# BAB IV

# IMPLEMENTASI PROGRAM

Setelah desain sistem telah ditentukan maka selanjutnya adalah mewujudkannya dalam implementasi program. Pada bab ini dijelaskan mengenai implementasi pada fitur-fitur yang dimiliki oleh aplikasi laundry ini. Desain yang dibuat pada bab sebelumnya dijelaskan pada bab ini.

## Login

Pada proses login ini, user diminta untuk memasukan username dan password pada kolom yang tersedia. Yang dapat melakukan login ini hanya user atau pegawai yang terdaftar dalam database pegawai. Berikut adalah potongan program pada proses login.

Segmen program 4.1 Login User

1. Try
2. If tb\_username.Text = "" Then
3. If tb\_password.Text = "" Then
4. MsgBox("Harap Isi Username dan Password!")
5. Else
6. MsgBox("Harap Isi Username!")
7. End If
8. ElseIf tb\_password.Text = "" Then
9. MsgBox("Harap Isi Password!")
10. ElseIf lc.koneksi(tb\_username.Text, tb\_password.Text) Then
11. If lc.cekhakakses(tb\_username.Text) Then
12. MsgBox("Selamat Datang, " + tb\_username.Text,

MsgBoxStyle.Information)

1. FormUtama.Show()
2. Me.Hide()
3. End If
4. End If
5. Catch ex As Exception

Pada segmen di atas adalah kode program yang akan dijalankan ketika tombol login pada form login ditekan. Pada baris 2 hingga baris 9 merupakan pengecekan untuk field username dan

password kosong atau tidak. Jika username atau password kosong makan akan keluar pesan jika username atau password kosong. Pada baris 10 merupakan pengecekan untuk username danpassword yang sedang dimasukan ada pada database atau tidak dan data tersebut dilempar pada class apliasi untuk dilakukan query pada database. Berikut adalah class untuk proses login.

Segmen program 4.2 Class Login User

1. Public Function koneksi(ByVal user As String, ByVal pass As String) As Boolean
2. Try
3. Dim sr As New StreamReader("DBName.txt")
4. conn.ConnectionString = "Data Source=" & sr.ReadLine() & ";user id="

& user & ";password=" & pass

1. sr.Close()
2. conn.Open()
3. MsgBox("Berhasil Koneksi!")
4. Return True
5. Catch ex As OracleException
6. If ex.Number = 1017 Then
7. MsgBox("User belum terdaftar/user salah")
8. Else
9. MsgBox("Gagal Koneksi krn " & ex.Message)
10. End If
11. Return False
12. End Try
13. End Function
14. Public Function cekhakakses(ByVal user As String) As Boolean
15. Try
16. Dim cmd As New OracleCommand
17. cmd.Connection = conn

cmd.CommandText = "select iduser, fitur from thakakses where

iduser='" & user & "'"

1. Dim reader As OracleDataReader = cmd.ExecuteReader()
2. While reader.Read() 'Jika data ditemukan
3. If reader("fitur") = "admin" Then
4. Jabatan = "admin"
5. ElseIf reader("fitur") = "user" Then
6. Jabatan = "user"
7. End If
8. Username = reader("iduser")
9. End While
10. Return True
11. Catch ex As Exception
12. MsgBox(ex.Message)
13. Return False
14. End Try
15. End Function

Segmen di atas melakukan pengecekan data dari form login dengan database. Baris 1 hingga baris 18 merupakan pengecekan terhadap pegawai yang memasukan username dan password benar atau tidak pada database. Pada baris 4 terdapat query untuk mengecek data pada database. Kemudian baris 19 hingga baris 40 merupakan pengecekan terhadap hakakses user yang didapat pada tabel “thakakses”.

