

Lampiran

BarangsController

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using inventori.Models;
using System.IO;
using inventori.Laporan;
using PagedList;
using PagedList.Mvc;

namespace inventori.Controllers
{
    [Authorize]
    public class BarangsController : Controller
    {
        private InvenEntities db = new
        InvenEntities();

        //GET: Barangs
        public ActionResult Index(string
        searchString, int? Page, string
        Kategori_Barang_Fk, string Merk_Fk)
        {
            int page = Page ?? 1;
            ViewBag.Kategori_Barang_Fk = new
            SelectList(db.Kategori_Barang,
            "Kategori_Barang_id",
            "Nama_Kategori_Barang");
            ViewBag.Merk_Fk = new
            SelectList(db.Merk, "Merk_id",
            "Nama_Merk");
            var barangs = db.Barang/*.Include(b =>
            b.Merk.Kategori_Barang.Nama_Kategori_Bara
            ng)*/.Include(b => b.Merk);
            DateTime tgl = DateTime.Now;
            try
            {
                tgl =
                Convert.ToDateTime(searchString);
            }
            catch (Exception e)
            {
                e.ToString();
                if (!string.IsNullOrEmpty(Merk_Fk))
                {
                    int sa =
                    Convert.ToInt32(Merk_Fk);
                    List<Barang> plist =
                    db.Barang.Where(x => x.Merk_Fk ==
                    sa).ToList();
                    return View(plist.OrderBy(x =>
                    x.Barang_id).ToPagedList(page, 5));
                }
            }
        }

        //dibawah pencarian data menu search
        if (!string.IsNullOrEmpty(searchString))
        {
            barangs = barangs.Where(s =>
            s>Nama_Barang.Contains(searchString) ||

            s.Merk>Nama_Merk.Contains(searchString) ||

            s.Merk.Kategori_Barang>Nama_Kategori_Barang.Cont
            ains(searchString)
            );
        }
        if (!string.IsNullOrEmpty(Kategori_Barang_Fk)
        && (!string.IsNullOrEmpty(Merk_Fk)))
        {
            int b =
            Convert.ToInt32(Kategori_Barang_Fk);
            int l = Convert.ToInt32(Merk_Fk);
            List<Barang> plist = db.Barang.Where(x =>
            x.Kategori_Barang_Fk == b && x.Merk_Fk ==
            l).ToList();
            return View(plist.OrderBy(x =>
            x.Barang_id).ToPagedList(page, 5));
        }
        else
        {
            if
            (!string.IsNullOrEmpty(Kategori_Barang_Fk))
            {
                int sa =
                Convert.ToInt32(Kategori_Barang_Fk);
                List<Barang> plist = db.Barang.Where(x
                => x.Kategori_Barang_Fk == sa).ToList();
                return View(plist.OrderBy(x =>
                x.Barang_id).ToPagedList(page, 5));
            }
        }

        public ActionResult Laporan(DateTime?
        fromDate, DateTime? toDate)
        {
            //Laporan Barang();
            if (!fromDate.HasValue) fromDate =
            DateTime.Now.Date;
            if (!toDate.HasValue) toDate =
            fromDate.GetValueOrDefault(DateTime.Now.Date).Da
            te.AddDays(1);
            if (toDate < fromDate) toDate =
            fromDate.GetValueOrDefault(DateTime.Now.Date).Da
            te.AddDays(1);
            ViewBag.fromDate = fromDate;
            ViewBag.toDate = toDate;
            List<M_Laporanbarang> barangList = new
            List<M_Laporanbarang> ();
        }
    }
}
```

```

        return View(barangs.OrderBy(x =>
x.Barang_id).ToPagedList(page, 5));
    }
}

//Filter data atau dropdownlist di index
barang

public ActionResult
LaporanBarang(DateTime? fromDate,
DateTime? toDate, int? Page)
{
    var barangs = db.Barang.Include(b =>
b.Merk);
    if (!fromDate.HasValue) fromDate =
DateTime.Now.Date;
    if (!toDate.HasValue) toDate =
fromDate.GetValueOrDefault(DateTime.Now.
Date).Date.AddDays(1);
    if (toDate < fromDate) toDate =
fromDate.GetValueOrDefault(DateTime.Now.
Date).Date.AddDays(1);
    ViewBag.fromDate = fromDate;
    ViewBag.toDate = toDate;
    {
        var barangss = db.Barang.Include(b
=> b.Merk).Where(c => c.Tanggal >=
fromDate && c.Tanggal < toDate).ToList();
        return View(barangss);
    }
}
ReportBarang rb = new ReportBarang();
rb.Load();
rb.SetDataSource(barangList);
Stream CetakLaporan =
rb.ExportToStream(CrystalDecisions.Shared.E
xportFormatType.PortableDocFormat);

return File(CetakLaporan,
"application/pdf");
}
// GET: Barangs/Details/5
public ActionResult Details(int? id)
{
    if (id == null)
    {
        return new
 HttpStatusCodeResult(HttpStatusCode.BadReq
uest);
    }
    Barang barang = db.Barang.Find(id);
    if (barang == null)
    {
        return HttpNotFound();
    }
    return View(barang);
}

// GET: Barangs/Create

var barangss = db.Barang.Include(b =>
b.Merk).Where(b => b.Tanggal >= fromDate &&
b.Tanggal < toDate).ToList();
foreach (var item in barangss)
{
    barangList.Add(new M_Laporanbarang
    {
        Barang_id = item.Barang_id,
        Kode_Barang = item.Kode_Barang,
        Nama_Barang = item.Nama_Barang,
        Merk_Fk = item.Merk.Nama_Merk,
        Kategori_Barang_Fk =
item.Merk.Kategori_Barang.Nama_Kategori_Barang,
        Jumlah = item.Jumlah == null ? "0" :
item.Jumlah.ToString(),
        Tanggal = item.Tanggal == null ?
DateTime.Now : Convert.ToDateTime(item.Tanggal),
        Foto = item.Foto
    });
}
db.Configuration.ProxyCreationEnabled = false;
List<Merk> ML = db.Merk.Where(x =>
x.Kategori_Fk == Merk_Fk).ToList();
return Json(ML,
 JsonRequestBehavior.AllowGet);
}
// POST: Barangs/Create
// To protect from overposting attacks, please
enable the specific properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
[HttpPost]
[ValidateAntiForgeryToken]
/*CODING UNTUK MENAMBAHKAN
FOTO*/
public ActionResult Create([Bind(Include =
"Barang_id,Kategori_Barang_Fk,Merk_Fk,Status_Fk,
Nama_Barang,Foto,Jumlah,Tanggal")] Barang barang,
HttpPostedFileBase Foto1)
{
    if (ModelState.IsValid)
    {
        var db = new InvenEntities();
        /*CODING UNTUK MENAMBAHKAN FOTO */
        if (Foto1 != null)
        {
            barang.Foto = new
byte[Foto1.ContentLength];
            Foto1.InputStream.Read(barang.Foto, 0,
Foto1.ContentLength);
        }
        //coding untuk kode barang
        DateTime Date = barang.Tanggal ??
DateTime.Now;
        var Nama_Barang = barang.Nama_Barang;
        string firstFiveChar = new
string(Nama_Barang.Take(4).ToArray());
        var cek_barang = db.Barang.Where(a =>
a.Tanggal == "Kategori_Barang_id",
>Nama_Kategori_Barang",
barang.Kategori_Barang_Fk);

```

//CODING UNTUK MEMBUT
 DROPDOWN SECR OTOMATIS PADA
 SAAT PEMILIHAN KATEGORI DAN
 MERK

```

    public ActionResult Create()
    {
        ViewBag.Kategori_Barang_Fk = new
        SelectList(db.Kategori_Barang,
        "Kategori_Barang_id",
        "Nama_Kategori_Barang");
        ViewBag.Status_Fk = new
        SelectList(db.Status_Barang,
        "Status_Barang_id", "Nama_Status_Barang");
        return View();
    }
    public JsonResult GetMerkList(int
    Merk_Fk)
    {
        barang.Tanggal && a>Nama_Barang ==
        barang>Nama_Barang);
        var count_barang =
        cek_barang.Count();
        if (count_barang > 0)
        {
            var angka_akhir = count_barang +
            1;
            barang.Kode_Barang =
            firstFiveChar + Date.ToString("-dd") +
            Date.ToString("MM") + Date.ToString("yy-")
            + angka_akhir;
        }
        else
        {
            barang.Kode_Barang =
            firstFiveChar + Date.ToString("-dd") +
            Date.ToString("MM") + Date.ToString("yy-")
            + "1";
        }
        db.Barang.Add(barang);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

    ViewBag.Merk_Fk = new
    SelectList(db.Merk, "Merk_id", "Nama_Merk",
    barang.Merk_Fk);
    return View(barang);
}

// GET: Barang/Edit/5
public ActionResult Edit(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.BadReq
        uest);
    }
    Barang barang = db.Barang.Find(id);
    if (barang == null)
    {

```

```

        ViewBag.Merk_Fk = new SelectList(db.Merk,
        "Merk_id", "Nama_Merk", barang.Merk_Fk);
        return View(barang);
    }

    // POST: Barang/Edit/5
    // To protect from overposting attacks, please
    enable the specific properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId=317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    //Coding dibawah untuk Edit Foto
    public ActionResult Edit([Bind(Include =
    "Barang_id,Kategori_Barang_Fk,Merk_Fk,Status_Fk,
    Nama_Barang,Jumlah,Tanggal")] Barang barang,
    HttpPostedFileBase Foto1)/*tambahan buat edit foto,
    file Foto di hapus */
    {
        if (ModelState.IsValid)
        {
            if (Foto1 != null)
            {
                barang.Foto = new
                byte[Foto1.ContentLength];
                Foto1.InputStream.Read(barang.Foto, 0,
                Foto1.ContentLength);
                db.Entry(barang).State =
                EntityState.Modified;
            }
            else
            {
                db.Barang.Attach(barang);
                db.Entry(barang).Property(x =>
                x.Kategori_Barang_Fk).IsModified = true;
                db.Entry(barang).Property(x =>
                x.Merk_Fk).IsModified = true;
                db.Entry(barang).Property(x =>
                x>Nama_Barang).IsModified = true;
                db.Entry(barang).Property(x =>
                x.Jumlah).IsModified = true;
                db.Entry(barang).Property(x =>
                x.Tanggal).IsModified = true;
            }
            db.SaveChanges();
            return RedirectToAction("Index");
        }
        //coding foto diatas ini
        Barang barang = db.Barang.Find(id);
        db.Barang.Remove(barang);
        db.SaveChanges();
        return RedirectToAction("Index");
    }

