('input[name="' + input.name +"]').val();
        var val = ThisVal.replace(/^[^\d.]/g, "").replace(/(?<!\.*)(\d)(?=(?:\d{3})+(?:\.\d|$))/g,
        '$1,');
        $('input[name="' + input.name +"]').val(val);
    }

```

```

function SetCurrency(input) {
    return
    input.toString().replace(/^[^\d.]/g,
    "").replace(/(?<!\.)*(\d)(?=(?:\d{3})+(?:\.\d?))/g, '$1,');
}

function DownloadAttachment(val) {
    window.open(val);
}
$('.btnCancel').click(function () {
    location.reload();
});

//window.onload = function(){ setTimeout( function(){ ready(); }, 1000); };

//function ready() {
//    $('img').attr('src', function (i, src) {
//        return '/PGN' + $(this).attr('src');
//    })
//}

</script>
</body>
</html>

```

Error.cshtml

```

<!DOCTYPE html>
<html>
<head>
    <meta name="viewport" content="width=device-width" />
    <title>Error</title>
</head>
<body>
    <hgroup>
        <h1>Error.</h1>
        <h2>An error occurred while processing your request.</h2>
    </hgroup>
</body>
</html>

```

5. User

_ModalUser.cshtml

```
@using CMS.Website.Models.User;
```

```
@model MasterUserVM
```

```
@Html.AntiForgeryToken()
```

```
@Html.ValidationSummary(true, "", new { @class = "text-danger" })
```

```

<div id="viewData" class="modal fade" tabindex="-1" role="dialog" aria-
labelledby="exampleModalLabel" aria-hidden="true">
  <div class="modal-dialog modal-lg" role="document">
    <div class="modal-content">
      <div class="modal-header">
        <h5 class="modal-title" id="exampleModalLabel">Maintenance User</h5>
        <button type="button" class="close" data-dismiss="modal" aria-label="Close">
          <span aria-hidden="true">&times;</span>
        </button>
      </div>
      @Html.HiddenFor(model => model.ID, new { htmlAttributes = new { @class =
"form-control" } })
      @Html.HiddenFor(model => model.CreatedBy, new { htmlAttributes = new {
@class = "form-control" } })
      @Html.HiddenFor(model => model.CreatedDate, new { htmlAttributes = new {
@class = "form-control" } })
      <div class="modal-body">
        <div class="m-portlet__body">
          <div class="row">
            <div class="col-md-12">
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-
label">Username</label>
                <div class="col-10">
                  @Html.EditorFor(model => model.Username, new { htmlAttributes
= new { @class = "form-control", @autocomplete = "off" } })
                  @Html.ValidationMessageFor(model => model.Username, "", new
{ @class = "text-danger" })
                </div>
              </div>
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-
label">Password</label>
                <div class="col-10">
                  @Html.EditorFor(model => model.Password, new { htmlAttributes
= new { @class = "form-control", @autocomplete = "off", @Type="password" } })
                  @Html.ValidationMessageFor(model => model.Password, "", new {
@class = "text-danger" })
                </div>
              </div>
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-
label">Aktif</label>
                <div class="col-10">
                  <span class="m-switch m-switch--outline m-switch--icon m-switch-
-success">

```

```

        <label>
            <input id="IsActive" name="IsActive" type="checkbox"
checked="checked" value="true">
            <span></span>
        </label>
    </span>
</div>
</div>
</div>
</div>
</div>
</div>
</div>
<div class="modal-footer">
    <button id="btnSimpan" type="button" class="btn btn-
primary">Submit</button>
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
</div>
</div>
</div>
</div>
</div>

```

_TableUser.cshtml

```

<div class="row">
    <div class="col-md-12">
        <div class="panel panel-primary list-panel" id="list-panel">
            <div class="panel-heading list-panel-heading">
                <h1 class="panel-title list-panel-title">@ViewBag.Title</h1>
            </div>
            <div class="panel-body">
                <div class="row">
                    <div class="col-md-12">
                        <button id="btnTambah" type="reset" class="btn btn-success"> + Tambah
User</button>
                    </div>
                    <div class="col-md-12">
                        <table id="DataTable" class="table table-striped table-bordered"></table>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
</div>
</div>

```

_General.cshtml