## Insert Data Master

Pada proses tambah master ini, pegawai akan memasukan beberapa data yang dibutuhkan untuk informasi master pada form. Data tersebut bisa beragam sesuai dengan form master tersebut. Fungsi insert atau tambah ini terdapat pada setiap form master. Source code pada setiap form pada umumnya memiliki karakteristik yang sama. Berikut contoh implementasi source code untuk menambah data pada form master member.

Segmen program 4.3 Tambah Data Master Member

1. Dim tgllhr As String
2. Dim jeniskelamin As String
3. If RadioButton1.Checked Then
4. jeniskelamin = "Pria"
5. Else
6. jeniskelamin = "Wanita"
7. End If
8. tgllhr = DateTimePicker1.Value.ToString("dd-MM-yyyy")
9. Dim msk As String = MaskedTextBox1.Text
10. msk = msk.Replace("(", "").Replace(")", "").Replace(" ", "")
11. If textBoxTerisiSemua() Then
12. Try
13. Dim iya As Integer = MessageBox.Show("Anda Yakin Insert Member " &

tb\_nama.Text & "?", "Konfirmasi", MessageBoxButtons.YesNo)

1. If iya = DialogResult.Yes Then
2. FormLogin.lc.insmember(idmember, tb\_nama.Text, tb\_alamat.Text, msk,

tgllhr, jeniskelamin)

1. End If
2. Catch ex As OracleException
3. If ex.Number = 1 Then
4. MsgBox("Data Yang Anda Masukkan Sudah Ada")
5. ElseIf ex.Number = 947 Then
6. MsgBox("Data Yang Dimasukkan Terlalu Panjang")
7. Else
8. MsgBox(ex.Message)
9. End If
10. End Try
11. FormLogin.lc.conn.Close()
12. Else
13. MsgBox("Harap Isi Data Yang Kosong")
14. End If
15. FormLogin.lc.loadMember(DataGridView1)
16. loadulang()

Pada baris 1 hingga baris 10 merupakan deklarasi dari variable yang membantu untuk memberikan value yang sama dengan format database. Sedangkan baris 11 merupakan pengecekan pada textbox form sudah terisi semua atau belum. Kemudian baris 12 hingga baris 25 merupakan proses memasukan data pada database. Proses ini mengirimkan data pada class untuk dimasukan ke database. Berikut penjelasan class tambah member.

Segment program 4.3 Class Tambah Data Master Member

1. Try
2. If conn.State = ConnectionState.Closed Then
3. conn.Open()
4. End If
5. Dim cmd As New OracleCommand
6. cmd.Connection = conn
7. cmd.CommandText = "insert into tmember(idmember, nama, alamat, telp,

tgllahir, jk) values(:idmember,:nama,:alamat,:telp,TO\_DATE(:tgllahir,

'dd-mm-yyyy'),:jk)"

1. cmd.Parameters.Add(New OracleParameter(":idmember",

OracleDbType.Varchar2, 6, idmember, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":nama", OracleDbType.Varchar2,

20, nama, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":alamat",

OracleDbType.Varchar2, 30, alamat, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":telp", OracleDbType.Varchar2,

7, telp, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":tgllahir", OracleDbType.Date,

tgllahir, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":jk", OracleDbType.Varchar2,

10, jk, ParameterDirection.Input))

1. cmd.ExecuteNonQuery()
2. Return True
3. Catch ex As Exception
4. Return False
5. End Try

Data yang dikirim pada class ini beragam sesuai dengan master yang bersangkutan. Sedangkan data id pada masing-masing master merupakan data yang diciptakan oleh trigger sebelum data dimasukan kedalam database. Pada baris 7 merupakan query yang digunakan untuk memasukan data pada database. Dan baris 8 hingga baris 13 digunakan untuk membuat parameter diskripsi dari data-data pada query baris 7. Sehingga data yang dimasukkan sesuai dengan kategori yang digunakan database. Baris 14 digunakan untuk eksekusi query yang sudah dibuat.