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

```

```

        return HttpNotFound();
    }
    ViewBag.Kategori_Barang_Fk = new
SelectList(db.Kategori_Barang,

        ViewBag.Kategori_Barang_Fk =
new SelectList(db.Kategori_Barang,
"Kategori_Barang_id",
"Nama_Kategori_Barang",
barang.Kategori_Barang_Fk);
    ViewBag.Merk_Fk = new
SelectList(db.Merk, "Merk_id", "Nama_Merk",
barang.Merk_Fk);
    return View(barang);
}

// GET: Barang/Delete/5
public ActionResult Delete(int? id)
{
    if (id == null)
    {
        return new
HttpStatusCodeResult(HttpStatusCode.BadReq
uest);
    }
    Barang barang = db.Barang.Find(id);
    if (barang == null)
    {
        return HttpNotFound();
    }
    return View(barang);
}

// POST: Barang/Delete/5
[HttpPost, ActionName("Delete")]
[ValidateAntiForgeryToken]
public ActionResult DeleteConfirmed(int
id)
{

```

PenempatanController

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using inventori.Models;
using System.IO;
using inventori.Laporan;
using PagedList;
using PagedList.Mvc;

namespace inventori.Controllers
{
    [Authorize]
    public class PenempatansController
    : Controller
    {
        private InvenEntities db = new
InvenEntities();
        // GET: Penempatans
        public ActionResult
Index(string Barang_Fk, string
Lokasi_Fk, string searchString, int?
Page, string Status_Fk)
        {
            return
View(plist.OrderBy(x =>
x.Penempatan_id).ToPagedList(page,
5));

```

```

        ViewBag.Barang_Fk =
new SelectList(db.Barang,
"Barang_id", "Nama_Barang");
        ViewBag.Lokasi_Fk =
new SelectList(db.Lokasi_Barang,
"Lokasi_Barang_id",
"Nama_Lokasi_Barang");
        ViewBag.Status_Fk =
new SelectList(db.Status_Barang,
"Status_Barang_id",
"Nama_Status_Barang");
        var penempatan =
db.Penempatan.Include(p =>
p.Barang).Include(p =>
p.Lokasi_Barang).Include(p =>
p.Status_Barang);

        if
(!String.IsNullOrEmpty(searchStri
ng))
        {
            penempatan =
penempatan.Where(s =>
s.Barang>Nama_Barang.Contains(sea
rchString) ||

s.Lokasi_Barang>Nama_Lokasi_Baran
g.Contains(searchString) ||

s.Status_Barang>Nama_Status_Baran
g.Contains(searchString)

);
        }
        if
(!String.IsNullOrEmpty(Barang_Fk)
&&
(!String.IsNullOrEmpty(Lokasi_Fk)
&&
(!String.IsNullOrEmpty(Status_Fk)
)))
        {
            int b =
Convert.ToInt32(Barang_Fk);
            int l =
Convert.ToInt32(Lokasi_Fk);
            int s =
Convert.ToInt32(Status_Fk);
            List<Penempatan>
plist = db.Penempatan.Where(x =>
x.Barang_Fk == b && x.Lokasi_Fk
== l && x.Status_Fk ==
s).ToList();

            int l
= Convert.ToInt32(Lokasi_Fk);
            int s
= Convert.ToInt32(Status_Fk);
        }
    }
    else
    {
        if
(!String.IsNullOrEmpty(Barang_Fk) &&
(!String.IsNullOrEmpty(Lokasi_Fk) ))
        {
            int b =
Convert.ToInt32(Barang_Fk);
            int l =
Convert.ToInt32(Lokasi_Fk);
            List<Penempatan>
plist = db.Penempatan.Where(x =>
x.Barang_Fk == b && x.Lokasi_Fk ==
l).ToList();

            return
View(plist.OrderBy(x =>
x.Penempatan_id).ToPagedList(page,
5));
        }
        else
        {
            if
(!String.IsNullOrEmpty(Barang_Fk) &&
(!String.IsNullOrEmpty(Status_Fk)))
            {
                int b =
Convert.ToInt32(Barang_Fk);
                int s =
Convert.ToInt32(Status_Fk);
                List<Penempatan> plist =
db.Penempatan.Where(x => x.Barang_Fk
== b && x.Status_Fk == s).ToList();
                return
View(plist.OrderBy(x =>
x.Penempatan_id).ToPagedList(page,
5));
            }
            else {
                if
(!String.IsNullOrEmpty(Lokasi_Fk) &&
(!String.IsNullOrEmpty(Status_Fk)))
                {
                    db.Penempatan.Where(x => x.Status_Fk
== sa).ToList();

                    return
View(plist.OrderBy(x =>
x.Penempatan_id).ToPagedList(page,
5));
                }
            }
            return
View(penempatan.OrderBy(x =>
x.Penempatan_id).ToPagedList(page,
5));
        }
    }
}

```



```

        //Laporan
        Penempatan();
        if
        (!fromDate.HasValue) fromDate =
        DateTime.Now.Date;
        if (!toDate.HasValue)
        toDate =
        fromDate.GetValueOrDefault(DateTime
        .Now.Date).Date.AddDays(1);
        if (toDate <
        fromDate) toDate =
        fromDate.GetValueOrDefault(DateTime
        .Now.Date).Date.AddDays(1);
        ViewBag.fromDate =
        fromDate;
        ViewBag.toDate =
        toDate;

        List<M_Laporanpenempatan>
        santriList = new
        List<M_Laporanpenempatan>();
        var penempatans =
        db.Penempatan.Include(p =>
        p.Barang).Include(p =>
        p.Lokasi_Barang).Include(p =>
        p.Status_Barang).Where(c =>
        c.Tanggal >= fromDate &&
        c.Tanggal < toDate).ToList();
        foreach (var item in
        penempatans)
        {
            santriList.Add(new
            M_Laporanpenempatan
            {
                Penempatan_id
                = item.Penempatan_id,
                Barang_Fk =
                item.Barang.Nama_Barang,
                Kode_Barang =
                item.Barang.Kode_Barang,
                Lokasi_Fk =
                item.Lokasi_Barang.Nama_Lokasi_Ba
                rang,
                Status_Fk =
                item.Status_Barang.Nama_Status_Ba
                rang,
                Jumlah =
                item.Jumlah == null ? "0" :
                item.Jumlah.ToString(),
                Keterangan =
                item.Keterangan,
                "Lokasi_Barang_id",
                "Nama_Lokasi_Barang");
            ViewBag.Status_Fk =
            new SelectList(db.Status_Barang,
            red.ExportFormatType.PortableDocFormat
            );
            return
            File(CetakLaporanSantri,
            "application/pdf");
        }

        // GET: Penempatans/Details/5
        public ActionResult
        Details(int? id)
        {
            if (id == null)
            {
                return new
                HttpStatusCodeResult(HttpStatusCode.Ba
                dRequest);
            }
            Penempatan penempatan =
            db.Penempatan.Find(id);
            if (penempatan == null)
            {
                return HttpNotFound();
            }
            return View(penempatan);
        }

        // GET: Penempatans/Create
        public ActionResult
        Create(string message)
        {
            ViewBag.Message = message;
            ViewBag.Barang_Fk = new
            SelectList(db.Barang, "Barang_id",
            "Nama_Barang");
            ViewBag.Lokasi_Fk = new
            SelectList(db.Lokasi_Barang,
            {
                var cek_penempatan
                = db.Penempatan.Where(m => m.Barang_Fk
                == penempatan.Barang_Fk).ToList();
                if
                (cek_penempatan.Count() > 0)
                {
                    var
                    jml_barang_1 = 0;
                    jml_barang_1 =
                    cek_penempatan.Sum(m =>
                    Convert.ToInt32(m.Jumlah));
                    var hitung =
                    jml_barang - jml_barang_1;
                    if (hitung <=
                    0)
                    {
                        ViewBag.Message = "jumlah barang