```

<script>

```

```

$(document).ready(function () {
    $("#ID").val(0);
    dataTable = $('#DataTable').DataTable({
        serverSide: true,
        processing: true,
        responsive: true,
        ajax: {
            type: "POST",
            url: "@Url.Action("GetData", "User")",
            dataSrc: function (json) {
                number = json.startRow;
                return json.data;
            }
        },
        columns: [
            {
                title: "No",
                data: null
            },
            {
                title: "Username",
                data: "Username",
            },
            {
                title: "Status",
                data: "IsActive",
                render: function (val, type, row, meta)
                {
                    var dataVal = "Tidak Aktif";
                    if (val == true) {
                        dataVal = "Aktif";
                    }
                    return dataVal;
                }
            },
            {
                title: "Action",
                data: "ID",
                render: function (val, type, row, meta) {
                    var name = ""+row.Username+"";
                    var dataVal = " <button type='button' class='btn btn-primary btn-sm'
value="" + val +"" onclick='View(this.value, true)'> <i class='la la-align-justify'></i>
</button> "+
                    " <button type='button' class='btn btn-info btn-sm' value="" + val
+ "" onclick='View(this.value, false)'> <i class='la la-edit'></i> </button> " +

```

```

        " <button type='button' class='btn btn-danger btn-sm' value='' +
val + " onclick='DeleteConfirm(this.value, "+name+" )'> <i class='la la-bitbucket-
square'></i> </button> ";
        return dataVal;
    },
    ],
    lengthMenu: [[10, 25, 50, 100], [10, 25, 50, 100]],
});

dataTable.on("draw.dt", function () {
    dataTable.column(0, { search: 'applied', order: 'applied' }).nodes().each(function
(cell, i) {
        i = i + number + 1
        cell.innerHTML = i;
    });
}).draw();

$("#btnTambah").click(function ()
{
    $("#form0").closest('form').find("input[type=text], textarea").val("");
    $("#ID").val('00000000-0000-0000-0000-000000000000');
    $("input").prop("disabled", false);
    $("#btnSimpan").removeAttr("hidden");
    $("#IsActive").attr('checked', 'checked');
    $("#viewData").modal("show");
});

$("#btnBatal").click(function () {
    window.location.href = "@Url.Action("Index")";
});

$("#btnSimpan").click(function () {
    $("#form0").submit();
});
});

function View(id, isDisabled)
{
    $.ajax({
        type: "GET",
        url: "@Url.Action("GetById", "User")",
        data: {ID:id},
        success: function (res) {
            if (res.status.IsSuccess == true) {
                $("input").prop("disabled", isDisabled);
            }
        }
    });
}

```

```

        $(".textarea").prop("disabled", isDisabled);
        $(".select").prop("disabled", isDisabled);
    if (isDisabled == true) {
        $("#btnSimpan").hide();
    }
    else {
        $("#btnSimpan").show();
    }
        SetValue(res.data);
    }
}
});

$("#viewData").modal("show");
}

function DeleteConfirm(id) {
    bootbox.confirm({
        title: 'Confirm',
        message: "Apakah Anda Yakin Hapus Data Ini ?",
        buttons: {
            confirm: {
                label: 'Yes',
                className: 'btn-success'
            },
            cancel: {
                label: 'No',
                className: 'btn-danger'
            }
        },
        callback: function (result) {
            if (result) {
                $.ajax({
                    type: "POST",
                    url: "@Url.Action("Delete", "User")",
                    data: {ID:id},
                    success: function (res) {
                        if (res.Success == true) {
                            window.location.href = "@Url.Action("Index")";
                        }
                        else {
                            bootbox.dialog({
                                title: 'Error',
                                message: res.Errors[0],
                                size: 'medium',
                                buttons: {

```