## Update Data Master

Pada proses update member, user dapat memilih data pada gridview masing-masing form master. Pada umumnya source code untuk mengubah data master memiliki karakterisitik yang sama. Data yang digunakan akan disesuaikan dengan formnya. Berikut contoh implementasi source code untuk mengubah data pada form master member.

Segmen program 4.4 Update Data Master Member

1. Dim tgllhr As String
2. Dim jeniskelamin As String
3. If RadioButton1.Checked Then
4. jeniskelamin = "Pria"
5. Else
6. jeniskelamin = "Wanita"
7. End If
8. tgllhr = DateTimePicker1.Value.ToString("dd-MM-yyyy")
9. Dim msk As String = MaskedTextBox1.Text
10. msk = msk.Replace("(", "").Replace(")", "").Replace(" ", "")
11. If textBoxTerisiSemua() Then
12. Try
13. Dim iya As Integer = MessageBox.Show("Anda Yakin Update Member " &

tb\_nama.Text & "?", "Konfirmasi", MessageBoxButtons.YesNo)

1. If iya = DialogResult.Yes Then
2. FormLogin.lc.updmember(tb\_idmember.Text, tb\_nama.Text,
3. tb\_alamat.Text, msk, tgllhr, jeniskelamin)
4. End If
5. Catch ex As OracleException
6. If ex.Number = 1 Then
7. MsgBox("Data Yang Anda Masukkan Sudah Ada")
8. ElseIf ex.Number = 947 Then
9. MsgBox("Data Yang Dimasukkan Terlalu Panjang")
10. Else
11. MsgBox(ex.Message)
12. End If
13. End Try
14. FormLogin.lc.conn.Close()
15. Else
16. MsgBox("Harap Isi Data Yang Kosong")
17. End If
18. FormLogin.lc.loadMember(DataGridView1)
19. loadulang()

Pada baris 1 hingga baris 8 merupakan deklarasi variable dari value yang didapat dari form master. Kemudian pada baris 11 merupakan pengecekan pada data yang dimasukan telah terisi. Data dari form tersebut kemudian dilemparkan ke class dari program untuk diolah dan dilanjutkan kedalam database master. Berikut potongan program dari class yang digunakan untuk memasukan data pada database master.

Segmen program 4.4 Class Update Data Master Member

1. Try
2. If conn.State = ConnectionState.Closed Then
3. conn.Open()
4. End If
5. Dim cmd As New OracleCommand
6. cmd.Connection = conn
7. cmd.CommandText = "update tmember set

idmember=:idmember,nama=:nama,alamat=:alamat,telp=:telp,

tgllahir=TO\_DATE(:tgllahir, 'dd-mm-yyyy'),jk=:jk where

idmember=:idmember"

1. cmd.Parameters.Add(New OracleParameter(":idmember",

OracleDbType.Varchar2, 6, idmember, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":nama", OracleDbType.Varchar2,

20, nama, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":alamat",

OracleDbType.Varchar2, 30, alamat, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":telp", OracleDbType.Varchar2,

7, telp, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":tgllahir", OracleDbType.Date,

tgllahir, ParameterDirection.Input))

1. cmd.Parameters.Add(New OracleParameter(":jk", OracleDbType.Varchar2,

10, jk, ParameterDirection.Input))

1. cmd.ExecuteNonQuery()
2. Return True
3. Catch ex As Exception
4. Return False
5. End Try

Pada baris 7 pada potongan program merupakan query yang akan dikirimkan kedalam database. Pada baris 8 hingga baris 13 adalah deskripsi value data dari program untuk query database. Dan dieksekusi pada baris 14.

## Delete Data Master

Pada setiap master pada program memiliki fitur delete untuk menghapus data pada database. Data yang akan dihapus dapat beragam sesuai dengan masternya. Pada umumnya kode program dari menghapus ini memiliki karakteristik yang sama pada setiap master. Berikut adalah contoh potongan program menghapus data dari form master member.

