LAMPIRAN

Lampiran I. Hasil wawancara

Berikut ini merupakan tabel hasil wawancara yang dilakukan dengan Bapak Sugeng Siswoyo dan Ibu Weningsari Isti Wahyuni selaku pemilik usaha, serta Ibu Tri Wahyudiati selaku sekretaris SG Mandiri. Wawancara dilakukan kepada narasumber-narasumber tersebut karena mereka benar-benar mengetahui permasalahan yang sedang dihadapi dan dapat memberikan masukan terhadap sistem yang akan dikembangkan.

Nama	Sugeng Siswoyo
Jabatan	Pemilik Usaha
Tanggal Wawancara	20 Oktober 2015
Jam Wawancara	09.00
Tempat	UD SG Mandiri
Daftar Pertanyaan 1. Bagaimanakah sistem administrasi yang saat ini berjalan di SG Mandiri terkait dengan penggunaan teknologi? Jawaban : Saat ini proses-proses yang dilakukan belum memakai komputer sama sekali, semua prosesnya dilakukan manual mulai dari transaksi hingga mencatat data pegawai dan pelanggan. Untuk transaksi sudah	

menggunakan smartphone sebagai media komunikasi, biasa menggunakan *Whatsapp* dan *BBM*.

2. Bagaimana proses transaksi jual dan beli yang dilakukan?

   Jawaban :

   Untuk transaksi beli, pelanggan yang datang ditimbang barangnya oleh penimbang dan hasil timbangan dilaporkan ke sekretaris lalu dihitung bayaran yang akan diberikan. Barang dari pelanggan kami kumpulkan di gudang, kalau sudah memenuhi kuota untuk pengiriman maka kami hubungi pihak pembeli untuk dilakukan penjadwalan.

3. Berapa kali dalam sehari dapat melakukan transaksi?

   Jawaban :

   Sekitar 50 hingga 60 kali transaksi pembelian.

Nama	Weningsari Isti Wahyuni
Jabatan	Pemilik Usaha
Tanggal Wawancara	20 Oktober 2015
Jam Wawancara	10.00
Tempat	UD SG Mandiri

Daftar Pertanyaan

1. Berapa banyak karyawan yang bekerja di SG Mandiri?

   Yang bekerja saat ini total ada sekitar 27 karyawan.

2. Bagaimana proses presensi karyawan yang dilakukan?

   Jawaban :

   Karyawan yang hadir laporan ke sekretaris, lalu ditulis di buku kehadiran karyawan. Setiap hari sabtu data kehadiran tersebut direkap dan dihitung gaji yang didapat pegawai, dihitung dari gaji harian dikalikan dengan jumlah hadir dalam satu minggu.

3. Bagaimana proses kredit yang saat ini dilakukan?

   Jawaban :

   Siapa yang mau melakukan kredit lapor kepada sekretaris, lalu disampaikan kepada kami untuk dibelikan barangnya. Angsuran biasanya dipotong dari gaji atau setoran tapi jumlah dan frekuensi pembayarannya tidak menentu.

4. Siapa saja yang bisa melakukan kredit?

   Jawaban :

   Saat ini yang dapat melakukan kredit hanyalah karyawan dan penyetor tetap karena selain sasaran kami adalah kalangan menengah kebawah tapi juga untuk pembayaran angsuran dapat lebih terpantau.

Nama	Tri Wahyudiati
Jabatan	Sekretaris
Tanggal Wawancara	27 Oktober 2015

Jam Wawancara	09.00
Tempat	UD SG Mandiri
Daftar Pertanyaan 1. Apakah anda sudah biasa mengoperasikan komputer? Jawaban : Biasa menggunakan di rumah. Dan saya bisa belajar kalau memang diperlukan. 2. Program apa saja yang biasa anda gunakan di dalam komputer? Jawaban : Microsoft word, excel, dan google chrome. 3. Dapatkah saya mendapatkan data-data mengenai aktifitas-aktifitas yang dilakukan di SG Mandiri? Jawaban : Iya, kami dapat memberikan data-data yang dibutuhkan 4. Apakah ada kesulitan khusus dalam mengoperasikan komputer seperti tidak bisa membaca teks dengan jelas dan membedakan warna? Jawaban : Tidak ada.	

Lampiran II. Source Code

1. Model *Class* Pelanggan

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Pelanggan
    {
        public Pelanggan()
        {
            this.Transaksi = new HashSet<Transaksi>();
        }

        public int Id_Pelanggan { get; set; }
        public string Nama { get; set; }
        public string Alamat { get; set; }
        public string NoTelp { get; set; }
        public Nullable<int> Id_Jenis { get; set; }
        public virtual Jenis_Pelanggan Jenis_Pelanggan { get; set; }
        public virtual ICollection<Transaksi> Transaksi { get; set; }
    }
}
```

2. Model *Class* Pegawai

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Pelanggan
    {
        public Pelanggan()
        {
            this.Transaksi = new HashSet<Transaksi>();
        }
        public int Id_Pelanggan { get; set; }
        public string Nama { get; set; }
        public string Alamat { get; set; }
        public string NoTelp { get; set; }
        public Nullable<int> Id_Jenis { get; set; }
        public virtual Jenis_Pelanggan Jenis_Pelanggan { get; set; }
        public virtual ICollection<Transaksi> Transaksi { get; set; }
    }
}
```

3. Model *Class* Barang

```
namespace SG_SIM.Models

{
    using System;
    using System.Collections.Generic;
    public partial class Barang
    {
        public Barang()
        {
            this.Transaksi_Details = new HashSet<Transaksi_Details>();
        }

        public int Id_Barang { get; set; }
        public string Nama_Barang { get; set; }
        public Nullable<decimal> HargaJual { get; set; }
        public Nullable<decimal> HargaBeli1 { get; set; }
        public Nullable<decimal> HargaBeli2 { get; set; }
        public Nullable<decimal> HargaBeli3 { get; set; }
        public Nullable<int> Id_Kategori { get; set; }

        public virtual Kategori_Barang Kategori_Barang { get; set; }
        public virtual ICollection<Transaksi_Details> Transaksi_Details {
get; set; }
    }
}
```