```

function SetValue(data)
{
    $("#ID").val(data.ID);
    $("#Username").val(data.Username);
    //$("#Password").val(data.Password);
    $("#CreatedBy").val(data.CreatedBy);
    var createDate = ConvertJsonDate(data.CreatedDate);
    $("#CreateDate").val(createDate);
    if (data.IsActive == true) {
        $("#IsActive").attr('checked', true);
    }
    else {
        $("#IsActive").attr('checked', false);
    }
}

function previewImage() {
    document.getElementById("image-preview").style.display = "block";
    var oFReader = new FileReader();
    oFReader.readAsDataURL(document.getElementById("image-source").files[0]);

    oFReader.onload = function (oFREvent) {
        document.getElementById("image-preview").src = oFREvent.target.result;
    };
};
</script>

```

Index.cshtml

```

@using CMS.Website.Models.User;
@model MasterUserVM
@{
    AjaxOptions ajaxOptions = new AjaxOptions
    {
        HttpMethod = "POST",
        OnSuccess = "OnSuccess",
        OnFailure = "OnFailure",
        OnBegin = "AjaxValidate",
        Url = Url.Action("Edit")
    };
}

```

```

<div class="m-content">
    <div class="row">
        <div class="col-md-12">

            <div class="m-portlet m-portlet--tab">

```

```

<div class="m-portlet__head">
  <div class="m-portlet__head-caption">
    <div class="m-portlet__head-title">
      <span class="m-portlet__head-icon m--hide">
        <i class="la la-gear"></i>
      </span>
      <h3 class="m-portlet__head-text">
        Maintenance User
      </h3>
    </div>
  </div>
</div>
</div>

<div class="m-portlet__body">
  <div class="row">
    <div class="col-md-12">
      @Html.Partial("~/Views/User/Component/_TableUser.cshtml")
    </div>
  </div>

</div>
</div>

</div>
</div>
</div>

@using (Ajax.BeginForm(null, null, ajaxOptions, new { @class = "form-horizontal" }))
{
  @Html.Partial("~/Views/User/Component/_ModalUser.cshtml")
}
@Html.Partial("~/Views/User/Script/_General.cshtml")

```

6. _ViewStart.cshtml

```
@{
  Layout = "~/Views/Shared/_Layout.cshtml";
}
```

7. Index.cshtml

```
<style>
  #image-preview {
    display: none;
    width: 300px;
    height: 250px;
  }
</style>
```

```

<div class="m-content">
  <div class="row">
    <div class="col-md-12">

      <div class="m-portlet m-portlet--tab">
        <div class="m-portlet__head">
          <div class="m-portlet__head-caption">
            <div class="m-portlet__head-title">
              <span class="m-portlet__head-icon m--hide">
                <i class="la la-gear"></i>
              </span>
              <h3 class="m-portlet__head-text">
                Investor Home
              </h3>
            </div>
          </div>
        </div>
      </div>

      <form class="m-form m-form--fit m-form--label-align-right">
        <div class="m-portlet__body">
          <div class="row">
            <div class="col-md-12">
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-label">Nama
Menu</label>
                <div class="col-10">
                  <input class="form-control m-input" type="text" value=""
id="example-text-input">
                </div>
              </div>
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-label">Nama
Side Menu</label>
                <div class="col-10">
                  <input class="form-control m-input" type="text" value=""
id="example-text-input">
                </div>
              </div>
              <div class="form-group m-form__group row">
                <label for="example-text-input" class="col-2 col-form-
label">Title</label>
                <div class="col-10">
                  <input class="form-control m-input" type="text" value=""
id="example-text-input">
                </div>
              </div>
            </div>
          </div>
        </div>
      </form>
    </div>
  </div>
</div>

```