Segmen program 4.6 Delete Data Master Member

1. If tb\_idmember.Text <> "" Then
2. Try
3. Dim iya As Integer = MessageBox.Show("Anda Yakin Delete Member " &

tb\_nama.Text & "?", "Konfirmasi", MessageBoxButtons.YesNo)

1. If iya = DialogResult.Yes Then
2. FormLogin.lc.delmember(tb\_idmember.Text)

Segmen program 4.6 Delete Data Master Member (Lanjutan)

1. End If
2. Catch ex As OracleException
3. If ex.Number = 1 Then
4. MsgBox("Data Yang Anda Masukkan Sudah Ada")
5. ElseIf ex.Number = 947 Then
6. MsgBox("Data Yang Dimasukkan Terlalu Panjang")
7. Else
8. MsgBox(ex.Message)
9. End If
10. End Try
11. FormLogin.lc.conn.Close()
12. Else
13. MsgBox("Harap Isi Data Yang Kosong")
14. End If
15. FormLogin.lc.loadMember(DataGridView1)
16. loadulang()

Pada potongan program ini, baris pertama memberikan pengecekan terhadap data yang akan dieksekusi sudah terisi. Kemudian pada baris 5, data dilempar ke class program. Berikut potongan program pada class.

Segmen program 4.7 Class Delete Data Master Member

1. Try
2. If conn.State = ConnectionState.Closed Then
3. conn.Open()
4. End If
5. Dim cmd As New OracleCommand
6. cmd.Connection = conn
7. cmd.CommandText = "delete from tmember where idmember=:idmember"
8. cmd.Parameters.Add(New OracleParameter(":idmember",

OracleDbType.Varchar2, 6, idmember, ParameterDirection.Input))

1. cmd.ExecuteNonQuery()
2. Return True
3. Catch ex As Exception
4. Return False
5. End Try

Pada potongan program class ini, data yang dilepar dari form master akan masukan query pada baris baris 7 dan baris 8. Dan query tersebut dieksekusi pada baris 9.

## Load Data Tabel

Setiap form yang memiliki data tabel membutuhkan source code untuk menampilan isi data. Pada umumnya source code untuk menampilkan data tabel memiliki karakterisitik yang sama. Data yang ada pada tabel akan disesuaikan dengan form nya. Berikut contoh implementasi source code untuk menampilkan data tabel pada form master member.

Segmen program 4.8 Load Data Tabel Master Member

1. Try
2. If conn.State = ConnectionState.Closed Then
3. conn.Open()
4. End If
5. Dim adp As New OracleDataAdapter("SELECT idmember AS IDMember, nama AS

NamaMember FROM tmember WHERE idmember <> 'Non Member'" & input, conn)

1. Dim ds As New DataSet
2. adp.Fill(ds)
3. smrg.DataSource = ds.Tables(0)
4. Return True
5. Catch ex As Exception
6. MsgBox(ex.Message)
7. Return False
8. End Try

## Save Transaksi

Setiap form transaksi dapat melakukan penyimpanan data. Pada umumnya source code untuk menambah data transaksi memiliki karakterisitik yang sama. Query yang digunakan akan disesuaikan dengan form nya. Berikut contoh implementasi source code untuk menambah data pada form transaksi laundry baru.