4. Model *Class* Transaksi

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Transaksi
    {
        public Transaksi()
        {
            this.Transaksi_Details = new HashSet<Transaksi_Details>();
        }

        public long Id_Transaksi { get; set; }
        public Nullable<System.DateTime> Tanggal { get; set; }
        public string Tipe { get; set; }
        public Nullable<int> Id_Pelanggan { get; set; }
        public string Keterangan { get; set; }
        public Nullable<int> flag { get; set; }

        public virtual Pelanggan Pelanggan { get; set; }
        public virtual ICollection<Transaksi_Details> Transaksi_Details {
get; set; }
    }
}
```

5. Model *Class* Transaksi_Details

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Transaksi_Details
    {
        public int Id_TransaksiDetails { get; set; }
        public Nullable<decimal> Nominal { get; set; }
        public Nullable<long> id_transaksi { get; set; }
        public Nullable<int> id_barang { get; set; }
        public Nullable<decimal> Harga_Satuan { get; set; }

        public virtual Barang Barang { get; set; }
        public virtual Transaksi Transaksi { get; set; }
    }
}
```

6. Model *Class* Presensi

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Presensi
    {
        public Presensi()
        {
            this.Presensi_Details = new HashSet<Presensi_Details>();
        }

        public long Id_Presensi { get; set; }
        public Nullable<System.DateTime> Tanggal { get; set; }

        public virtual ICollection<Presensi_Details> Presensi_Details {
get; set; }
    }
}
```

7. Model *Class* Presensi_Details

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Presensi_Details
    {
        public long Id_PresensiDetails { get; set; }
        public Nullable<long> Id_Presensi { get; set; }
        public Nullable<int> Id_Pegawai { get; set; }
        public bool isMasuk { get; set; }

        public virtual Presensi Presensi { get; set; }
        public virtual Pegawai Pegawai { get; set; }
    }
}
```

8. Model *Class* Pemohon

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Pemohon
    {
        public Pemohon()
        {
            this.Kredit = new HashSet<Kredit>();
        }

        public int Id_Pemohon { get; set; }
        public string Nama { get; set; }
        public string Alamat { get; set; }
        public string NoTelp { get; set; }

        public virtual ICollection<Kredit> Kredit { get; set; }
    }
}
```

9. Model *Class* Kredit

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;

    public partial class Kredit
    {
        public Kredit()
        {
            this.Angsuran = new HashSet<Angsuran>();
        }

        public int Id_Kredit { get; set; }
        public Nullable<int> Id_Pemohon { get; set; }
        public Nullable<System.DateTime> Tanggal { get; set; }
        public Nullable<decimal> Total_Nominal { get; set; }
        public string Keterangan { get; set; }
        public Nullable<int> StatusLunas { get; set; }

        public virtual ICollection<Angsuran> Angsuran { get; set; }
        public virtual Pemohon Pemohon { get; set; }
    }
}
```

10. Model *Class* Angsuran

```
namespace SG_SIM.Models
{
    using System;
    using System.Collections.Generic;
    public partial class Angsuran
    {
        public int Id_Angsuran { get; set; }
        public Nullable<int> Id_Kredit { get; set; }
        public Nullable<System.DateTime> Tanggal { get; set; }
        public Nullable<decimal> Nominal { get; set; }

        public virtual Kredit Kredit { get; set; }
    }
}
```

