Class insurance

Selanjutnya kita akan membuat form insurance atau form asuransi. Form asuransi ini akan digunakan untuk mencatatkan semua nama-nama asuransi yang bekerja sama dengan perusahaan. Asuransi dapat dibedakan menjadi beberapa sesuai dengan kebutuhan. Untuk membuat form insurance kita perlu membuat class insurance sebagai teman pemprosessan data. Dan class form inventaris hanya sebagai tampilan dan penampilan informasi saja.

Coding class insurance

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using MySql.Data.MySqlClient;

using Core\_Inventaris;

using System.Windows.Forms;

namespace Core\_Inventaris

{

public class Insurance

{

// variable definition

Connection Connetion = new Connection();

MySqlDataReader Reader;

public List<InsuranceData> getData()

{

// function for get all insurance data from database server

Reader = Connetion.selectTable("select \* from insurance");

List<InsuranceData> InsDatas = new List<InsuranceData>();

while (Reader.Read())

{

InsuranceData InsData = new InsuranceData();

InsData.Id = Reader.GetString(0);

InsData.Name = Reader.GetString(1);

InsData.Address = Reader.GetString(2);

InsData.City = Reader.GetString(3);

InsData.Phone = Reader.GetString(4);

InsData.Contact = Reader.GetString(5);

InsDatas.Add(InsData);

}

Reader.Close();

return InsDatas;

}

public bool toServer()

{

// function for chekking connection to database server

List<string> Database = Connetion.readSever();

try

{

if (Connetion.connection(Database[0], Database[1], Database[2]))

{

return true;

}

else

{

return false;

}

}

catch(Exception e)

{

MessageBox.Show(e.Message);

return false;

}

}

public string deleteData(string ID)

{

// delete query

Reader = Connetion.selectTable("Delete from insurance where id='" + ID + "'");

Reader.Close();

return "Deleted!";

}

public string insertData(string ID,string InsuranceName,string Address, string City, string Phone, string Contact)

{

// insert data query

Reader = Connetion.selectTable("insert into insurance (id,insurancename,address,city,phone,contact) values ('" + ID + "','" +

InsuranceName + "','" + Address + "','" + City + "','" + Phone +"','" + Contact +"')");

Reader.Close();

return "Data Saved!";

}

public string updatetData(string ID, string InsuranceName, string Address, string City, string Phone, string Contact)

{

// update data query

Reader = Connetion.selectTable("update insurance set insurancename = '"+ InsuranceName +"', address='"+ Address + "', city ='"+

City + "',phone = '"+ Phone +"', contact = '"+ Contact +"' where id = '" + ID + "'");

Reader.Close();

return "Data Update!";

}

}

}