Segmen program 4.9 Save Transaksi Laundry Baru

1. Try
2. If bayar\_txt.Text <> "" Then
3. bayar\_txt.Text = CDec(bayar\_txt.Text)
4. bayar\_txt.SelectAll()
5. bayar\_txt.Text = Format(Val(bayar\_txt.Text), "#,###")
6. TextBox3\_Validated(sender, e)
7. End If
8. cektotal()
9. Dim k As New KonfirmasiNewLaundry
10. k.lbKode.Text = FormLogin.lc.autogenkodetrans()
11. k.lbCustomer.Text = namaCust
12. k.lbKategori.Text = kategori
13. k.lbTipe.Text = tipe
14. k.lbNamaBrg.Text = nama\_barang
15. k.lbBerat.Text = beratCucian & " kg"
16. k.lbPewangi.Text = pewangi
17. k.lbLain2.Text = lain2
18. k.lbTglMasuk.Text = DateTimePicker1.Text
19. k.lbTglKeluar.Text = DateTimePicker2.Text
20. k.lbTotal.Text = "Rp. " & Format(Val(grandTotal), "#,###")
21. If bayar <> 0 Then
22. k.lbBayar.Text = "Rp. " & Format(Val(bayar), "#,###")
23. Else
24. k.lbBayar.Text = "Rp. 0"
25. End If
26. k.lbStatus.Text = status
27. If status = "Belum Lunas" Then
28. k.lbKurang.Visible = True
29. k.lbKurang0.Visible = True
30. k.lbKurang1.Visible = True
31. k.lbKurang.Text = "Rp. " & kembali\_txt.Text
32. End If
33. k.ShowDialog(Me)
34. If k.DialogResult = DialogResult.OK Then
35. FormLogin.lc.saveTrans()
36. End If
37. simpan\_btn.Enabled = Not simpan\_btn.Enabled
38. print\_btn.Enabled = Not print\_btn.Enabled
39. print\_btn.Focus()
40. Catch ex As Exception
41. MsgBox(ex.Message)
42. End Try
43. cleardgv()
44. FormLogin.lc.tampildetail(DataGridView1)

Pada potongan program transaksi ini terdapat beberapa poin yang penting. Pada baris 2 hingga baris 7 adalah pengecekan terhadap inputan bayar, sehingga ketika tidak diisi program tidak akan mengalami eror. Pada baris baris 9 hingga baris 32 merupakan deklarasi value untuk ditampilkan pada form KonfirmasiNewLaundry dan baris 33 digunakan untuk menjalankan form tersebut. Pada baris 35 value data tersebut dilemparkan ke dalam class program. Berikut potongan program transaksi pada class.

Segmen program 4.10 Class Save Transaksi Laundry Baru

1. Dim trans As OracleTransaction = conn.BeginTransaction
2. Try
3. If conn.State = ConnectionState.Closed Then
4. conn.Open()
5. End If
6. Dim builder As New OracleCommandBuilder(da)
7. da.InsertCommand = builder.GetInsertCommand
8. da.UpdateCommand = builder.GetUpdateCommand
9. da.DeleteCommand = builder.GetDeleteCommand
10. da.Update(ds.Tables("detailTransaksi"))
11. trans.Commit()
12. MsgBox("data berhasil di insert ke database, harap print struk")
13. Return True
14. Catch ex As Exception
15. MsgBox(ex.Message)
16. trans.Rollback()
17. Return False
18. End Try

## Pencarian Data

Pada beberapa form terdapat fitur pencarian. Pada umumnya source code untuk menambah data transaksi memiliki karakterisitik yang sama. Query yang digunakan akan disesuaikan dengan form nya. Berikut contoh potongan program pencarian pada form ViewTrans.

Segmen program 4.11 Pencarian Data Transaksi

1. Try
2. If conn.State = ConnectionState.Closed Then
3. conn.Open()
4. End If
5. Dim query As String = "SELECT kodetrans, idmember FROM ttransaksi WHERE

status\_cucian = 'On Process'" & input

1. Dim adp As New OracleDataAdapter("", conn)
2. adp.SelectCommand.CommandText = query
3. Dim ds As New DataSet
4. adp.Fill(ds)
5. dgview.DataSource = ds.Tables(0)
6. Return True
7. Catch ex As OracleException
8. MsgBox(ex.Message)
9. Return False
10. End Try

Pada baris 5 merupakan query yang digunakan menentukan data yang dicari. Setelah data ditemukan, maka data akan ditampilkan pada tabel program.