11. *Controller Class* Transaksi

```
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 SG_SIM.Models;
using PagedList;
using System.Data.Objects.SqlClient;

namespace SG_SIM.Controllers
{
    public class TransaksiController : Controller
    {
        private SG_DB_MAINEntities db = new SG_DB_MAINEntities();

        public ActionResult Index(string sortOrder, string CurrentSort,
int? page)
        {
            if (Session["UserName"] != null)
            {
                int pageSize = 30;
                int pageIndex = 1;
                pageIndex = page.HasValue ? Convert.ToInt32(page) : 1;

                ViewBag.CurrentSort = sortOrder;

                ViewBag.flag = 0;
                IPagedList<Transaksi> transaksi = null;

                transaksi = db.Transaksi.Include(b => b.Pelanggan).Where(v
=> SqlFunctions.DatePart("dayofyear", v.Tanggal) ==
SqlFunctions.DatePart("dayofyear", DateTime.Now) &&
SqlFunctions.DatePart("year", v.Tanggal) == SqlFunctions.DatePart("year",
DateTime.Now)).OrderBy(m => m.Id_Transaksi).ToPagedList(pageIndex,
pageSize);

                return View(transaksi);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        [HttpPost]
        public ActionResult SearchDaily(string sortOrder, string 
CurrentSort, int? page,string searchy)
        {
            int pageSize = 30;
            int pageIndex = 1;
            pageIndex = page.HasValue ? Convert.ToInt32(page) : 1;

            var datasearch = db.Transaksi.Where(v => 
SqlFunctions.DatePart("dayofyear", v.Tanggal) == 
SqlFunctions.DatePart("dayofyear", searchy) && 
SqlFunctions.DatePart("year", v.Tanggal) == SqlFunctions.DatePart("year", 
searchy)).OrderBy(m => m.Id_Transaksi).ToPagedList(pageIndex, pageSize);
            return PartialView("_RekapTransDaily", datasearch);
        }

        // GET: /Transaksi/

        // GET: /Transaksi/Details/5
        public ActionResult Details(long? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new 
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                var transdet = db.Transaksi_Details.Where(m => 
m.id_transaksi == id);

                Transaksi transaksi = db.Transaksi.Find(id);
                @ViewBag.Nama = transaksi.Pelanggan.Nama;
                @ViewBag.Tanggal = transaksi.Tanggal;

                if (transaksi == null)
                {
                    return HttpNotFound();
                }
                return View(transdet.ToList());
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        // GET: /Transaksi/Create
        public ActionResult Create()
        {
            if (Session["UserName"] != null)
            {
                ViewBag.Id_Pelanggan = new SelectList(db.Pelanggan, 
"Id_Pelanggan", "Nama");
                return View();
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        // POST: /Transaksi/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="Id_Transaksi,Tanggal,Tipe,Id_Pelanggan")] Transaksi 
transaksi)
        {
            if (ModelState.IsValid)
            {
                db.Transaksi.Add(transaksi);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            ViewBag.Id_Pelanggan = new SelectList(db.Pelanggan, 
"Id_Pelanggan", "Nama", transaksi.Id_Pelanggan);
            return View(transaksi);
        }

        // GET: /Transaksi/Edit/5
        public ActionResult Edit(long? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new 
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                Transaksi transaksi = db.Transaksi.Find(id);
                if (transaksi == null)
                {
                    return HttpNotFound();
                }

                return View(transaksi);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }
        // POST: /Transaksi/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="Id_Transaksi,Tanggal,Tipe,Id_Pelanggan,Keterangan,flag"
)] Transaksi transaksi)
        {
                transaksi.flag = 0;
                db.Entry(transaksi).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");

        }

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

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

        protected override void Dispose(bool disposing)
        {
            if (disposing)
            {
                db.Dispose();
            }
            base.Dispose(disposing);
        }
        public ActionResult Transaction()
        {
            if (Session["UserName"] != null)
            {
                ViewBag.vId_Barang = new SelectList(db.Barang, "Id_Barang", 
"Nama_Barang");
                ViewBag.vId_Pelanggan = new SelectList(db.Pelanggan, 
"Id_Pelanggan", "Nama");
                return View();
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Transaction(List<Transaksi_Details> 
transdetails,Transaksi trans )
        {
            string tipe = trans.Tipe;

            trans.Pelanggan = db.Pelanggan.Find(trans.Id_Pelanggan);

            for (int o = 0; o < transdetails.Count; o++)
            {
                if (transdetails[o] == null)
                {
                    continue;
                }
                else {
                    Barang abc=db.Barang.Find(transdetails[o].id_barang);
                    var oko = transdetails[o].Nominal;
                    var abcd = 
db.Barang.Where(x=>x.Id_Barang==transdetails[o].id_barang);

                        if(tipe.Equals("1")){
                           transdetails[o].Harga_Satuan = abc.HargaJual;
                           // transdetails[o].Harga_Satuan = 
abcd.FirstOrDefault().HargaJual ;
                        }
                        else if(tipe.Equals("2")){
                           transdetails[o].Harga_Satuan = abc.HargaBeli1;
                           // transdetails[o].Harga_Satuan = 
abcd.FirstOrDefault().HargaBeli1;
                        }
                        else if(tipe.Equals("3")){
                           transdetails[o].Harga_Satuan = abc.HargaBeli2;
                           // transdetails[o].Harga_Satuan = 
abcd.FirstOrDefault().HargaBeli2;
                        }
                        else{
                           transdetails[o].Harga_Satuan = abc.HargaBeli3;
                          //  transdetails[o].Harga_Satuan = 
abcd.FirstOrDefault().HargaBeli3;
                        }
                     transdetails[o].Barang = abc;
                }

            }
            TempData["TransTemp"] = trans;
            TempData["TransDetTemp"] = transdetails;
            return RedirectToAction("ConfirmTrans");
        }
        public ActionResult ConfirmTrans()
        {
            if (TempData["TransTemp"] != null)
            {
                Transaksi trans = (Transaksi)TempData["TransTemp"];
                List<Transaksi_Details> transdet = 
(List<Transaksi_Details>)TempData["TransDetTemp"];
                ViewBag.ListTrans = transdet;
                return View(trans);
            }
            else
            {
                return RedirectToAction("NotFounded");
            }
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult ConfirmTrans(List<Transaksi_Details> 
transdetails, Transaksi trans)
        {
            trans.flag = 1;
            db.Transaksi.Add(trans);
            db.SaveChanges();
            long id = trans.Id_Transaksi;
            for (int o = 0; o < transdetails.Count; o++)
            {
                if (transdetails[o] == null)
                {
                    continue;
                }
                else {
                    transdetails[o].id_transaksi= id;
                    db.Transaksi_Details.Add(transdetails[o]);
                }

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

        public ActionResult NotFounded()
        {
            return View();
        }
        public ActionResult AllReport(string sortOrder, string CurrentSort, 
int? page)
        {
            if (Session["UserName"] != null)
            {
                int pageSize = 10;
                int pageIndex = 1;
                pageIndex = page.HasValue ? Convert.ToInt32(page) : 1;
                ViewBag.CurrentSort = sortOrder;

                ViewBag.flag = 0;
                IPagedList<Transaksi> tranz = null;

                tranz = db.Transaksi.Include(b => b.Pelanggan).Where(b =>
b.flag == 1).OrderBy(m => m.Tanggal).ToPagedList(pageIndex, pageSize);

                return View(tranz);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }
        }
        [HttpPost]
        public ActionResult SearchAll(string sortOrder, string CurrentSort,
int? page, string searchy,string searchy2)
        {
            int pageSize = 10;
            int pageIndex = 1;
            pageIndex = page.HasValue ? Convert.ToInt32(page) : 1;

            ViewBag.CurrentSort = sortOrder;

            ViewBag.flag = 0;
            IPagedList<Transaksi> tranz = null;

            tranz = db.Transaksi.Include(b => b.Pelanggan).Where(b =>
b.flag ==
1&&b.Pelanggan.Nama.Contains(searchy)&&b.Tipe.Contains(searchy2)).OrderBy(m
=> m.Tanggal).ToPagedList(pageIndex, pageSize);
            return PartialView("_RekapTransDaily", tranz);
        }

    }
}
```