```

```

        "Status_Barang_id",
        "Nama_Status_Barang");
        return View();
    }

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

    [ValidateAntiForgeryToken]
    public ActionResult
    Create([Bind(Include =
    "Penempatan_id,Barang_Fk,Lokasi_F
    k,Status_Fk,Jumlah,Keterangan,Tan
    ggal")] Penempatan penempatan)
    { /*coding untuk
    penjumlahan barang | Dibawah ini
    |hanya di babagian create atau
    tambah barang*/
        if
        (ModelState.IsValid)
        {
            var cek_barang =
            db.Barang.Where(m => m.Barang_id
            ==
            penempatan.Barang_Fk).ToList();
            var jml_barang =
            0;
            foreach (var itm
            in cek_barang)
            {
                jml_barang +=
                cek_barang.Sum(m =>
                Convert.ToInt32(m.Jumlah));
            }
            if
            (penempatan.Jumlah > jml_barang)
            {

                ViewBag.Message = "jumlah barang
                melebihi kapasitas barang";
                return
                RedirectToAction("Create", new {
                message = ViewBag.Message });
            }
            else

            return RedirectToAction("Index");
        }
        else
    
```

```

        melebihi kapasitas barang di
        penempatan";
        return
        RedirectToAction("Create", new {
        message = ViewBag.Message });
    }
    if (hitung <
    penempatan.Jumlah)
    {
        ViewBag.Message = "jumlah barang
        terlalu banyak dari jumlah
        penempatan";
        return
        RedirectToAction("Create", new {
        message = ViewBag.Message });
    }
    //coding untuk
    tambah jika data udah ada di
    penempatan
    else
    {
        var
        cek_penempatan2 =
        db.Penempatan.Where(m => m.Barang_Fk
        == penempatan.Barang_Fk && m.Lokasi_Fk
        == penempatan.Lokasi_Fk && m.Status_Fk
        == penempatan.Status_Fk).ToList();
        if
        (cek_penempatan2.Count() > 0)
        {
            var
            id_penempatan =
            cek_penempatan2.FirstOrDefault().Penem
            patan_id;

            Penempatan penempatanB =
            db.Penempatan.Where(x =>
            x.Penempatan_id ==
            id_penempatan).FirstOrDefault();

            penempatanB.Jumlah = penempatan.Jumlah
            +
            cek_penempatan2.FirstOrDefault().Jumla
            h;

            penempatanB.Barang_Fk =
            cek_penempatan2.FirstOrDefault().Baran
            g_Fk;

            penempatanB.Lokasi_Fk =
            cek_penempatan2.FirstOrDefault().Lokas
            i_Fk;

            penempatanB.Status_Fk =
            cek_penempatan2.FirstOrDefault().Statu
            s_Fk;
    
```



```

        {
            db.Penempatan.Add(penempatan);

            db.SaveChanges();

            return RedirectToAction("Index");
        }
    }

    db.Penempatan.Add(penempatan);

    db.SaveChanges();

    return
    RedirectToAction("Index");
}
/*coding untuk
penjumlahan barang | Dibawah atas
|hanya di babagian create atau
tambah barang*/

    ViewBag.Barang_Fk =
    new SelectList(db.Barang,
    "Barang_id", "Nama_Barang",
    penempatan.Barang_Fk);
    ViewBag.Lokasi_Fk =
    new SelectList(db.Lokasi_Barang,
    "Lokasi_Barang_id",
    "Nama_Lokasi_Barang",
    penempatan.Lokasi_Fk);
    ViewBag.Status_Fk =
    new SelectList(db.Status_Barang,
    "Status_Barang_id",
    "Nama_Status_Barang",
    penempatan.Status_Fk);
    return
    View(penempatan);
}

// GET:
Penempatans/Edit/5
public ActionResult
Edit(int? id)
{
    if (id == null)
    {
        return new
        HttpStatusCodeResult(HttpStatusCode.
        BadRequest);
    }
    Penempatan penempatan
    = db.Penempatan.Find(id);
    if (penempatan ==
    null)
    {
        penempatanB.Keterangan =
        penempatan.Keterangan;

        penempatanB.Tanggal =
        penempatan.Tanggal;

        penempatanB.Lokasi_Barang =
        cek_penempatan2.FirstOrDefault().Lokas
        i_Barang;

        penempatanB.Mutasi_Barang =
        cek_penempatan2.FirstOrDefault().Mutas
        i_Barang;

        penempatanB.Barang =
        cek_penempatan2.FirstOrDefault().Baran
        g;

        penempatanB.Status_Barang =
        cek_penempatan2.FirstOrDefault().Statu
        s_Barang;

        db.Entry(penempatanB).State =
        EntityState.Modified;

        db.SaveChanges();
        == penempatan.Barang_Fk).ToList();
        var jml_barang = 0;
        foreach (var itm in
        cek_barang)
        {
            jml_barang +=
            cek_barang.Sum(m =>
            Convert.ToInt32(m.Jumlah));
        }
        if (penempatan.Jumlah >
        jml_barang)
        {
            ViewBag.Message =
            "jumlah barang melebihi kapasitas
            barang";
            return
            RedirectToAction("Edit", new { id =
            penempatan.Penempatan_id , message =
            ViewBag.Message });
        }
        else
        {
            db.Entry(penempatan).State =
            EntityState.Modified;
            db.SaveChanges();
            return
            RedirectToAction("Index");
        }
    }
}

```

```

        return
    HttpNotFound();
    }
    var barang =
    db.Barang.Select(s => new
    {
        Barang_id =
    s.Barang_id,
        Nama =
    s>Nama_Barang + " + " +
    s.Kode_Barang
    }).ToList();
    ViewBag.Barang_Fk =
    new SelectList(barang,
    "Barang_id", "Nama");
    ViewBag.Lokasi_Fk =
    new SelectList(db.Lokasi_Barang,
    "Lokasi_Barang_id",
    "Nama_Lokasi_Barang",
    penempatan.Lokasi_Fk);
    ViewBag.Status_Fk =
    new SelectList(db.Status_Barang,
    "Status_Barang_id",
    "Nama_Status_Barang",
    penempatan.Status_Fk);
    return
    View(penempatan);
    }

```

```

    // POST:
    Penempatans/Edit/5
    // To protect from
    overposting attacks, please
    enable the specific properties
    you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?L
    inkId=317598.
    [HttpPost]

```

```

[ValidateAntiForgeryToken]
    public ActionResult
    Edit([Bind(Include =
    "Penempatan_id,Barang_Fk,Lokasi_F
    k,Status_Fk,Jumlah,Keterangan,Tan
    ggal")] Penempatan penempatan)
    { /*coding untuk
    penjumlahan barang | Dibawah ini
    | Disini untuk mengedit Jumlah
    Barang*/
        var cek_barang =
    db.Barang.Where(m => m.Barang_id
    if (id == null)
    {
        return new
    HttpStatusCodeResult(HttpStatusCode
    de.BadRequest);
    }

```

```

    }/*coding untuk
    penjumlahan barang | Dibawah atas
    |Disini untuk mengedit Jumlah Barang*/
    ViewBag.Barang_Fk = new
    SelectList(db.Barang, "Barang_id",
    "Nama_Barang", penempatan.Barang_Fk);
    ViewBag.Lokasi_Fk = new
    SelectList(db.Lokasi_Barang,
    "Lokasi_Barang_id",
    "Nama_Lokasi_Barang",
    penempatan.Lokasi_Fk);
    ViewBag.Status_Fk = new
    SelectList(db.Status_Barang,
    "Status_Barang_id",
    "Nama_Status_Barang",
    penempatan.Status_Fk);
    return View(penempatan);
    }
    // GET: Penempatans/Delete/5
    public ActionResult
    Delete(int? id)
    {

```

MutasisController

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using inventori.Models;
using System.IO;
using inventori.Laporan;
using PagedList;
using PagedList.Mvc;

namespace inventori.Controllers
{
    [Authorize]
    public class
    Mutasi_BarangController : Controller
    {
        private InvenEntities db = new
    InvenEntities();

    // GET: Mutasi_Barang

```

```

    }
    Penempatan penempatan
= db.Penempatan.Find(id);
    if (penempatan ==
null)
    {
        return
 HttpNotFound();
    }
    return
 View(penempatan);
}

// POST:
Penempatans/Delete/5
[HttpPost,
ActionName("Delete")]

[ValidateAntiForgeryToken]
public ActionResult
DeleteConfirmed(int id)
{
    Penempatan penempatan
= db.Penempatan.Find(id);

    db.Penempatan.Remove(penempatan);
    db.SaveChanges();
    return
 RedirectToAction("Index");
}

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

    base.Dispose(disposing);
}
}

return new
 HttpStatusCodeResult(HttpStatusCode.
BadRequest);
}

```

```

public ActionResult Index(int
? Page, string Lokasi_Baru_Fk, string
searchString)
{
    int page = Page ?? 1;
    ViewBag.Lokasi_Baru_Fk =
new SelectList(db.Lokasi_Barang,
"Lokasi_Barang_id",
>Nama_Lokasi_Barang");
    ViewBag.Lokasi_Fk = new
SelectList(db.Penempatan,
"Penempatan_id", "Lokasi_Fk");
    var mutasi_Barang =
db.Mutasi_Barang.Include(m =>
m.Lokasi_Barang).Include(m =>
m.Penempatan);

    if
(!String.IsNullOrEmpty(searchString))
    {
        mutasi_Barang =
mutasi_Barang.Where(s =>
s.Penempatan.Barang>Nama_Barang.Contai
ns(searchString) ||
s.Lokasi_Barang>Nama_Lokasi_Barang.Con
tains(searchString)
);
    }
}

```

Kategori_BarangController

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using inventori.Models;

namespace inventori.Controllers
{
    [Authorize]
    public class
Kategori_BarangController : Controller
    {
        private InvenEntities db = new
InvenEntities();

        // GET: Kategori_Barang
public ActionResult Details(int? id)
    {
        if (id == null)
        {

```

```

        Kategori_Barang
    kategori_Barang =
    db.Kategori_Barang.Find(id);
        if (kategori_Barang
    == null)
        {
            return
    HttpNotFound();
        }
        return
    View(kategori_Barang);
    }

    // GET:
    Kategori_Barang/Create
    public ActionResult
    Create()
    {
        return View();
    }

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

    [ValidateAntiForgeryToken]
    public ActionResult
    Create([Bind(Include =
    "Kategori_Barang_id,Nama_Kategori
    _Barang")] Kategori_Barang
    kategori_Barang)
    {
        if
    (ModelState.IsValid)
    {
        db.Kategori_Barang.Add(kategori_B
    arang);
        db.SaveChanges();
        return
    RedirectToAction("Index");
    }

    return
    View(kategori_Barang);
    }

    return new
    HttpStatusCodeResult(HttpStatusCode.
    de.BadRequest);
}

// GET:
    Kategori_Barang/Edit/5
    public ActionResult Edit(int?
    id)
    {
        if (id == null)
        {
            return new
    HttpStatusCodeResult(HttpStatusCode.Ba
    dRequest);
        }
        Kategori_Barang
    kategori_Barang =
    db.Kategori_Barang.Find(id);
        if (kategori_Barang ==
    null)
        {
            return HttpNotFound();
        }
        return
    View(kategori_Barang);
    }

    // POST:
    Kategori_Barang/Edit/5
    // To protect from overposting
    attacks, please enable the specific
    properties you want to bind to, for
    // more details see
    http://go.microsoft.com/fwlink/?LinkId
    =317598.
    [HttpPost]
    [ValidateAntiForgeryToken]
    public ActionResult
    Edit([Bind(Include =
    "Kategori_Barang_id,Nama_Kategori_Ba
    rang")] Kategori_Barang kategori_Barang)
    {
        if (ModelState.IsValid)
        {
            db.Entry(kategori_Barang).State =
    EntityState.Modified;
            db.SaveChanges();
            return
    RedirectToAction("Index");
        }
        return
    View(kategori_Barang);
    }

    // GET:
    Kategori_Barang/Delete/5
    public ActionResult
    Delete(int? id)
    {
        if (id == null)