BAB V

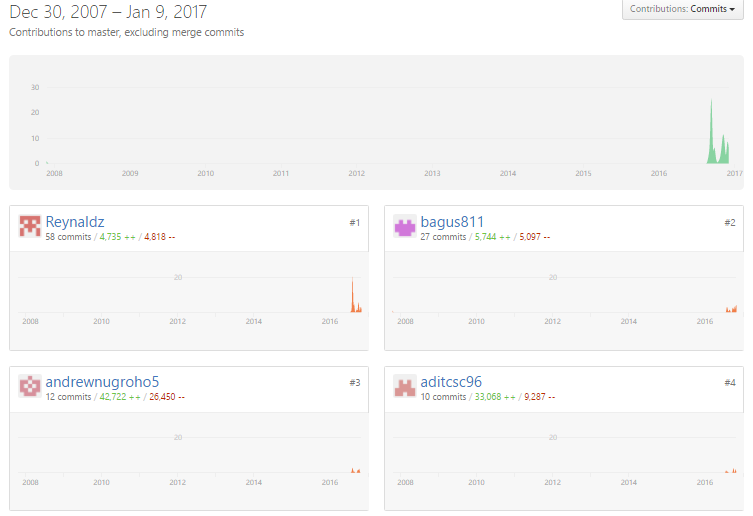
EVALUASI

Dalam bab ini akan dijelaskan mengenai evaluasi terhadap gantt chatt dan program laundry yang dibuat termasuk kendala – kendala yang dihadapi.

5.1 Evaluasi

Pada proses pembuatan aplikasi sistem inventori gudang toko elektronik terdapat penjadwalan yang telah dibuat dengan menggunakan gantt chart yang dapat dilihat pada Lampiran A. Saat proses pembuatannya membutuhkan waktu 86 hari mulai dari tanggal 12 September 2016 sampai dengan 09 Januari 2017.

Kami menggunakan Github sebagai media untuk versioning sehingga dapat terlihat proses pembuatannya. Ada beberapa modul yang terlambat namun mampu terselesaikan dengan baik dan sesuai jadwal yang telah ditetapkan bersama. Adapun dalam pembuatannya dibagi menjadi beberapa modul yang terdiri dari modul login, modul master, modul transaksi dan modul laporan. Dalam gantt chart terdapat pula jadwal pengerjaam buku laporan.

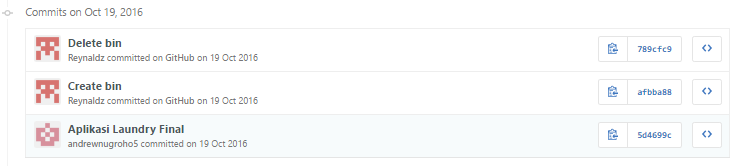


Gambar 5.1

Grafik Github

Pada Gambar 5.1, dapat dilihat bahwa mulai melakukan upload dan commit adalah 30 Oktober 2007, dikarenakan program ini merupakan program jadi tetapi diperbaharui dengan berbagai modul seharusnya mulai pembuatan adalah 29 September 2016. Sempat tertinggal diawal namun dapat kembali sesuai jadwal. Terlihat juga trafik dari masing-masing contributor pada pembuatan sistem laundry ini. Berikut ini adalah detail dari setiap modulnya:

1. Modul Login

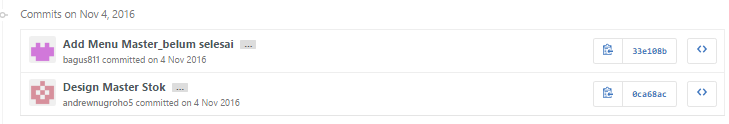


Gambar 5.2

Modul Login

Pada Gambar 5.1, dapat dilihat modul login telah diupload pada tanggal yang sama yakni 19 Oktober 2016. Apabila sesuai denga Gantt Chart maka seharusnya modul master telah dilakukan mulai tanggal 29 September 2016 sampai dengan 13 Oktober 2016. Terjadi keterlambatan dikarenakan, pengguna masih belum mengetahui cara penggunaan github dan masih mencoba-coba. Pada modul login mengalami keterlambatan dikarenakan masih melakukan tahap analisa sistem dan menyelesaikan desain database. Namun untuk modul login dilakukan pembuatan secara pararel sambil melanjutkan modul berikutnya.