12. *Controller Class* Presensi

```
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 SG_SIM.Models;
using System.Data.Objects.SqlClient;

namespace SG_SIM.Controllers
{
    public class PresensiController : Controller
    {
        private SG_DB_MAINEntities db = new SG_DB_MAINEntities();
        public ActionResult Index()
        {
            if (Session["UserName"] != null)
            {
                var absen = db.Presensi.Where(v =>
SqlFunctions.DatePart("dayofyear", v.Tanggal) ==
SqlFunctions.DatePart("dayofyear", DateTime.Now) &&
SqlFunctions.DatePart("year", v.Tanggal) == SqlFunctions.DatePart("year",
DateTime.Now));
                ViewBag.pegawai = db.Pegawai.ToList();
                ViewBag.Tanggalan = DateTime.Now.ToShortDateString();
                if (absen.FirstOrDefault() == null)
                {
                    return View();
                }
                else
                {
                    return RedirectToAction("RekapDaily");
                }
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        public ActionResult InsertNewDate(DateTime datey)
        {
            ViewBag.pegawai = db.Pegawai.ToList();
            ViewBag.Tanggalan = datey.ToShortDateString();
            return View();
        }
        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult InsertNewDate(List<Presensi_Details>
listnama,string tanggalan)
        {
            if (ModelState.IsValid)
            {
                DateTime Tanggalan =Convert.ToDateTime(tanggalan);
                Presensi abzen = new Presensi
                {
                    Tanggal = Tanggalan
                };
                db.Presensi.Add(abzen);
                db.SaveChanges();
                var absen = db.Presensi.Where(v =>
SqlFunctions.DatePart("dayofyear", v.Tanggal) ==
SqlFunctions.DatePart("dayofyear", Tanggalan) &&
SqlFunctions.DatePart("year", v.Tanggal) == SqlFunctions.DatePart("year",
Tanggalan));
                long id = absen.FirstOrDefault().Id_Presensi;

                for (int o = 0; o < listnama.Count; o++)
                {
                    if (listnama[o] == null)
                    {
                        continue;
                    }
                    else
                    {
                        listnama[o].Id_Presensi = id;
                        if (listnama[o].isMasuk == null)
                        {
                            listnama[o].isMasuk = false;
                        }
                        db.Presensi_Details.Add(listnama[o]);
                    }
                }
                db.SaveChanges();
                ModelState.Clear();
                return RedirectToAction("RekapDaily");
            }
            return HttpNotFound();
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult Index(List<Presensi_Details> listnama)
        {
            if (ModelState.IsValid)
            {
                Presensi abzen = new Presensi
                {
                    Tanggal = DateTime.Now
                };
                db.Presensi.Add(abzen);
                db.SaveChanges();
                var absen = db.Presensi.Where(v =>
SqlFunctions.DatePart("dayofyear", v.Tanggal) ==
SqlFunctions.DatePart("dayofyear", DateTime.Now) &&
SqlFunctions.DatePart("year", v.Tanggal) == SqlFunctions.DatePart("year",
DateTime.Now));
                long id = absen.FirstOrDefault().Id_Presensi;

                for (int o = 0; o < listnama.Count; o++)
                {
                    if (listnama[o] == null)
                    {
                        continue;
                    }
                    else {
                        listnama[o].Id_Presensi = id;
                        db.Presensi_Details.Add(listnama[o]); }
                }
                db.SaveChanges();
                ModelState.Clear();
                return RedirectToAction("RekapDaily");
            }
            return HttpNotFound();
        }
        public ActionResult Edit(int? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                var absoin = db.Presensi_Details.Where(v => v.Id_Presensi
== id);
                if (absoin == null)
                {
                    return HttpNotFound();
                }
                return View(absoin.ToList());
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        // 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]
        public ActionResult Edit(List<Presensi_Details> listbarng)
        {
            if (ModelState.IsValid)
            {
                for (int o = 0; o < listbarng.Count; o++)
                {
                    if (listbarng[o] == null)
                    {
                        continue;
                    }
                    else { db.Entry(listbarng[o]).State =
EntityState.Modified; }

                }
                db.SaveChanges();
                return RedirectToAction("RekapDaily","Presensi");
            }
            return RedirectToAction("Index","Home");
        }
        public ActionResult RekapDaily()
        {
            if (Session["UserName"] != null)
            {
                var absen = db.Presensi_Details.Where(v =>
SqlFunctions.DatePart("dayofyear", v.Presensi.Tanggal) ==
SqlFunctions.DatePart("dayofyear", DateTime.Now) &&
SqlFunctions.DatePart("year", v.Presensi.Tanggal) ==
SqlFunctions.DatePart("year", DateTime.Now));
                ViewBag.pegawai = db.Pegawai.ToList();
                if (absen.FirstOrDefault() != null)
                {
                    ViewBag.idPresensi =
absen.FirstOrDefault().Id_Presensi;
                }
                ViewBag.Tanggal = DateTime.Now.ToShortDateString();
                return View(absen.ToList());
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }
        [HttpPost]
        public ActionResult SearchDaily(string searchy)
        {

            var datasearch = db.Presensi_Details.Where(v =>
SqlFunctions.DatePart("dayofyear", v.Presensi.Tanggal) ==
SqlFunctions.DatePart("dayofyear", searchy) &&
SqlFunctions.DatePart("year", v.Presensi.Tanggal) ==
SqlFunctions.DatePart("year", searchy)).ToList();
            if (datasearch.FirstOrDefault() != null)
            {
                ViewBag.idPresensi =
datasearch.FirstOrDefault().Id_Presensi;

            }
            else
            {
                ViewBag.Tanggal = searchy;
                ViewBag.LinkTanggal=searchy;
            }
            ViewBag.Tanggal = searchy;
            return PartialView("_RekapDaily", datasearch);
        }

        public ActionResult Penggajian()
        {
            if (Session["UserName"] != null)
            {
                int days = DateTime.Now.DayOfWeek - DayOfWeek.Sunday;
                DateTime weekTotal = DateTime.Now.AddDays(-days-1);


                var joinquery = db.Pegawai.GroupJoin(db.Presensi_Details, k
=> k.Id_Pegawai, v => v.Id_Pegawai, (k, v) => new { IdPeg = k.Id_Pegawai,
NamaPeg = k.Nama, GajiPeg = k.Gaji, totalmsk = v.Where(x => x.isMasuk ==
true && x.Presensi.Tanggal >= weekTotal).Count() });

                List<JoinPenggajian> jopeng = new List<JoinPenggajian>();

                foreach (var item in joinquery)
                {
                    JoinPenggajian jp = new JoinPenggajian();
                    jp.Id_Pegawai = item.IdPeg;
                    jp.Nama = item.NamaPeg;
                    jp.Gaji = item.GajiPeg;
                    jp.TotalMasuk = item.totalmsk;

                    jopeng.Add(jp);
                }

                ViewBag.tanggalMulai = 
weekTotal.AddDays(1).ToShortDateString();

                ViewBag.JoinPenggajian = jopeng;
                return View();
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }
        public ActionResult DetailRekap(int? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new 
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                Pegawai pegawai = db.Pegawai.Find(id);
                if (pegawai == null)
                {
                    return HttpNotFound();
                }

                int days = DateTime.Now.DayOfWeek - DayOfWeek.Sunday;
                DateTime weekTotal = DateTime.Now.AddDays(-days-1);

                var joinqueryabsen = db.Presensi_Details.Where(x => 
x.Presensi.Tanggal >= weekTotal && x.Id_Pegawai == 
id).OrderBy(x=>x.Presensi.Tanggal).ToList();

                ViewBag.tanggalMulai = 
weekTotal.AddDays(1).ToShortDateString();

                ViewBag.AbsenDetail = joinqueryabsen;

                return View(pegawai);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }
        }
      }
```