```

        <div class="form-group m-form__group row">
            <label for="example-text-input" class="col-2 col-form-
label">Konten</label>
            <div class="col-10">
                <textarea class="summernote" id="exampleTextarea"
rows="5"></textarea>
            </div>
        </div>
        <div class="form-group m-form__group row">
            <label for="example-text-input" class="col-2 col-form-
label">URL</label>
            <div class="col-10">
                <input class="form-control m-input" type="text" value=""
id="example-text-input">
            </div>
        </div>
        <div class="form-group m-form__group row">
            <label for="example-text-input" class="col-2 col-form-label">Slider
1</label>
            <div class="col-10">
                <div class="m-widget5__pic">
                    
                </div>
            </div>
        </div>
        <div class="form-group m-form__group row">
            <label for="example-text-input" class="col-2 col-form-
label">&nbsp;</label>
            <div class="col-4">
                <select class="form-control m-input" id="exampleSelect1">
                    <option>1</option>
                    <option>2</option>
                    <option>3</option>
                    <option>4</option>
                    <option>5</option>
                </select>
            </div>
        </div>
        <div class="form-group m-form__group row">
            <label for="example-text-input" class="col-2 col-form-label">Slider
2</label>
            <div class="col-10">
                <div class="m-widget5__pic">
                    

```

```

        </div>
    </div>
</div>
<div class="form-group m-form__group row">
    <label for="example-text-input" class="col-2 col-form-
label">&nbsp;</label>
    <div class="col-4">
        <select class="form-control m-input" id="exampleSelect1">
            <option>1</option>
            <option>2</option>
            <option>3</option>
            <option>4</option>
            <option>5</option>
        </select>
    </div>
</div>
<div class="form-group m-form__group row">
    <label for="example-text-input" class="col-2 col-form-label">Slider
3</label>
    <div class="col-10">
        <div class="m-widget5__pic">
            
        </div>
    </div>
</div>
<div class="form-group m-form__group row">
    <label for="example-text-input" class="col-2 col-form-
label">&nbsp;</label>
    <div class="col-4">
        <select class="form-control m-input" id="exampleSelect1">
            <option>1</option>
            <option>2</option>
            <option>3</option>
            <option>4</option>
            <option>5</option>
        </select>
    </div>
</div>
<div class="form-group m-form__group row">
    <label for="example-text-input" class="col-2 col-form-label">Foto
1</label>
    <div class="col-10">
        <img id="image-preview">
        <br />

```

```

        <input                type="file"                id="image-source"
onchange="previewImage();" />
    </div>
</div>
<div class="form-group m-form__group row">
    <label    for="example-text-input"    class="col-2    col-form-
label">Caption Foto 1</label>
    <div class="col-10">
        <input class="form-control m-input" type="text" value=""
id="example-text-input">
    </div>
</div>
<div class="form-group m-form__group row">
    <label for="example-text-input" class="col-2 col-form-label">Foto
2</label>
    <div class="col-10">
        <img id="image-preview2">
        <br />
        <input                type="file"                id="image-source2"
onchange="previewImage2();" />
    </div>
</div>
<div class="form-group m-form__group row">
    <label    for="example-text-input"    class="col-2    col-form-
label">Caption Foto 2</label>
    <div class="col-10">
        <input class="form-control m-input" type="text" value=""
id="example-text-input">
    </div>
</div>
<div class="form-group m-form__group row">
    <label    for="example-text-input"    class="col-2    col-form-
label">Aktif</label>
    <div class="col-10">
        <span class="m-switch m-switch--outline m-switch--icon m-
switch--success">
            <label>
                <input type="checkbox" checked="checked" name="">
                <span></span>
            </label>
        </span>
    </div>
</div>
</div>

```