```

```

    }
    Kategori_Barang
    kategori_Barang =
    db.Kategori_Barang.Find(id);
    if (kategori_Barang
    == null)
    {
        return
    HttpNotFound();
    }
    return
    View(kategori_Barang);
}

// POST:
Kategori_Barang/Delete/5
[HttpPost,
ActionName("Delete")]

[ValidateAntiForgeryToken]
public ActionResult
DeleteConfirmed(int id)
{
    Kategori_Barang
    kategori_Barang =
    db.Kategori_Barang.Find(id);

    db.Kategori_Barang.Remove(kategor
    i_Barang);

    db.SaveChanges();
    return
    RedirectToAction("Index");
}

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

    base.Dispose(disposing);
}
}
}

```

```

// To protect from overposting
attacks, please enable the
specific properties you want to
bind to, for

```

Lokasi_BarangController

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using inventori.Models;

namespace inventori.Controllers
{
    [Authorize]
    public class
    Lokasi_BarangController : Controller
    {
        private InvenEntities db = new
    InvenEntities();

        // GET: Lokasi_Barang
        public ActionResult Index()
        {
            return
            View(db.Lokasi_Barang.ToList());
        }

        // GET:
        Lokasi_Barang/Details/5
        public ActionResult
        Details(int? id)
        {
            if (id == null)
            {
                return new
            HttpStatusCodeResult(HttpStatusCode.Ba
            dRequest);
            }

            Lokasi_Barang
            lokasi_Barang =
            db.Lokasi_Barang.Find(id);
            if (lokasi_Barang == null)
            {
                return HttpNotFound();
            }

            return
            View(lokasi_Barang);
        }

        // GET: Lokasi_Barang/Create
        public ActionResult Create()
        {
            return View();
        }
    }
}

```

```

        // more details see
        http://go.microsoft.com/fwlink/?L
        inkId=317598.
        [HttpPost]

[ValidateAntiForgeryToken]
    public ActionResult
Create([Bind(Include =
"Lokasi_Barang_id,Nama_Lokasi_Bar
ang,Keterangan")] Lokasi_Barang
lokasi_Barang)
    {
        if
(ModelState.IsValid)
        {

db.Lokasi_Barang.Add(lokasi_Barang);
            db.SaveChanges();
            return
RedirectToAction("Index");
        }

        return
View(lokasi_Barang);
    }

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

    // POST:
    Lokasi_Barang/Edit/5
    Lokasi_Barang
lokasi_Barang =
db.Lokasi_Barang.Find(id);

```

```

        // POST: Lokasi_Barang/Create

        // To protect from
        overposting attacks, please enable the
        specific properties you want to bind
        to, for
        // more details see
        http://go.microsoft.com/fwlink/?LinkId
        =317598.
        [HttpPost]
[ValidateAntiForgeryToken]
    public ActionResult
Edit([Bind(Include =
"Lokasi_Barang_id,Nama_Lokasi_Barang,K
eterangan")] Lokasi_Barang
lokasi_Barang)
    {
        if (ModelState.IsValid)
        {

db.Entry(lokasi_Barang).State =
EntityState.Modified;
            db.SaveChanges();
            return
RedirectToAction("Index");
        }
        return
View(lokasi_Barang);
    }

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

    // POST:
    Lokasi_Barang/Delete/5
    [HttpPost,
ActionName("Delete")]
[ValidateAntiForgeryToken]

```

```

db.Lokasi_Barang.Remove(lokasi_Ba
rang);
        db.SaveChanges();
        return
RedirectToAction("Index");
    }

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

base.Dispose(disposing);
    }
}

```

MerkSController

```

using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
using System.Linq;
using System.Net;
using System.Web;
using System.Web.Mvc;
using inventori.Models;

namespace inventori.Controllers
{
    [Authorize]
    public class MerksController
: Controller
    {
        private InvenEntities db
= new InvenEntities();

        // GET: Merks
        public ActionResult
Index()
        {
            var merk =
db.Merk.Include(m =>
m.Kategori_Barang);
            return
View(merk.ToList());
        }

        // GET: Merks/Details/5
    }

    // GET: Merks/Edit/5

        public ActionResult
DeleteConfirmed(int id)
        {
            public ActionResult
Details(int? id)
            {
                if (id == null)
                {
                    return new
HttpStatusCodeResult(HttpStatusCode.Ba
dRequest);
                }
                Merk merk =
db.Merk.Find(id);
                if (merk == null)
                {
                    return HttpNotFound();
                }
                return View(merk);
            }

            // GET: Merks/Create
            public ActionResult Create()
            {
                ViewBag.Kategori_Fk = new
SelectList(db.Kategori_Barang,
"Kategori_Barang_id",
"Nama_Kategori_Barang");
                return View();
            }

            // POST: Merks/Create
            // To protect from overposting
attacks, please enable the specific
properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId=317598.
            [HttpPost]
            [ValidateAntiForgeryToken]
            public ActionResult
Create([Bind(Include =
"Merk_id,Nama_Merk,Kategori_Fk")] Merk
merk)
            {
                if (ModelState.IsValid)
                {
                    db.Merk.Add(merk);
                    db.SaveChanges();
                    return
RedirectToAction("Index");
                }

                ViewBag.Kategori_Fk = new
SelectList(db.Kategori_Barang,
"Kategori_Barang_id",

```

```

        public ActionResult
Edit(int? id)
    {
        if (id == null)
        {
            return new
 HttpStatusCodeResult(HttpStatusCode.
de.BadRequest);
        }
        Merk merk =
db.Merk.Find(id);
        if (merk == null)
        {
            return
HttpNotFound();
        }
        ViewBag.Kategori_Fk =
new
 SelectList(db.Kategori_Barang,
"Kategori_Barang_id",
"Nama_Kategori_Barang",
merk.Kategori_Fk);
        return View(merk);
    }

    // POST: Merks/Edit/5
    // To protect from
overposting attacks, please
enable the specific properties
you want to bind to, for
    // more details see
http://go.microsoft.com/fwlink/?L
inkId=317598.
    [HttpPost]

[ValidateAntiForgeryToken]
    public ActionResult
Edit([Bind(Include =
"MerK_id,Nama_Merk,Kategori_Fk")]
Merk merk)
    {
        if
(ModelState.IsValid)
        {
            db.Entry(merk).State =
 EntityState.Modified;
            db.SaveChanges();
            return
RedirectToAction("Index");
        }
        ViewBag.Kategori_Fk =
new
 SelectList(db.Kategori_Barang,
"Kategori_Barang_id",
using System.Linq;
using System.Net;

        "Nama_Kategori_Barang",
merk.Kategori_Fk);
        return View(merk);
    }

    // GET: Merks/Delete/5
    public ActionResult
Delete(int? id)
    {
        if (id == null)
        {
            return new
 HttpStatusCodeResult(HttpStatusCode.Ba
dRequest);
        }
        Merk merk =
db.Merk.Find(id);
        if (merk == null)
        {
            return HttpNotFound();
        }
        return View(merk);
    }

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

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

using System;

```

Status_BarangController

```
using System;
```



```

using System.Web;
using System.Web.Mvc;
using inventori.Models;

namespace inventori.Controllers
{
    [Authorize]
    public class
Status_BarangController :
Controller
    {
        private InvenEntities db
= new InvenEntities();

        // GET: Status_Barang
public ActionResult
Index()
    {
        return
View(db.Status_Barang.ToList());
    }

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

        // GET:
Status_Barang/Create
public ActionResult
Create()
    {
        return View();
    }

        public ActionResult
Edit([Bind(Include =
using System.Collections.Generic;
using System.Data;
using System.Data.Entity;
// POST: Status_Barang/Create
// To protect from overposting
attacks, please enable the specific
properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId
=317598.
[HttpPost]
[ValidateAntiForgeryToken]
public ActionResult
Create([Bind(Include =
"Status_Barang_id, Nama_Status_Barang")
] Status_Barang status_Barang)
    {
        if (ModelState.IsValid)
        {
            db.Status_Barang.Add(status_Barang);
            db.SaveChanges();
            return
RedirectToAction("Index");
        }

        return
View(status_Barang);
    }

        // GET: Status_Barang/Edit/5
public ActionResult Edit(int?
id)
    {
        if (id == null)
        {
            return new
 HttpStatusCodeResult(HttpStatusCode.Ba
dRequest);
            Status_Barang
status_Barang =
db.Status_Barang.Find(id);
            if (status_Barang == null)
            {
                return HttpNotFound();
            }
            return
View(status_Barang);
        }

        // POST: Status_Barang/Edit/5
    {

[ValidateAntiForgeryToken]
public ActionResult
DeleteConfirmed(int id)

```

```

        "Status_Barang_id,Nama_Status_Barang")]] Status_Barang
status_Barang)
    {
        if
(ModelState.IsValid)
        {
            db.Entry(status_Barang).State =
EntityState.Modified;
            db.SaveChanges();
            return
RedirectToAction("Index");
        }
        return
View(status_Barang);
    }