13. *Controller Class* Kredit

```
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 SG_SIM.Models;

namespace SG_SIM.Controllers
{
    public class KreditController : Controller
    {
        private SG_DB_MAINEntities db = new SG_DB_MAINEntities();

        // GET: /Kredit/
        public ActionResult Index()
        {
            if (Session["UserName"] != null)
            {
                var kredit = db.Kredit.Include(k =>
k.Pemohon).Where(k=>k.StatusLunas==0);
                return View(kredit.ToList());
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        [HttpPost]
        public ActionResult SearchAll(string searchy,string searchy2)
        {
            int varo = Convert.ToInt32(searchy2);
            var kredit = db.Kredit.Where(a =>
a.Pemohon.Nama.Contains(searchy)||a.Keterangan.Contains(searchy));
            kredit = kredit.Where(a => a.StatusLunas == varo);
            return PartialView("_KreditPartial", kredit);
        }



        // GET: /Kredit/Details/5
        public ActionResult DetailAngsurans(int? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }

                Kredit bon = db.Kredit.Find(id);

                ViewBag.datakredit = bon;

                var angsuransa = db.Angsuran.Include(b => b.Kredit);
                angsuransa = angsuransa.Where(k => k.Id_Kredit == id);

                if (bon == null)
                {
                    return HttpNotFound();
                }
                return View(angsuransa);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        public ActionResult TambahAngsuran()
        {
            if (Session["UserName"] != null)
            {

                return View();
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }
        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult TambahAngsuran([Bind(Include =
"Id_Angsuran,Id_Kredit,Tanggal,Nominal")] Angsuran angsuran)
        {
            if (ModelState.IsValid)
            {
                db.Angsuran.Add(angsuran);
                db.SaveChanges();
                return RedirectToAction("DetailAngsurans", new { id =
angsuran.Id_Kredit });
            }

            return View(angsuran);
        }

        public ActionResult CreateAngsuran(int? id, int? nominal)
        {
            if (Session["UserName"] != null)
            {
                Angsuran model = new Angsuran();
                model.Id_Kredit = id;
                ViewBag.NominalKet = nominal;
                return View(model);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }



        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult CreateAngsuran([Bind(Include =
"Id_Angsuran,Id_Kredit,Tanggal,Nominal")] Angsuran angsuran,decimal
Tempdata)
        {
            if (ModelState.IsValid)
            {
                if (angsuran.Nominal > Tempdata)
                {
                    ViewData["Error"] = "Angsuran Melebihi Total";
                    ViewBag.NominalKet = Tempdata;
                    return View(angsuran);
                }
                else
                {
                    db.Angsuran.Add(angsuran);
                    db.SaveChanges();
                    return RedirectToAction("DetailAngsurans", new { id =
angsuran.Id_Kredit });
                }
            }

            return View(angsuran);
        }

        public ActionResult EditAngsuran(int? id, decimal totalnoms)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                Angsuran bon_history = db.Angsuran.Find(id);
                ViewBag.NominalKet = totalnoms;
                if (bon_history == null)
                {
                    return HttpNotFound();
                }
                return View(bon_history);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult EditAngsuran([Bind(Include =
"Id_Angsuran,Id_Kredit,Tanggal,Nominal")] Angsuran angsuran,decimal
Tempdata)
        {
            if (ModelState.IsValid)
            {
                if (angsuran.Nominal > Tempdata)
                {
                    ViewData["Error"] = "Angsuran Melebihi Total";
                    ViewBag.NominalKet = Tempdata;
                    return View(angsuran);
                }
                else if (angsuran.Nominal <= 0)
                {
                    ViewData["Error"] = "Angsuran harus lebih dari 0";
                    ViewBag.Id_Kredit = angsuran.Id_Kredit;
                    ViewBag.NominalKet = Tempdata;
                    return View(angsuran);
                }
                else
                {
                    db.Entry(angsuran).State = EntityState.Modified;
                    db.SaveChanges();
                    return RedirectToAction("DetailAngsurans", new { id =
angsuran.Id_Kredit });
                }

            }

            return View(angsuran);
        }

        public ActionResult Create()
        {
            if (Session["UserName"] != null)
            {
                ViewBag.Id_Pemohon = new SelectList(db.Pemohon,
"Id_Pemohon", "Nama");
                return View();
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        [HttpPost]
        [ValidateAntiForgeryToken]
        public ActionResult
Create([Bind(Include="Id_Kredit,Id_Pemohon,Tanggal,Total_Nominal,Keterangan
")] Kredit kredit)
        {
            if (ModelState.IsValid)
            {
                kredit.StatusLunas=0;
                db.Kredit.Add(kredit);
                db.SaveChanges();
                return RedirectToAction("Index");
            }

            ViewBag.Id_Pemohon = new SelectList(db.Pemohon, "Id_Pemohon",
"Nama", kredit.Id_Pemohon);
            return View(kredit);
        }

        // GET: /Kredit/Edit/5
        public ActionResult Edit(int? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                Kredit kredit = db.Kredit.Find(id);
                if (kredit == null)
                {
                    return HttpNotFound();
                }
                ViewBag.Id_Pemohon = new SelectList(db.Pemohon,
"Id_Pemohon", "Nama", kredit.Id_Pemohon);
                return View(kredit);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        // POST: /Kredit/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="Id_Kredit,Id_Pemohon,Tanggal,Total_Nominal,Keterangan")
] Kredit kredit)
        {
            if (ModelState.IsValid)
            {
                db.Entry(kredit).State = EntityState.Modified;
                db.SaveChanges();
                return RedirectToAction("Index");
            }
            ViewBag.Id_Pemohon = new SelectList(db.Pemohon, "Id_Pemohon",
"Nama", kredit.Id_Pemohon);
            return View(kredit);
        }

        public ActionResult Delete(int? id)
        {
            if (Session["UserName"] != null)
            {
                if (id == null)
                {
                    return new
HttpStatusCodeResult(HttpStatusCode.BadRequest);
                }
                Angsuran bon_history = db.Angsuran.Find(id);
                if (bon_history == null)
                {
                    return HttpNotFound();
                }
                return View(bon_history);
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }

        // POST: /Angsuran/Delete/5
        [HttpPost, ActionName("Delete")]
        [ValidateAntiForgeryToken]
        public ActionResult DeleteConfirmed(int id)
        {
            Angsuran bon_history = db.Angsuran.Find(id);
            var kk = bon_history.Id_Kredit;
            db.Angsuran.Remove(bon_history);
            db.SaveChanges();
            return RedirectToAction("DetailAngsurans", new { id = kk });
        }

        public ActionResult CreateAngsurans(int? id, int? nominal)
        {
            if (Session["UserName"] != null)
            {
                ViewBag.Id_Kredit = id;
                return View();
            }
            else
            {
                return RedirectToAction("Login", "Home");
            }

        }
        [HttpPost]
        public ActionResult CreateAngsurans([Bind(Include =
"Id_Angsuran,Id_Kredit,Tanggal,Nominal")] Angsuran angsuran, decimal
Tempdata)
        {
            if (ModelState.IsValid)
            {

                    db.Angsuran.Add(angsuran);
                    db.SaveChanges();
                    return RedirectToAction("DetailAngsurans", new { id =
angsuran.Id_Kredit });

            }

            return View(angsuran);
        }

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

14. *View Layout*

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8" />
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>@ViewBag.Title - SG Mandiri</title>
    @Styles.Render("~/Content/css")
    @Scripts.Render("~/bundles/modernizr")
</head>
<body>
    @if (@Session["UserName"] != null)
    {
        <div class="container body-content">
        <div class="navbar navbar-default">
            <div class="container">
                <div class="navbar-header">
                    <button type="button" class="navbar-toggle" data-
toggle="collapse" data-target=".navbar-collapse">
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                        <span class="icon-bar"></span>
                    </button>
                    @Html.ActionLink("SG MANDIRI", "Index", "Home", null,
new { @class = "navbar-brand" })
                </div>
                <div ></div>
                <div class="navbar-collapse collapse dropdown">
                    <ul class="nav navbar-nav">
                        @if (@Session["RoleUser"].Equals("1") ||
@Session["RoleUser"].Equals("2") || @Session["RoleUser"].Equals("3"))
                        {
                        <li>
                            <a href="#" data-toggle="dropdown"
class="dropdown-toggle" data-hover="dropdown">Transaksi<b
class="caret"></b></a>
                            <ul class="dropdown-menu">
                                <li>@Html.ActionLink("Transaksi",
"Transaction", "Transaksi")</li>
                                <li>@Html.ActionLink("Rekap Harian",
"Index", "Transaksi")</li>
                                <li>@Html.ActionLink("Rekap Semua",
"AllReport", "Transaksi")</li>
                            </ul>
                        </li>
                        }
                        @if (@Session["RoleUser"].Equals("1") ||
@Session["RoleUser"].Equals("2") || @Session["RoleUser"].Equals("3"))
                        {
                        <li>
                            <a href="#" data-toggle="dropdown"
class="dropdown-toggle" data-hover="dropdown">Kepegawaian<b
class="caret"></b></a>
                            <ul class="dropdown-menu">
                                <li>@Html.ActionLink("Presensi", "Index",
"Presensi")</li>
                                <li>@Html.ActionLink("Laporan Absen",
"RekapDaily", "Presensi")</li>
                                <li>@Html.ActionLink("Penggajian",
"Penggajian", "Presensi")</li>
                            </ul>
                        </li>
                        }
                        @if (@Session["RoleUser"].Equals("1") ||
@Session["RoleUser"].Equals("2") || @Session["RoleUser"].Equals("4"))
                        {
                        <li>
                            <a href="#" data-toggle="dropdown"
class="dropdown-toggle" data-hover="dropdown">Kredit<b
class="caret"></b></a>
                            <ul class="dropdown-menu">
                                <li>@Html.ActionLink("Daftar Kredit",
"Index", "Kredit")</li>
                                <li>@Html.ActionLink("Daftar Pemohon",
"Index", "Pemohon")</li>
                            </ul>
                        </li>
                        }
                        @if (@Session["RoleUser"].Equals("1") ||
@Session["RoleUser"].Equals("2"))
                        {
                        <li>
```

```
                            <a href="#" data-toggle="dropdown"
class="dropdown-toggle" data-hover="dropdown">Pengaturan<b
class="caret"></b></a>
                            <ul class="dropdown-menu">
                                <li>@Html.ActionLink("Data Pegawai",
"Index", "Pegawai")</li>
                                <li>@Html.ActionLink("Data Pelanggan",
"Index", "Pelanggan")</li>
                                <li>@Html.ActionLink("Data Barang",
"Index", "Barang")</li>
                                <li>@Html.ActionLink("Data Jenis
Pelanggan", "Index", "JenisPelanggan")</li>
                                <li>@Html.ActionLink("Data Kategori
Barang", "Index", "KategoriBarang")</li>
                            </ul>
                        </li>
                        }
                    </ul>
                    <div class="navbar-collapse collapse">
                        <ul class="nav navbar-nav navbar-right navbar-
default">
                            <li>
                                <a href="#" data-toggle="dropdown"
class="dropdown-toggle" data-hover="dropdown"><i class="glyphicon
glyphicon-cog" style="color:white"></i></a>
                                <ul class="dropdown-menu">
                                    <li>@Html.ActionLink("Menu Utama",
"Index", "Home")</li>
                                    @if (@Session["RoleUser"].Equals("1"))
                                    {
                                    <li>@Html.ActionLink("Data Pengguna",
"Index", "Pengguna")</li>
                                    }
                                    <li>@Html.ActionLink("Log Out",
"Logout", "Home")</li>
                                </ul>
                            </li>
                        </ul>
                    </div>

                </div>
            </div>
        </div>
        @RenderBody()
        <hr />

    </div>
    }
    else
    {
    <center>
        <h3>Mohon Log In Terlebih Dahulu</h3>
        @Html.ActionLink("Login", "Login", "Home")
    </center>
    }
    @Scripts.Render("~/bundles/jquery")
    @Scripts.Render("~/bundles/bootstrap")
    @RenderSection("scripts", required: false)
</body>
</html>
```