1. Modul Master



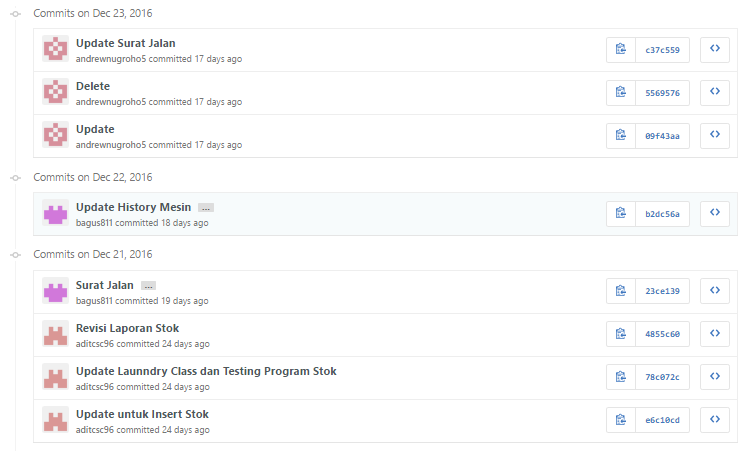
Gambar 5.3

Modul Master

Pada Gambar 5.3, dapat dilihat modul master diupload pada tanggal yang sama yakni 4 November 2016. Terjadi keterlambatan dikarenakan masih terdapat error dan tidak kompaknya kinerja tim. Apabila Progress sesuai dengan Gantt Chart seharusnya desain master stok dilakukan mulai tanggal 29 September 2016 sampai dengan 24 Oktober 2016. Namun dilakukan pembuatan secara pararel sambil melanjutkan modul berikutnya

1. Modul Transaksi

Pada modul transaksi, hanya pada form laporan stok yang sesuai dengan jadwal yang direncanakan sedangkan form lainnya tidak selesai tepat waktu.



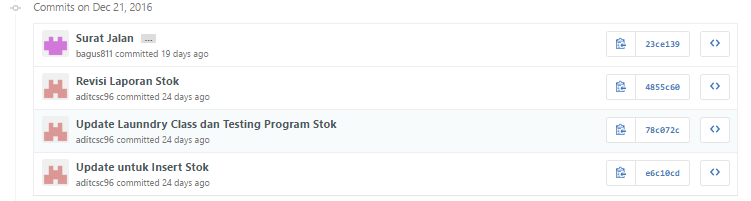
Gambar 5.4

Modul Transaksi

Pada Gambar 5.4, dapat dilihat pembuatan dimulai tanggal 21 Desember 2016 sampai 23 Desember 2016. Terjadi keterlambatan dikarenakan masih terdapat error dan telah dijelaskan bahwa hamper semua modul mengalami error hanya laporan stok saja yang sesuai dengan tepat waktu

1. Modul Laporan

Pada modul laporan mengalami keterlambatan dikarenakan harus memperbaiki error yang muncul dan menyelesaikan masalah pada cristal report. Secara garis besar, pembuatan program ini berjalan sesuai harapan dengan waktu yang cukup. Lebih detail dapat dilihat pada Gambar 5.5 dibawah ini



Gambar 5.5

Modul Laporan

Pada Gamabr 5.5, dapat dilihat seharunya dimulai pada tanggal 21 Desember 2016. Laporan dapat diselesaikan dengan tepat waktu. Dan melanjutkan perbaikan dibeberapa modul yang mengalami error.