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

// POST:
Status_Barang/Delete/5
[HttpPost,
ActionName("Delete")]
<a class="btn icon-btn btn-
primary"
href="http://localhost:13438/Bara
ngs/Create"><span
class="glyphicon btn-glyphicon
glyphicon-plus img-circle text-
danger"></span>Tambah Barang</a>
}

</p>
@*@Html.ActionLink("Tambah
Barang", "Create", null,

```

```

        Status_Barang
status_Barang =
db.Status_Barang.Find(id);
// To protect from overposting
attacks, please enable the specific
properties you want to bind to, for
// more details see
http://go.microsoft.com/fwlink/?LinkId
=317598.
[HttpPost]
[ValidateAntiForgeryToken]
db.Status_Barang.Remove(status_Barang)
;
        db.SaveChanges();
        return
RedirectToAction("Index");
    }

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

```

Barang_Index_View

```

@using PagedList.Mvc;
@using PagedList;
@*di atas untuk nambahin paging di list*@

@model IPagedList<inventori.Models.Barang>
@{
    ViewBag.Title = "Data Barang";
}

<link href="~/Content/home.css" rel="stylesheet" />
<link href="~/Content/home1.css" rel="stylesheet" />
<center><h2>Daftar Semua Data Barang
</h2></center>
<p>
@if (Request.IsAuthenticated)
    {
        </table>
    }
<p></p>
<table id="Barang" class="table table-striped table-
bordered" cellspacing="0" width="100%">
    <tr>
        <th>No</th>
        <th>Kode Barang</th>

```

```

htmlAttributes: new { @class =
"btn btn-primary"})*@
    @using
(Html.BeginForm("Index",
"Barangs", FormMethod.Get))
    {
        <table>
            <tr>
                <th>Pencarian: </th>

                <th>@Html.TextBox("SearchString",
null, htmlAttributes: new {
@class = "form-control", style =
"color:default" })/>
                    <th><input
type="submit" value="Cari"class="
btn btn-primary form-control"
/></th>

                    <th
style="width:200px">

                    @Html.DropDownList("Kategori_Bara
ng_Fk", null, "Semua",
htmlAttributes: new { @class =
"form-control", style =
"color:default" })
                        </th>
                            <th
style="width:200px">

                            @Html.DropDownList("Merk_Fk",
null, "Semua", htmlAttributes:
new { @class = "form-control",
style = "color:default" })
                                </th>
                                    <th>
                                        <input style=""
type="submit" value="Filter"
class=" btn btn-primary form-
control" />
                                            </th>
                                                </tr>
                                                    @*//TAMBAHAN
CODING UNTUK MEMBUAT PENCARIAN
SESUAI NAMA BARANG DAN LOKASI
BARANG*@
                                                        @*<td>

                                                        @Html.DisplayFor(modelItem =>
item.Status_Barang>Nama_Status_Ba
rang)
                                                            </td>*@
                                                                <td>

                                                                @Html.DisplayFor(modelItem =>
item.Jumlah)
                                                                    </td>

                                                                    <th>Nama Barang</th>
                                                                    <th>Merk Barang</th>
                                                                    <th>Kategori Barang</th>
                                                                    <th>Jumlah Barang</th>
                                                                    <th>Gambar Barang</th>
                                                                    <th>Tanggal Masuk</th>
                                                                    <th>Action</th>
                                                                    </tr>
                                                                    @*@ { int rowNo = 0; }*@
                                                                    @ {
                                                                    var pagenumber = Model.PageCount <
Model.PageNumber ? 0 : Model.PageNumber;
                                                                    var rowperpage = Model.PageSize;
                                                                    int rowNo = 0;
                                                                    if (pagenumber > 1)
                                                                    {
                                                                    int angka = pagenumber - 1;
                                                                    rowNo = rowperpage * angka;
                                                                    }
                                                                    }
                                                                    @foreach (var item in Model)
                                                                    {
                                                                    <tr>
                                                                    <td>@(rowNo += 1)</td>
                                                                    <td>
                                                                    @Html.DisplayFor(modelItem =>
item.Kode_Barang)
                                                                    </td>
                                                                    <td>
                                                                    @Html.DisplayFor(modelItem =>
item>Nama_Barang)
                                                                    </td>
                                                                    <td>
                                                                    @Html.DisplayFor(modelItem =>
item.Merk>Nama_Merk)
                                                                    </td>
                                                                    <td>
                                                                    @Html.DisplayFor(modelItem =>
item.Merk.Kategori_Barang>Nama_Kategori_Barang)
                                                                    </td>
                                                                    @if (Request.IsAuthenticated)
                                                                    {
                                                                    <center>
                                                                    <br />
                                                                    <br />
                                                                    <a>@Html.ActionLink(" ", "Delete",
new
                                                                    { id = item.Barang_id }, htmlAttributes: new { @class
= "btn btn-danger glyphicon glyphicon-trash" })/>
                                                                    </center>
                                                                    }
                                                                    </div>
                                                                    </td>
                                                                    </tr>
                                                                    }
                                                                    <tr>
                                                                    <th>No</th>
                                                                    <th>Kode Barang</th>
                                                                    <th>Nama Barang</th>

```

```

        <td >
            @{
                var base64 =
Convert.ToBase64String(item.Foto)
;
                var imgsrc =
string.Format("data:image/jpg;base64,{0}", base64);
            }
            <img src=@imgsrc'
width="150" height="150" />
        </td>
    </td>

```

```

@Html.DisplayFor(modelItem =>
item.Tanggal)
</td>

```

```

    <td>
        <div
class="glyphicon">
            @if
(Request.IsAuthenticated)
            {
                <center>

```

```

<a>@Html.ActionLink(" ", "Edit",
new { id = item.Barang_id },
htmlAttributes: new { @class =
"btn btn-primary glyphicon
glyphicon-edit" })</a>
                </center>
            }
            <center>
                <br />
                <br />

```

```

<a>@Html.ActionLink(" ",
"Details", new { id =
item.Barang_id }, htmlAttributes:
new { @class = "btn btn-primary
glyphicon glyphicon-list-alt
icon" })</a>
            </center>
        });
    });
</script>

```

```

@using (Html.BeginForm("Create",
"Barangs", FormMethod.Post, new {

```

```

<th>Merk Barang</th>
<th>Kategori Barang</th>
<th>Jumlah Barang</th>
<th>Gambar Barang</th>
<th>Tanggal Masuk</th>
<th>Action</th>
</tr>
</table>
@*untuk nambahin paging di list*@
<div id="container" style="margin-left: 20px">
    <p></p>
    <p></p>
    <div class="pagination" style="margin-left: 400px">
        Page @ (Model.PageCount < Model.PageNumber
? 0 : Model.PageNumber)
        of @ Model.PageCount
    @Html.PagedListPager(Model, page =>
Url.Action("Index", new { page }))
    </div>
</div>

```

Barang_Create_View

```

@model inventori.Models.Barang
@{
    ViewBag.Title = "Tambah Barang";
}
<link href="~/Content/home.css" rel="stylesheet" />
<link href="~/Content/home1.css" rel="stylesheet" />
<h2>Form Tambah Barang</h2>
@*TAMBAHAN CODING UNTUK
MEMBUAT OTOMATIS PEMILIHAN KATEGORI
DAN MERK DI VIEW*@
<script src="~/Scripts/jquery-1.10.2.min.js"></script>
<script>
    $(document).ready(function () {
        $("#Kategori_Barang_Fk").change(function () {
            $.get("/Barangs/GetMerkList", { Merk_Fk:
$("#Kategori_Barang_Fk").val() }, function (data) {
                $("#Merk_Fk").empty();
                $.each(data, function (create, row) {
                    $("#Merk_Fk").append("<option value="
+ row.Merk_id + ">" + row>Nama_Merk +
"</option>");
                });
            }
            -----", new { @class = "form-control"
        })
    });
    @Html.ValidationMessageFor(model =>
model.Kategori_Barang_Fk, "", new { @class = "text-
danger" })
</div>
</div>
@*CODING UNTUK TAMBAHAN
DI KATEGORI AGAR OTOMATIS*@
@*CODING UNTUK TAMBAHAN DI
MERK AGAR OTOMATIS*@
<div class="form-group">
    @Html.LabelFor(model => model.Merk_Fk,
"Merk Barang ", htmlAttributes: new { @class =
"control-label col-md-2" })

```