15. *View* Transaksi Transaction

```
@model List<SG_SIM.Models.Transaksi_Details>

@{
    ViewBag.Title = "Transaksi Pembelian";
}

<center><h2>Transaksi</h2></center>
<br />
@{
    SG_SIM.Models.Transaksi trans = new SG_SIM.Models.Transaksi();
    List<SG_SIM.Models.Transaksi_Details> transdet = new
List<SG_SIM.Models.Transaksi_Details>();
    }

        <form id="formval" name="formval">
            <table class="table">
                <tr>
                    <th>Nama</th>
                    <th>@Html.DropDownList("vId_Pelanggan", String.Empty)
</th>
                    <th></th>
                    <th>Tanggal</th>
                    <th><input class="text-box single-line" data-val="true"
data-val-date="The field Tanggal must be a date." data-val-required="The
field is required." id="vTanggal" name="vTanggal" type="date" value=""
/></th>
                </tr>
                <tr>
                    <td>Jenis Transaksi</td>
                    <td>
                        <select id="vTipe" name="vTipe">
                            <option value=""></option>
                            <option value="1">Penjualan</option>
                            <option value="2">Pembelian
(Bangkelan)</option>
                            <option value="3">Pembelian (Rumahan)</option>
                            <option value="4">Pembelian (Eceran)</option>
                        </select></td>
                    <td></td>
                    <td>Keterangan</td>
                    <td><input class="text-box single-line"
id="vKeterangan" name="vKeterangan" type="text" value="" /></td>
                </tr>
            </table>
        <br/><hr /><br/>
            <table class="table">
                <tr>
                    <th>Barang</th>
                    <th>@Html.DropDownList("vId_Barang", String.Empty)
                    </th>
                    <th></th>
                    <th>Nominal</th>
                    <th><input class="text-box single-line" data-val="true"
data-val-required="The field is required." id="vNominalkilo"
name="vNominalkilo" type="number" value="0" />&nbsp;Kg</th>
                    <th><input class="text-box single-line" data-val="true"
data-val-required="The field is required." id="vNominalgram"
name="vNominalgram" type="number" value="0" />&nbsp;gram</th>
                    <th>(1 Ons = 10 gram)</th>
                </tr>
            </table>
        </form>
<center><button id="tambahlist" type="button" class="btn btn-
info">Tambah</button></center>
<br />

@using (Html.BeginForm("Transaction", "Transaksi", FormMethod.Post))
{
    @Html.AntiForgeryToken()
    @Html.ValidationSummary(true)
    <table class="table table-striped" id="dataTable">
        <tr>
            <th>Nama Barang</th>
            <th>Berat</th>
            <th>Hapus</th>
        </tr>
    </table>
    <div id="datastatic">

    </div>
    <center><input type="submit" value="Simpan & Cetak" class="btn btn-
default" /></center>
}
@section Scripts{
    @Scripts.Render("~/bundles/jqueryval")
    <script language="javascript">
        $(document).ready(function () {
            // Validasi
            $("#formval").validate({
                rules: {
                    "vId_Pelanggan": {
                        required: true
                    },
                    "vTanggal": {
                        required: true
                    },
                    "vTipe": {
                        required: true
                    },
                    "vId_Barang": {
                        required: true
                    }
                },
                messages: {
                    "vId_Pelanggan": "Masukkan ID",
                    "vTanggal": "Masukkan ID",
                    "vTipe": "Masukkan ID",
                    "vId_Barang": "Masukkan ID",
                }
            })

            $("#formval").submit(function () {
                window.print();
            })

            //1. Add new row
            var flags = 0;
            $("#tambahlist").click(function (e) {
                e.preventDefault();
                if (!$("#formval").valid()) {
                    $validator.showErrors(messages);
                    return false;
                }
                else {
                    var idpelanggan=$("#vId_Pelanggan :selected").val();
                    var tanggal=$("#vTanggal").val();
                    var tipe=$("#vTipe :selected").val();
                    var keterangan=$("#vKeterangan").val();
                    var idbarang = $("#vId_Barang :selected").val();
                    var namabarang = $("#vId_Barang :selected").text();
                    var nominalkilo = parseFloat($("#vNominalkilo").val());
                    var nominalgram = parseFloat($("#vNominalgram").val());
                    var nominaltotal = (nominalkilo*1000) + nominalgram;

                    var queryhiddenstatic = '<input id="Tanggal"
name="Tanggal" type="hidden" value="' + tanggal +
                        '"/><input id="Tipe" name="Tipe" type="hidden"
value="' + tipe +
                        '"/><input id="Id_Pelanggan" name="Id_Pelanggan"
type="hidden" value="' + idpelanggan +
                        '"/><input id="Keterangan" name="Keterangan"
type="hidden" value="' + keterangan + '"/>';

                    var queryhidden = '<input id="[' + flags +
'].id_barang" name="[' + flags + '].id_barang" type="hidden" value="' +
idbarang + '" />';
                    queryhidden += '<input id="[' + flags + '].Nominal"
name="[' + flags + '].Nominal" type="hidden" step="0.01" value="' +
nominaltotal + '" />';

                    $("#datastatic").html(queryhiddenstatic);
                    $("#dataTable").append('<tr><td>' + queryhidden +
namabarang + '</td><td>' + nominaltotal + '</td><td><a
href="javascript:void(0);" class="removes">Hapus</a></td>');
                    flags++;
                }
            });
            // 2. Remove
            $("#dataTable").on('click', '.removes', function () {
                $(this).parent().parent().remove();
                flags--;
            });
        });
```

```
    </script>
     }
```