```

<div class="col-md-6">

</div>

<div class="col-md-12">
  <div class="m-content">
    <ul class="nav nav-tabs" role="tablist">
      <li class="nav-item">
        <a class="nav-link active" data-toggle="tab" href="#" data-
target="#news">Recent News</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" data-toggle="tab" href="#" data-
target="#events">Upcoming Events</a>
      </li>
      <li class="nav-item">
        <a class="nav-link" data-toggle="tab" href="#" data-
target="#new">What's New</a>
      </li>
    </ul>
    <div class="tab-content">
      <div class="tab-pane active" style="padding-left:20px;padding-
right:20px;" id="news" role="tabpanel">
        <div class="row">
          <button type="reset" class="btn btn-sm btn-success" data-
toggle="modal" data-target="#shareholdermodal"> + Tambah</button>
        </div>
        <div class="row">
          &nbsp;
        </div>
        <div class="row">
          <table class="table table-striped m-table m-table--head-bg-
info">
            <thead>
              <tr>
                <th width="10">No</th>
                <th width="*">Nama File</th>
                <th width="150">Tanggal</th>
                <th width="100">Status</th>
                <th width="150">Action</th>
              </tr>
            </thead>
            <tbody>
              <tr>
                <th scope="row">1</th>

```



```

<td>Laporan Bulanan Registrasi Pemegang
Efek</td>
<td>October 06, 2020</td>
<td>Aktif</td>
<td>
<button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
<button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
<button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
</td>
</tr>
<tr>
<th scope="row">2</th>
<td>Laporan Bulanan Registrasi Pemegang
Efek</td>
<td>October 06, 2020</td>
<td>Aktif</td>
<td>
<button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
<button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
<button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
</td>
</tr>
<tr>
<th scope="row">3</th>
<td>Laporan Bulanan Registrasi Pemegang
Efek</td>
<td>October 06, 2020</td>
<td>Aktif</td>
<td>
<button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
<button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
<button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
</td>
</tr>
</tbody>
</table>
</div>

```

```

</div>
<div class="tab-pane" style="padding-left:20px;padding-
right:20px;" id="events" role="tabpanel">
  <div class="row">
    <button type="reset" class="btn btn-sm btn-success" data-
toggle="modal" data-target="#shareholdermodal"> + Tambah</button>
  </div>
  <div class="row">
    &nbsp;
  </div>
  <div class="row">
    <table class="table table-striped m-table m-table--head-bg-
info">
      <thead>
        <tr>
          <th width="10">No</th>
          <th width="*">Event</th>
          <th width="150">Tanggal</th>
          <th width="100">Status</th>
          <th width="150">Action</th>
        </tr>
      </thead>
      <tbody>
        <tr>
          <th scope="row">1</th>
          <td>Test Event</td>
          <td>October 06, 2020</td>
          <td>Aktif</td>
          <td>
            <button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
            <button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
            <button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
          </td>
        </tr>
        <tr>
          <th scope="row">2</th>
          <td>Test Event</td>
          <td>October 06, 2020</td>
          <td>Aktif</td>
          <td>
            <button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>

```

```

                <button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
                <button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
            </td>
        </tr>
        <tr>
            <th scope="row">3</th>
            <td>Test Event</td>
            <td>October 06, 2020</td>
            <td>Aktif</td>
            <td>
                <button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
                <button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
                <button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
            </td>
        </tr>
    </tbody>
</table>
</div>
</div>
<div class="tab-pane" style="padding-left:20px;padding-
right:20px;" id="new" role="tabpanel">
    <div class="row">
        <button type="reset" class="btn btn-sm btn-success" data-
toggle="modal" data-target="#shareholdermodal"> + Tambah</button>
    </div>
    <div class="row">
        &nbsp;
    </div>
    <div class="row">
        <table class="table table-striped m-table m-table--head-bg-
info">
            <thead>
                <tr>
                    <th width="10">No</th>
                    <th width="*">Nama File</th>
                    <th width="150">Tanggal</th>
                    <th width="100">Status</th>
                    <th width="150">Action</th>
                </tr>
            </thead>
            <tbody>

```