```

enctype = "multipart/form-data"
}))
{
    @Html.AntiForgeryToken()
    <a class="btn icon-btn btn-
primary"
href="http://localhost:13438/Kate
gori_Barang/Create"><span
class="glyphicon btn-glyphicon
glyphicon-plus img-circle text-
danger"></span>Tambah
Kategori</a>
    <a class="btn icon-btn btn-
primary"
href="http://localhost:13438/Merk
s/Create"><span class="glyphicon
btn-glyphicon glyphicon-plus img-
circle text-danger"></span>Tambah
Merk</a>
    <div class="form-horizontal">
        <br /> <hr />
        @*CODING
UNTUK TAMBAHAN DI KATEGORI AGAR
OTOMATIS*@
    @Html.ValidationSummary(true, "",
new { @class = "text-danger" })
    <div class="form-group">
        @Html.LabelFor(model
=>
model.Kategori_Barang_Fk, "Kategor
i Barang" , htmlAttributes: new {
@class = "control-label col-md-2"
})
        <div class="col-md-
10">
            @if
(ViewBag.Kategori_Barang_Fk !=
null)
            {
    @Html.DropDownListFor(model =>
model.Kategori_Barang_Fk,
ViewBag.Kategori_Barang_Fk as
SelectList, "-----
Pilih Kategori
    @Html.LabelFor(model =>
model.Tanggal, "Tanggal Masuk",
htmlAttributes: new { @class = "control-label
col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model =>
model.Tanggal, new { htmlAttributes = new {
@class = "form-control" } })
            @Html.ValidationMessageFor(model
=> model.Tanggal, "", new { @class = "text-
danger" })
        </div>
        <div class="col-md-10">
            @Html.DropDownListFor(model =>
model.Merk_Fk, new SelectList(" "), "-----
Pilih Merk-----", new { @class = "form-
control" })
            @Html.ValidationMessageFor(model =>
model.Merk_Fk, "", new { @class = "text-danger" })
        </div>
        <div>
        </div>
        @*CODING UNTUK
TAMBAHAN DI MERK AGAR OTOMATIS*@
        <div class="form-group">
            @Html.LabelFor(model =>
model>Nama_Barang, "Nama Barang",htmlAttributes:
new { @class = "control-label col-md-2" })
            <div class="col-md-10">
                @Html.EditorFor(model =>
model>Nama_Barang, new { htmlAttributes = new {
@class = "form-control" } })
                @Html.ValidationMessageFor(model =>
model>Nama_Barang, "", new { @class = "text-danger"
})
            </div>
        </div>
        @*CODING UNTUK
MENAMBAHKAN FOTO ATAU GAMBAR*@
        <div class="form-group">
            @Html.LabelFor(model => model.Foto, "Foto
Barang",htmlAttributes: new { @class = "control-label
col-md-2" })
            <div class="col-md-10">
                </div>
            </div>
        </div>
    @section Scripts {
    @Scripts.Render("~/bundles/jqueryval")
    }
    <input type="file" id="Foto1"
name="Foto1" />
    @Html.ValidationMessageFor(model =>
model.Foto, "", new { @class = "text-danger" })
    </div>
    @*CODING UNTUK
MENAMBAHKAN FOTO ATAU GAMBAR*@
    <div class="form-group">
        @Html.LabelFor(model =>
model.Jumlah, "Jumlah Barang", htmlAttributes: new {
@class = "control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model => model.Jumlah,
new { htmlAttributes = new { @class = "form-control"
} })
            @Html.ValidationMessageFor(model =>
model.Jumlah, "", new { @class = "text-danger" })
        </div>
    </div>
    <div class="form-group">
        Barang_Fk", null, htmlAttributes: new
{ @class = "form-control" })

```

```

    </div>
    <div class="form-group">
        <div class="col-md-offset-2 col-md-10">
            <a
href="http://localhost:13438/Barangs"
class="btn btn-primary">Kembali</a>
            <input type="submit"
value="Tambah" class="btn btn-primary" />

```

Barang_Edit_View

```
@model inventori.Models.Barang
```

```

@{
    ViewBag.Title = "Edit";
}
<link href="~/Content/home.css"
rel="stylesheet" />
<link href="~/Content/home1.css"
rel="stylesheet" />
<h2>Ubah Data Barang</h2>
@using (Html.BeginForm("Edit",
"Barangs", FormMethod.Post, new {
enctype = "multipart/form-data"
}))
{
    @Html.AntiForgeryToken()

    <div class="form-horizontal">
        <hr />

        @Html.ValidationSummary(true, "",
new { @class = "text-danger" })
        @Html.HiddenFor(model =>
model.Barang_id)

        <div class="form-group">
            @Html.LabelFor(model
=> model.Kategori_Barang_Fk,
"Kategori Barang",
htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">

                @Html.DropDownList("Kategori_

                    <img
src=@imgsrc' width="150"
height="150" />
                    <input
type="file" id="Foto1"
name="Foto1" />

                @Html.ValidationMessageFor(model
=> model.Foto, "", new { @class =
"text-danger" })

```

```

@Html.ValidationMessageFor(model =>
model.Kategori_Barang_Fk, "", new {
@class = "text-danger" })
        </div>
    </div>

```

```

    <div class="form-group">
        @Html.LabelFor(model =>
model.Merk_Fk, "Merk Barang",
htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">

```

```

@Html.DropDownList("Merk_Fk", null,
htmlAttributes: new { @class = "form-
control" })

```

```

@Html.ValidationMessageFor(model =>
model.Merk_Fk, "", new { @class =
"text-danger" })
        </div>
    </div>

```

```

    <div class="form-group">
        @Html.LabelFor(model =>
model>Nama_Barang, "Nama
Barang",htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model
=> model>Nama_Barang, new {
htmlAttributes = new { @class = "form-
control" } })

```

```

@Html.ValidationMessageFor(model =>
model>Nama_Barang, "", new { @class =
"text-danger" })
        </div>
    </div>

```

```

        @*CODING UNTUK
MENAMBAHKAN FOTO ATAU GAMBAR*@

```

```

        <div class="form-group">
            @Html.LabelFor(model =>
model.Foto, "Foto
Barang",htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">
                @{

```

```

                    var base64 =
Convert.ToBase64String(Model.Foto);
                    var imgsrc =
string.Format("data:image/jpg;base64,{
0}", base64);
                }

```

```

                Barangs" class="btn btn-
primary">Kembali</a>

```

```

        </div>
    </div>
    @*CODING
UNTUK MENAMBAHKAN FOTO ATAU
GAMBAR*@
    <div class="form-group">
        @Html.LabelFor(model
=> model.Jumlah, "Jumlah Barang",
htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-
10">
@Html.EditorFor(model =>
model.Jumlah, new {
htmlAttributes = new { @class =
"form-control" } })
@Html.ValidationMessageFor(model
=> model.Jumlah, "", new { @class
= "text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model
=> model.Tanggal, "Tanggal
Masuk",htmlAttributes: new {
@class = "control-label col-md-2"
})
        <div class="col-md-
10">
@Html.EditorFor(model =>
model.Tanggal, new {
htmlAttributes = new { @class =
"form-control" } })
@Html.ValidationMessageFor(model
=> model.Tanggal, "", new {
@class = "text-danger" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-
offset-2 col-md-10">
            <a
href="http://localhost:13438/
<th style="width:200px">
@Html.DropDownList("Barang_Fk",
null, "Semua", htmlAttributes:
new { @class = "form-control",
style = "color:default" })
        </th>
        <th
style="width:200px">
@Html.DropDownList("Lokasi_Fk",

```

```

        <input type="submit"
value="Simpan" class="btn btn-primary"
/>
    </div>
</div>
}

```

```
@section Scripts {
```

```
@Scripts.Render("~/bundles/jqueryval")
}
```

Penempatan_Index_View

```
@using PagedList.Mvc;
@using PagedList;
```

```
@model
```

```
IPagedList<inventori.Models.Penempatan
>
```

```
@{
```

```
    ViewBag.Title = "Penempatan";
```

```
}
```

```
<link href="~/Content/home.css"
rel="stylesheet" />
```

```
<link href="~/Content/home1.css"
rel="stylesheet" />
```

```
<center><h2>Daftar Barang Penempatan
</h2></center>
```

```
<p>
```

```
    <a class="btn icon-btn btn-
primary"
```

```
href="http://localhost:13438/Penempera
ns/Create">
```

```
    <span class="glyphicon btn-
glyphicon glyphicon-plus img-circle
text-danger"></span>Tambah
```

```
Penempatan</a>
```

```
</p>
```

```
@using (Html.BeginForm("Index",
"Penempatans", FormMethod.Get))
```

```
{
```

```
    <table>
```

```
        <tr>
```

```
            <th>Pencarian: </th>
```

```
<th>@Html.TextBox("SearchString",
null, htmlAttributes: new { @class =
"form-control", style =
"color:default" })</th>
```

```
        <th><input type="submit"
value="Cari" class=" btn btn-primary
form-control" /></th>
```

```

null, "Semua", htmlAttributes:
new { @class = "form-control",
style = "color:default" })
</th>
<th
style="width:200px">
@Html.DropDownList("Status_Fk",
null, "Semua", htmlAttributes:
new { @class = "form-control",
style = "color:default" })
</th>
<th>
<input style=""
type="submit" value="Filter"
class=" btn btn-primary form-
control" />
</th>
</tr> @*//TAMBAHAN
CODING UNTUK MEMBUAT PENCARIAN
SESUAI NAMA BARANG DAN LOKASI
BARANG*@
</table>
}
<p></p>
<table class="table">
<tr>
<th>No</th>
<th>Nama Barang</th>
<th>Kode Barang</th>
<th>Lokasi
Penempatan</th>
<th> Status Barang</th>
<th>Jumlah Barang</th>
<th> Keterangan
Penempatan</th>
<th>Tanggal
Penempatan</th>
<th>Action</th>
</tr>
@{
var pageNumber =
Model.PageCount <
Model.PageNumber ? 0 :
Model.PageNumber;

@Html.ActionLink(" ",
"Details", new { id =
item.Penempatan_id },
htmlAttributes: new { @class =
"btn btn-primary glyphicon
glyphicon-eye-open" })
@Html.ActionLink(" ",
"Delete", new { id =
item.Penempatan_id },
htmlAttributes: new { @class =
var rowperpage =
Model.PageSize;
int rowNo = 0;
if (pagenumber > 1 )
{
int angka = pagenumber -
1;
rowNo = rowperpage * angka;
}
}
@foreach (var item in Model)
{
<tr>
<td>@(rowNo += 1)</td>
<td>
@Html.DisplayFor(modelItem
=> item.Barang>Nama_Barang)
</td>
<td>
@Html.DisplayFor(modelItem
=> item.Barang.Kode_Barang)
</td>
<td>
@Html.DisplayFor(modelItem
=>
item.Lokasi_Barang>Nama_Lokasi_Barang)
</td>
<td>
@Html.DisplayFor(modelItem
=>
item.Status_Barang>Nama_Status_Barang)
</td>
<td>
@Html.DisplayFor(modelItem
=> item.Jumlah)
</td>
<td>
@Html.DisplayFor(modelItem
=> item.Keterangan)
</td>
<td>
@{
DateTime tanggal =
item.Tanggal ?? DateTime.Now;
var tanggal2 =
tanggal.ToString("dd/MM/yyyy");
}
@Html.DisplayFor(modelItem
=> tanggal2)
</td>
<td>
@Html.ActionLink(" ",
"Edit", new { id = item.Penempatan_id
}, htmlAttributes: new { @class = "btn
btn-primary glyphicon glyphicon-edit"
})
}
}
}

```