16. *View* Transaksi AllReport

```
@model PagedList.IPagedList<SG_SIM.Models.Transaksi>

@using PagedList.Mvc;
@{
    ViewBag.Title = "Laporan";
}

<h3>Laporan Semua Transaksi</h3>
@using (Ajax.BeginForm("SearchAll", "Transaksi",
    new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        LoadingElementId = "ajax-loader",
        UpdateTargetId = "listsearch",
    }))
{
     <table class="table">
        <tr>
            <th>
                <label>Filter Nama&nbsp;</label>
                <input type="text" id="searchy" name="searchy" />
                <select id="searchy2" name="searchy2">
                    <option value="">Semua</option>
                    <option value="1">Penjualan</option>
                    <option value="2">Pembelian (Bangkelan)</option>
                    <option value="3">Pembelian (Rumahan)</option>
                    <option value="4">Pembelian (Eceran)</option>
                </select>
                <input type="submit" value="Search" />
            </th>
        </tr>
    </table>
    <img id="ajax-loader" src="@Url.Content("~/Images/ajax-loader.gif")"
style="display:none" />
}
<table class="table" id="listsearch">
    <tr>
        <th>
            Nama
        </th>
        <th>
            Jenis Transaksi
        </th>
        <th>
            Keterangan
        </th>
        <th>Detail</th>
        <th>Hapus</th>
    </tr>

    @foreach (var item in Model)
    {
        <tr>
            <td>
                @Html.DisplayFor(modelItem => item.Pelanggan.Nama)
            </td>
            <td>
                @if(item.Tipe.Equals("1")){
                    @Html.Raw("Penjualan");
                }
                else if (item.Tipe.Equals("2"))
                {
                    @Html.Raw("Pembelian (Bangkelan)");
                }
                else if (item.Tipe.Equals("3"))
                {
                    @Html.Raw("Pembelian (Rumahan)");
                }
                else{
                    @Html.Raw("Pembelian (Eceran)");
                }

            </td>
            <td>
                @Html.DisplayFor(modelItem => item.Keterangan)
            </td>
            <td>
                @Html.ActionLink("Details", "Details", new { id =
item.Id_Transaksi })
            </td>
            <td>
                @Html.ActionLink("Hapus", "Edit", new { id =
item.Id_Transaksi })
            </td>
        </tr>
    }

</table>
<div id='Paging' style="text-align: left">
    @Html.PagedListPager(Model, page => Url.Action("Index", new { page }))
     </div>
```