```

<tr>
  <th scope="row">1</th>
  <td>PGAS Business Presentation 9M-2020</td>
  <td>October 06, 2020</td>
  <td>Aktif</td>
  <td>
    <button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
    <button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
    <button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
  </td>
</tr>
<tr>
  <th scope="row">2</th>
  <td>9M-2020 Consolidated Financial Statement</td>
  <td>October 06, 2020</td>
  <td>Aktif</td>
  <td>
    <button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
    <button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
    <button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
  </td>
</tr>
<tr>
  <th scope="row">3</th>
  <td>PGAS Operational Highlight September
2020</td>
  <td>October 06, 2020</td>
  <td>Aktif</td>
  <td>
    <button type="button" class="btn btn-primary btn-
sm"><i class="la la-align-justify"></i></button>
    <button type="button" class="btn btn-info btn-
sm"><i class="la la-edit"></i></button>
    <button type="button" class="btn btn-danger btn-
sm"><i class="la la-bitbucket-square"></i></button>
  </td>
</tr>
</tbody>
</table>
</div>

```



```

<div class="row">
  <div class="col-md-12">
    <div class="form-group m-form__group row">
      <label for="example-text-input" class="col-2 col-form-label">Title
Laporan Shareholder</label>
      <div class="col-10">
        <input class="form-control m-input" type="text" value=""
id="example-text-input">
      </div>
    </div>
    <div class="form-group m-form__group row">
      <label for="example-text-input" class="col-2 col-form-
label">Tanggal</label>
      <div class="col-10">
        <div class="input-group date">
          <input type="text" class="form-control m-input" readonly
value="05/20/2017" id="m_datepicker_3" />
          <div class="input-group-append">
            <span class="input-group-text">
              <i class="la la-calendar"></i>
            </span>
          </div>
        </div>
      </div>
    </div>
    <div class="form-group m-form__group row">
      <label for="example-text-input" class="col-2 col-form-
label">Aktif</label>
      <div class="col-10">
        <span class="m-switch m-switch--outline m-switch--icon m-
switch--success">
          <label>
            <input type="checkbox" checked="checked" name="">
            <span></span>
          </label>
        </span>
      </div>
    </div>
  </div>
</div>
<div class="col-md-12">
  <div class="m-content">
    <div class="row">
      <button type="reset" class="btn btn-sm btn-success" data-
toggle="modal" data-target="#valueshareholdermodal"> + Tambah</button>
    </div>
  </div>
</div>

```

```

<div class="row">
  &nbsp;
</div>
<div class="row">
  <table class="table table-striped m-table m-table--head-bg-info">
    <thead>
      <tr>
        <th>No</th>
        <th>Nama</th>
        <th>Posisi</th>
        <th>Jumlah Saham</th>
        <th>Percentage</th>
        <th>Status</th>
        <th>Action</th>
      </tr>
    </thead>
    <tbody>
      <tr>
        <th scope="row">1</th>
        <td>Arcandra Tahar</td>
        <td>President Commissioner</td>
        <td width="50px">0</td>
        <td width="50px">0%</td>
        <td width="50px">Aktif</td>
        <td width="150px">
          <button type="button" class="btn btn-primary btn-sm"><i class="la la-align-justify"></i></button>
          <button type="button" class="btn btn-info btn-sm"><i class="la la-edit"></i></button>
          <button type="button" class="btn btn-danger btn-sm"><i class="la la-bitbucket-square"></i></button>
        </td>
      </tr>
      <tr>
        <th scope="row">2</th>
        <td>Luky Alfirman</td>
        <td>Commisioners</td>
        <td width="50px">0</td>
        <td width="50px">0%</td>
        <td width="50px">Aktif</td>
        <td width="150px">
          <button type="button" class="btn btn-primary btn-sm"><i class="la la-align-justify"></i></button>
          <button type="button" class="btn btn-info btn-sm"><i class="la la-edit"></i></button>

```