```

        "btn btn-danger glyphicon
        glyphicon-trash" })
        </td>
    </tr>
}
</table>
<div id="container"
style="margin-left: 20px">
    <p></p>
    <p></p>
    <div class="pagination"
style="margin-left: 400px">
        Page @(pagenumber)
        of @Model.PageCount
    @Html.PagedListPager(Model, page
=> Url.Action("Index", new { page
}))
    </div>
</div>

```

Penempatan_Create_View

```

@model
inventori.Models.Penempatan
@{
    ViewBag.Title = "Tambah Data
Penempatan";
    var message =
ViewBag.Message; //validasi
Message
}
<link href="~/Content/home.css"
rel="stylesheet" />
<link href="~/Content/home1.css"
rel="stylesheet" />
<h2>Form Tambah Penempatan</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <hr />
    @Html.ValidationSummary(true, "",
new { @class = "text-danger" })
    @*Validasi Message*@
        @if (message != null)
        {
            model.Status_Fk,
            ViewBag.Kategori_Barang_Fk as
            SelectList, "Pilih Status
            Barang", new { @class = "form-
            control" })
        @*@Html.DropDownList("Status_Fk",
        null, htmlAttributes: new {
        @class = "form-control" })*@

```

```

        <div class="alert-
        danger">@message</div>
    @*Validasi Message*@
    }
    <div class="form-group">
        @Html.LabelFor(model =>
        model.Barang_Fk, "Nama Barang",
        htmlAttributes: new { @class =
        "control-label col-md-2" })
        <div class="col-md-10">
            @*@Html.DropDownListFor(model =>
            model.Barang_Fk,
            ViewBag.Kategori_Barang_Fk as
            SelectList, "Pilih Nama Barang", new {
            @class = "form-control"},)*@
            @Html.DropDownList("Barang_Fk", null,
            "Pilih Nama Barang",htmlAttributes:
            new { @class = "form-control", Style=
            "width: 500px, hight: 1000px" })
            @Html.ValidationMessageFor(model =>
            model.Barang_Fk, "", new { @class =
            "text-danger" })
        </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model =>
            model.Lokasi_Fk, "Lokasi Penempatan",
            htmlAttributes: new { @class =
            "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownListFor(model =>
                model.Lokasi_Fk,
                ViewBag.Kategori_Barang_Fk as
                SelectList, "Pilih Lokasi Penempatan",
                new { @class = "form-control" })
                @*@Html.DropDownList("Lokasi_Fk",
                null, htmlAttributes: new { @class =
                "form-control" })*@
                @Html.ValidationMessageFor(model =>
                model.Lokasi_Fk, "", new { @class =
                "text-danger" })
            </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model =>
            model.Status_Fk, "Status Barang",
            htmlAttributes: new { @class =
            "control-label col-md-2" })
            <div class="col-md-10">
                @Html.DropDownListFor(model =>

```

```

@Html.ValidationMessageFor(model
=> model.Status_Fk, "", new {
@class = "text-danger" })
</div>
</div>
<div class="form-group">
@Html.LabelFor(model
=> model.Jumlah, "Jumlah Barang",
htmlAttributes: new { @class =
"control-label col-md-2" })
<div class="col-md-
10">

@Html.EditorFor(model =>
model.Jumlah, new {
htmlAttributes = new { @class =
"form-control" } })

@Html.ValidationMessageFor(model
=> model.Jumlah, "", new { @class
= "text-danger" })
</div>
</div>

<div class="form-group">
@Html.LabelFor(model
=> model.Keterangan, "Keterangan
Penempatan", htmlAttributes: new
{ @class = "control-label col-md-
2" })

<div class="col-md-
10">

@Html.TextAreaFor(m
=> m.Keterangan, new { @class =
"form-control" })

@Html.ValidationMessageFor(model
=> model.Keterangan, "", new {
@class = "text-danger" })
</div>
<div class="form-group">
@Html.LabelFor(model
=> model.Tanggal, "Tanggal
Penempatan", htmlAttributes: new {
@class = "control-label col-md-2"
})

<div class="form-group">
@Html.LabelFor(model
=> model.Barang_Fk, "Nama
Barang", htmlAttributes: new {
@class = "control-label col-md-2"
})

<div class="col-md-
10">

<div class="col-md-10">
@Html.EditorFor(model
=> model.Tanggal, new { htmlAttributes
= new { @class = "form-control" } })

@Html.ValidationMessageFor(model
=> model.Tanggal, "", new { @class =
"text-danger" })
</div>
</div>
<div class="form-group">
<div class="col-md-offset-
2 col-md-10">
<a
href="http://localhost:13438/Penempata
ns" class="btn btn-
primary">Kembali</a>
<input type="submit"
value="Tambah" class="btn btn-primary"
/>
</div>
</div>
</div>
}
@section Scripts {
@Scripts.Render("~/bundles/jqueryval")
}

Penempatan_Edit_View

@model inventori.Models.Penempatan
@{
    ViewBag.Title = "Edit";
    var message = ViewBag.Message;
    //validasi Message
}
<link href="~/Content/home.css"
rel="stylesheet" />
<link href="~/Content/home1.css"
rel="stylesheet" />
<h2>Ubah Data Penempatan</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true,
"", new { @class = "text-danger" })
        @if (message != null)
        {
            <div class="alert-
danger">@message</div>
            @*Validasi Message*@
        }
        @Html.HiddenFor(model =>
model.Penempatan_id)

```

```

@Html.DropDownList("Barang_Fk",
null, htmlAttributes: new {
@class = "form-control" })

@Html.ValidationMessageFor(model
=> model.Barang_Fk, "", new {
@class = "text-danger" })
</div>
<div class="form-group">
@Html.LabelFor(model
=> model.Lokasi_Fk, "Lokasi
Penempatan", htmlAttributes: new
{ @class = "control-label col-md-
2" })
<div class="col-md-
10">

@Html.DropDownList("Lokasi_Fk",
null, htmlAttributes: new {
@class = "form-control" })

@Html.ValidationMessageFor(model
=> model.Lokasi_Fk, "", new {
@class = "text-danger" })
</div>
<div class="form-group">
@Html.LabelFor(model
=> model.Status_Fk, "Status
Penempatan", htmlAttributes: new
{ @class = "control-label col-md-
2" })
<div class="col-md-
10">

@Html.DropDownList("Status_Fk",
null, htmlAttributes: new {
@class = "form-control" })

@Html.ValidationMessageFor(model
=> model.Status_Fk, "", new {
@class = "text-danger" })
</div>
<div class="form-group">
@Html.LabelFor(model
=> model.Jumlah, "Jumlah Barang",
)
}
@section Scripts {
@Scripts.Render("~/bundles/jquery
val")
}

htmlAttributes: new {
@class = "control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model
=> model.Jumlah, new { htmlAttributes
= new { @class = "form-control" } })
@Html.ValidationMessageFor(model =>
model.Jumlah, "", new { @class =
"text-danger" })
</div>
</div>
<div class="form-group">
@Html.LabelFor(model =>
model.Keterangan, "Keterangan
Penempatan", htmlAttributes: new {
@class = "control-label col-md-2" })
<div class="col-md-10">

@*@Html.EditorFor(model =>
model.Keterangan, new { htmlAttributes
= new { @class = "form-control" } })*@
@Html.TextAreaFor(m =>
m.Keterangan, new { @class = "form-
control" })

@Html.ValidationMessageFor(model =>
model.Keterangan, "", new { @class =
"text-danger" })
</div>
</div>
<div class="form-group">
@Html.LabelFor(model =>
model.Tanggal, "Tanggal Penempatan",
htmlAttributes: new { @class =
"control-label col-md-2" })
<div class="col-md-10">
@Html.EditorFor(model
=> model.Tanggal, new { htmlAttributes
= new { @class = "form-control" } })

@Html.ValidationMessageFor(model =>
model.Tanggal, "", new { @class =
"text-danger" })
</div>
</div>
<div class="form-group">
<div class="col-md-offset-
2 col-md-10">
<a
href="http://localhost:13438/Penempera
ns" class="btn btn-
primary">Kembali</a>
<input type="submit"
value="Simpan" class="btn btn-primary"
/>
</div>
</div>

```

Mutasi_Barang_Index_View

```
@using PagedList.Mvc;
@using PagedList;
@*di atas untuk nambahin paging
di list*@

@model
IPagedList<inventori.Models.Mutasi
i_Barang>
@{
    ViewBag.Title = "Mutasi";
}
<link href="~/Content/home.css"
rel="stylesheet" />
<link href="~/Content/home1.css"
rel="stylesheet" />
<center><h2>Daftar Barang Mutasi
</h2></center>
<p>
    <a class="btn icon-btn btn-
primary"
href="http://localhost:13438/Muta
si_Barang/Create">
        <span class="glyphicon
btn-glyphicon glyphicon-plus img-
circle text-danger"></span>Mutasi
Barang
    </a>
</p>
@using (Html.BeginForm("Index",
"Mutasi_Barang", FormMethod.Get))
{
<table>
    <tr>
        <th>Pencarian: </th>

        <th>@Html.TextBox("SearchString",
null, htmlAttributes: new {
@class = "form-control", style =
"color:default" })</th>
        <th><input type="submit"
value="Cari" class=" btn btn-
primary form-control" /></th>
        <th style="width:200px">

        @Html.DropDownList("Lokasi_Baru_F
k", null, "Semua",
htmlAttributes: new { @class =