17. *View* Presensi RekapDaily

```
@model IEnumerable<SG_SIM.Models.Presensi_Details>

<style>

</style>
@{
    ViewBag.Title = "Index";
}
<h3>Laporan Presensi Harian</h3>

@using (Ajax.BeginForm("SearchDaily", "Presensi",
    new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        LoadingElementId = "ajax-loader",
        UpdateTargetId = "listsearch",
    }))
{
    <table class="table">
        <tr>
            <th>
                <input data-val="true" data-val-date="The field Tanggal
must be a date." id="searchy" name="searchy" type="date"
value="@DateTime.Now.ToString("yyyy-MM-dd")" />&nbsp;&nbsp;<input
type="submit" value="Search" />
            </th>
        </tr>
    </table>
    <img id="ajax-loader" src="@Url.Content("~/Images/ajax-loader.gif")"
style="display:none" />
}
<div id="listsearch">
    <h4>Tanggal : @ViewBag.Tanggal</h4>
    @if (ViewBag.idPresensi != null) {
    @Html.ActionLink("Edit Absen", "Edit", new { id = ViewBag.idPresensi })
    }
    @if (ViewBag.LinkTanggal != null)
{
    @Html.ActionLink("Tambah", "InsertNewDate", new { datey =
ViewBag.LinkTanggal })
}
    <br /><br/>

    <table class="table table-bordered">
        <tr>
            <th>
                Nama
            </th>
            <th>
                Masuk
            </th>
        </tr>
        @foreach (var item in Model)
        {
            <tr>
                <td>
                    @Html.DisplayFor(modelItem => item.Pegawai.Nama)
                </td>
                <td>
                    @if (item.isMasuk == true)
                    {

                        @Html.Label("Hadir", new { @class = "textbiru" })
                    }
                    else
                    {
                        @Html.Label("Absen", new { @class = "textmerah" })
                    }
                </td>
            </tr>
        }
    </table>
    <br />
</div>
```

18. *View* Presensi InsertNewDate

```
@model List<SG_SIM.Models.Presensi_Details>

@{
    ViewBag.Title = "Index";
}

@{List<SG_SIM.Models.Pegawai> listnama = ViewBag.pegawai;}
            <h3>Presensi</h3>
            <h4>Tanggal : @ViewBag.Tanggalan</h4>
            @using (Html.BeginForm("InsertNewDate", "Presensi", 
FormMethod.Post))
            {
                @Html.AntiForgeryToken()
                <input id="tanggalan" name="tanggalan" type="hidden" 
value="@ViewBag.Tanggalan" />
                <table class="table table-bordered">
                    <tr>
                        <th>
                            Nama
                        </th>
                        <th>
                            Masuk
                        </th>
                    </tr>
                    @for (int a = 0; a < listnama.Count(); a++)
                    {
                        <tr>
                            <td>
                                @Html.Raw(@listnama[a].Nama)
                            </td>
                            <td>
                                <input id="[@a].Id_Pegawai" 
name="[@a].Id_Pegawai" type="hidden" value="@listnama[a].Id_Pegawai" />
                                <input id="[@a].isMasuk" 
name="[@a].isMasuk" type="checkbox" value="true" /><input name="isMasuk" 
type="hidden" value="false" />
                            </td>
                        </tr>
                    }

                </table>
                <input type="submit" value="Create" class="btn btn-default" 
/>
            }
            <br />
```

19. View Kredit Index

```
@model IEnumerable<SG_SIM.Models.Kredit>

@{
    ViewBag.Title = "DAFTAR KREDIT";
}

<h3>Daftar Kredit</h3>
<br/>
@using (Ajax.BeginForm("SearchAll", "Kredit",
    new AjaxOptions
    {
        HttpMethod = "POST",
        InsertionMode = InsertionMode.Replace,
        LoadingElementId = "ajax-loader",
        UpdateTargetId = "listsearch",
    }))
{
    <table class="table">
        <tr>
            <th>
                <label>Filter Nama/Barang&nbsp;</label>
                <input type="text" id="searchy" name="searchy" />
                <select id="searchy2" name="searchy2">
                    <option value="0">Belum Lunas</option>
                    <option value="1">Lunas</option>
                </select>
                <input type="submit" value="Search" />
            </th>
        </tr>
    </table>
    <img id="ajax-loader" src="@Url.Content("~/Images/ajax-loader.gif")"
style="display:none" />
}
<hr />
<p>
    @Html.ActionLink("Tambah Baru", "Create")
</p>
<hr/>

<table class="table" id="listsearch">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Tanggal)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Total_Nominal)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Keterangan)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Pemohon.Nama)
        </th>
        <th></th>
        <th></th>
    </tr>
```

```
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Tanggal)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Total_Nominal)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Keterangan)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Pemohon.Nama)
        </td>
        <td>
            @Html.ActionLink("Edit", "Edit", new { id=item.Id_Kredit })

        </td>
        <td>
            @Html.ActionLink("Angsuran", "DetailAngsurans", new { id =
item.Id_Kredit })
        </td>
    </tr>
}

</table>
```

20.   View Kredit Angsuran

```
@model IEnumerable<SG_SIM.Models.Angsuran>

@{
    ViewBag.Title = "DetailAngsurans";
}

<h3>Detail Angsuran</h3>

<div>
    <hr />
    <dl class="dl-horizontal">
        <dt>
            Total
        </dt>
        <dd>
            @ViewBag.datakredit.Total_Nominal
        </dd>

        <dt>
            Keterangan
        </dt>

        <dd>
            @ViewBag.datakredit.Keterangan
        </dd>
        <dt>
            Status
        </dt>
        <dd>
            @ViewBag.datakredit.StatusLunas
        </dd>
    </dl>
</div>


<table class="table">
    <tr>
        <th>
            @Html.DisplayNameFor(model => model.Tanggal)
        </th>
        <th>
            @Html.DisplayNameFor(model => model.Nominal)
        </th>
        <th></th>
    </tr>
    @{var total = Convert.ToInt32(ViewBag.datakredit.Total_Nominal);
    var jumlah = 0;}
@foreach (var item in Model) {
    <tr>
        <td>
            @Html.DisplayFor(modelItem => item.Tanggal)
        </td>
        <td>
            @Html.DisplayFor(modelItem => item.Nominal)
            @{jumlah = jumlah + Convert.ToInt32(item.Nominal);}
        </td>
        <td>
            @Html.ActionLink("Edit", "EditAngsuran", new { id = 
item.Id_Angsuran, totalnoms = 
Convert.ToDecimal(ViewBag.datakredit.Total_Nominal) }) |
            @Html.ActionLink("Delete", "Delete", new { id=item.Id_Angsuran 
})
        </td>
    </tr>
}
    <tr>
        <td><strong>Total</strong></td>
        <td>@jumlah</td>
        <td></td>
    </tr>
    @{total = total - jumlah;}
    <tr>
        <td><strong>Kekurangan</strong></td>
        <td>@total</td>
        <td></td>
    </tr>
</table>
<p>
    @Html.ActionLink("Tambah Angsuran", "Create","Angsuran",new { id = 
@ViewBag.datakredit.Id_Kredit , nominal=@total},null)
</p>
<hr />
     @Html.ActionLink("Kembali ke Daftar Kredit", "Index")
```