```

        </button>
    </div>
    <div class="modal-body">
        <form>
            <div class="m-portlet__body">
                <div class="row">
                    <div class="col-md-12">
                        <div class="form-group m-form__group row">
                            <label for="example-text-input" class="col-2 col-form-
label">Nama</label>
                            <div class="col-10">
                                <input class="form-control m-input" type="text" value=""
id="example-text-input">
                            </div>
                        </div>
                    </div>
                    <div class="form-group m-form__group row">
                        <label for="example-text-input" class="col-2 col-form-
label">Posisi</label>
                        <div class="col-10">
                            <input class="form-control m-input" type="text" value=""
id="example-text-input">
                        </div>
                    </div>
                    <div class="form-group m-form__group row">
                        <label for="example-text-input" class="col-2 col-form-
label">Jumlah Saham</label>
                        <div class="col-4">
                            <input class="form-control m-input" type="text" value=""
id="example-text-input">
                        </div>
                        <label for="example-text-input" class="col-2 col-form-
label">Percentage</label>
                        <div class="col-3">
                            <input class="form-control m-input" type="text" value=""
id="example-text-input">
                        </div>
                        <label for="example-text-input" class="col-1 col-form-
label">%</label>
                    </div>
                    <div class="form-group m-form__group row">
                        <label for="example-text-input" class="col-2 col-form-
label">Aktif</label>
                        <div class="col-10">
                            <span class="m-switch m-switch--outline m-switch--icon m-
switch--success">
                                <label>

```

```

        <input type="checkbox" checked="checked" name="">
        <span></span>
    </label>
</span>
</div>
</div>
</div>
</div>

</div>
</div>
</form>
</div>
<div class="modal-footer">
    <button type="button" class="btn btn-primary">Save</button>
    <button type="button" class="btn btn-secondary" data-
dismiss="modal">Cancel</button>
</div>
</div>
</div>
</div>

```

```

<script>
function previewImage() {
    document.getElementById("image-preview").style.display = "block";
    var oFReader = new FileReader();
    oFReader.readAsDataURL(document.getElementById("image-source").files[0]);

    oFReader.onload = function (oFREvent) {
        document.getElementById("image-preview").src = oFREvent.target.result;
    };
};

function previewImage2() {
    document.getElementById("image-preview2").style.display = "block";
    var oFReader = new FileReader();
    oFReader.readAsDataURL(document.getElementById("image-source2").files[0]);

    oFReader.onload = function (oFREvent) {
        document.getElementById("image-preview2").src = oFREvent.target.result;
    };
};
</script>

```

8. Web.config

```

<?xml version="1.0"?>

<configuration>
    <configSections>

```

```

    <sectionGroup name="system.web.webPages.razor"
type="System.Web.WebPages.Razor.Configuration.RazorWebSectionGroup,
System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35">
    <section name="host"
type="System.Web.WebPages.Razor.Configuration.HostSection,
System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false" />
    <section name="pages"
type="System.Web.WebPages.Razor.Configuration.RazorPagesSection,
System.Web.WebPages.Razor, Version=3.0.0.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" requirePermission="false" />
    </sectionGroup>
</configSections>

<system.web.webPages.razor>
  <host factoryType="System.Web.Mvc.MvcWebRazorHostFactory, System.Web.Mvc,
Version=5.2.7.0, Culture=neutral, PublicKeyToken=31BF3856AD364E35" />
  <pages pageBaseType="System.Web.Mvc.WebViewPage">
    <namespaces>
      <add namespace="System.Web.Mvc" />
      <add namespace="System.Web.Mvc.Ajax" />
      <add namespace="System.Web.Mvc.Html" />
      <add namespace="System.Web.Optimization"/>
      <add namespace="System.Web.Routing" />
      <add namespace="CMS.Website" />
    </namespaces>
  </pages>
</system.web.webPages.razor>

<appSettings>
  <add key="webpages:Enabled" value="false" />
</appSettings>

<system.webServer>
  <handlers>
    <remove name="BlockViewHandler"/>
    <add name="BlockViewHandler" path="*" verb="*" preCondition="integratedMode"
type="System.Web.HttpNotFoundHandler" />
  </handlers>
</system.webServer>

<system.web>
  <compilation>
    <assemblies>

```

```
<add assembly="System.Web.Mvc, Version=5.2.7.0, Culture=neutral,
PublicKeyToken=31BF3856AD364E35" />
</assemblies>
</compilation>
</system.web>
</configuration>
```