        @Html.DisplayFor(modelItem =>
item.Penempatan.Barang>Nama_Baran
g)
        @*+

        @Html.DisplayFor(modelItem =>

"form-control", style =
"color:default" })
        </th>
        <th>
            <input style=""
type="submit" value="Filter" class="
btn btn-primary form-control" />
        </th>
    </tr>
    @*//TAMBAHAN CODING
UNTUK MEMBUAT PENCARIAN SESUAI NAMA
BARANG DAN LOKASI BARANG*@
</table>
}
<p></p>
<table class="table">
    <tr>
        <th>No</th>
        <th>
            Nama Barang Mutasi
        </th>
        <th>
            Lokasi Lama
        </th>
        <th>
            Lokasi Baru
        </th>
        <th>
            Jumlah Barang
        </th>
        <th>
            Tanggal Mutasi
        </th>
        <th>
            Keterangan Mutasi
        </th>
        <th>Action</th>
    </tr>
    @{
        var pagenumber =
Model.PageCount < Model.PageNumber ? 0
: Model.PageNumber;
        var rowperpage =
Model.PageSize;
        int rowNo = 0;
        if (pagenumber > 1)
        {
            int angka = pagenumber -
1;
            rowNo = rowperpage *
angka;
        }
    }
    @foreach (var item in Model) {
        <tr>
            <td>@(rowNo += 1)</td>
            <td>
```

```

item.Penempatan.Lokasi_Barang.Nam
a_Lokasi_Barang)*@
    </td>
    <td>

@Html.DisplayFor(modelItem =>
item.Penempatan.Lokasi_Barang.Nam
a_Lokasi_Barang)
    </td>
    <td>

@Html.DisplayFor(modelItem =>
item.Lokasi_Barang>Nama_Lokasi_Ba
rang)
    </td>
    <td>

@Html.DisplayFor(modelItem =>
item.Jumlah)
    </td>
    <td>
        @{
            DateTime tanggal
= item.Tanggal ?? DateTime.Now;
            var tanggal2 =
tanggal.ToString("dd/MM/yyyy");
        }

@Html.DisplayFor(modelItem =>
tanggal2)
    </td>
    <td>

@Html.DisplayFor(modelItem =>
item.Keterangan)
    </td>
    <td>
        @if
(Request.IsAuthenticated)
        {
            <a>@Html.ActionLink(" ", "Edit",
new { id = item.Mutasi_id },
htmlAttributes: new { @class =
"btn btn-primary glyphicon
glyphicon-edit" })</a>
        }

            <a>
@Html.ActionLink(" ", "Details",
@Html.LabelFor(model
=> model.Penempatan_Barang_id,
"Nama Barang Mutasi",
htmlAttributes: new { @class =
"control-label col-md-2" })
    <div class="col-md-
10">
new { id = item.Mutasi_id },
htmlAttributes: new { @class = "btn
btn-primary glyphicon glyphicon-eye-
open" })</a>
        @if
(Request.IsAuthenticated)
        {
            <a>@Html.ActionLink("
", "Delete", new { id = item.Mutasi_id
}, htmlAttributes: new { @class = "btn
btn-danger glyphicon glyphicon-trash"
})</a>
        }
    </td>
</tr>
}
</table>
<div id="container" style="margin-
left: 20px">
    <p></p>
    <p></p>
    <div class="pagination"
style="margin-left: 400px">
        Page @(Model.PageCount <
Model.PageNumber ? 0 :
Model.PageNumber)
        of @Model.PageCount
@Html.PagedListPager(Model, page =>
Url.Action("Index", new { page }))
    </div>
</div>

```

Mutasi_Barang_Create_View

```

@model inventori.Models.Mutasi_Barang
@{
    ViewBag.Title = "Tambah Mutasi";
    var message = ViewBag.Message;
}
<link href="~/Content/home.css"
rel="stylesheet" />
<link href="~/Content/home1.css"
rel="stylesheet" />
<h2>Form Tambah Mutasi</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <hr />
        @Html.ValidationSummary(true,
"", new { @class = "text-danger" })
        @if (message != null)
        {
            <div class="alert-
danger">@message</div>
        }
        <div class="form-group">

```

```

@Html.DropDownListFor(model =>
model.Penempatan_Barang_id,
ViewBag.Kategori_Barang_Fk as
SelectList, "Pilih Nama Barang",
new { @class = "form-control" })

@Html.ValidationMessageFor(model
=> model.Penempatan_Barang_id,
"", new { @class = "text-danger"
})

</div>
</div>
<div class="form-group">
    @Html.LabelFor(model
=> model.Lokasi_Baru_Fk, "Lokasi
Baru", htmlAttributes: new {
@class = "control-label col-md-2"
})
    <div class="col-md-
10">

@Html.DropDownListFor(model =>
model.Lokasi_Baru_Fk,
ViewBag.Kategori_Barang_Fk as
SelectList, "Pilih Lokasi
Mutasi", new { @class = "form-
control" })

@Html.ValidationMessageFor(model
=> model.Lokasi_Baru_Fk, "", new
{ @class = "text-danger" })
    </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model
=> model.Tanggal, "Tanggal
Mutasi",htmlAttributes: new {
@class = "control-label col-md-2"
})
        <div class="col-md-
10">

@Html.EditorFor(model =>
model.Tanggal, new {
htmlAttributes = new { @class =
"form-control" } })

@Html.ValidationMessageFor(model
=> model.Tanggal, "", new {
@class = "text-danger" })
    </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model =>
model.Jumlah, "Jumlah
Barang",htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model =>
model.Jumlah, new { htmlAttributes =
new { @class = "form-control" } })

@Html.ValidationMessageFor(model =>
model.Jumlah, "", new { @class =
"text-danger" })
        </div>
        </div>
        <div class="form-group">
            @Html.LabelFor(model =>
model.Keterangan, "Keterangan
Mutasi",htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-10">

@*@Html.EditorFor(model =>
model.Keterangan, new { htmlAttributes
= new { @class = "form-control" } })*@
            @Html.TextAreaFor(m =>
m.Keterangan, new { @class = "form-
control" })

@Html.ValidationMessageFor(model =>
model.Keterangan, "", new { @class =
"text-danger" })
        </div>
        </div>
        <div class="form-group">
            <div class="col-md-offset-
2 col-md-10">
                <a
href="http://localhost:13438/Mutasi_Ba
rang" class="btn btn-
primary">Kembali</a>
                <input type="submit"
value="Tambah" class="btn btn-primary"
/>
            </div>
        </div>
    </div>
}
@section Scripts {

@Scripts.Render("~/bundles/jqueryval")
}

```

Mutasi_Barang_Edit_View

```
@model
inventori.Models.Mutasi_Barang
@{
    ViewBag.Title = "Edit";
    var message =
ViewBag.Message;
}
<link href="~/Content/home.css"
rel="stylesheet" />
<link href="~/Content/home1.css"
rel="stylesheet" />
<h2>Ubah Data Mutasi</h2>
@using (Html.BeginForm())
{
    @Html.AntiForgeryToken()
    <div class="form-horizontal">
        <hr />

@Html.ValidationSummary(true, "",
new { @class = "text-danger" })
        @if (message != null)
        {
            <div class="alert-
danger">@message</div>
        }
        @Html.HiddenFor(model =>
model.Mutasi_id)

        <div class="form-group">
            @Html.LabelFor(model
=> model.Penempatan_Barang_id,
"Nama Barang Mutasi",
htmlAttributes: new { @class =
"control-label col-md-2" })
            <div class="col-md-
10">

@Html.DropDownList("Penempatan_Ba
rang_id", null, htmlAttributes:
new { @class = "form-control" })

@Html.ValidationMessageFor(model
=> model.Penempatan_Barang_id,
"", new { @class = "text-danger"
})

            </div>
        </div>
        @*<div class="form-
group">
            @Html.LabelFor(model
=> model.Penempatan.Lokasi_Fk,
"Lokasi Lama", htmlAttributes:
new { @class = "control-label
col-md-2" })
```

```
<div class="col-md-10">

@Html.Editor("Lokasi_Fk", null,
htmlAttributes: new { @class = "form-
control" })

@Html.ValidationMessageFor(model =>
model.Penempatan.Lokasi_Fk, "", new {
@class = "text-danger" })
        </div>
    </div>*@
    <div class="form-group">
        @Html.LabelFor(model =>
model.Lokasi_Baru_Fk, "Lokasi Baru",
htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">

@Html.DropDownList("Lokasi_Baru_Fk",
null, htmlAttributes: new { @class =
"form-control" })

@Html.ValidationMessageFor(model =>
model.Lokasi_Baru_Fk, "", new { @class
= "text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model =>
model.Jumlah,"Jumlah Barang",
htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
            @Html.EditorFor(model
=> model.Jumlah, new { htmlAttributes
= new { @class = "form-control" } })

@Html.ValidationMessageFor(model =>
model.Jumlah, "", new { @class =
"text-danger" })
        </div>
    </div>
    <div class="form-group">
        @Html.LabelFor(model =>
model.Tanggal,"Tanggal Mutasi",
htmlAttributes: new { @class =
"control-label col-md-2" })
        <div class="col-md-10">
@Html.EditorFor(model =>
model.Tanggal, new { htmlAttributes =
new { @class = "form-control" } })

@Html.ValidationMessageFor(model =>
model.Tanggal, "", new { @class =
"text-danger" })
        </div>
    </div>
```

```

        <div
class="form-group">
    @Html.LabelFor(model =>
model.Keterangan, "Keterangan
Mutasi" ,htmlAttributes: new {
@class = "control-label col-md-2"
})
        <div class="col-md-
10">

@Html.TextAreaFor(m =>
m.Keterangan, new { @class =
"form-control" })

@Html.ValidationMessageFor(model
=> model.Keterangan, "", new {
@class = "text-danger" })
        </div>
    </div>
    <div class="form-group">
        <div class="col-md-
offset-2 col-md-10">
            <a
href="http://localhost:13438/Muta
si_Barang" class="btn btn-
primary">Kembali</a>
            <input
type="submit" value="Simpan"
class="btn btn-primary" />
        </div>
    </div>
</div>
}

@section Scripts {

@Scripts.Render("~/bundles/jquery
val")
}

```