

KATA PENGANTAR

Puji dan syukur kami hantarkan kepada Tuhan yang Maha Esa atas berkat dan rahmat-Nya sehingga kami dapat menyelesaikan laporan Final Project untuk mata kuliah Pemrograman Desktop dengan baik dan tepat waktu.

Laporan ini disusun sebagai dokumentasi dari perancangan aplikasi kasir untuk UMKM Cirengku Cimolku, sebuah UMKM di Surabaya yang membutuhkan sistem pencatatan transaksi yang lebih efisien. Melalui project ini, kami berupaya memberikan solusi teknologi yang dapat membantu meningkatkan efektivitas operasional UMKM tersebut.

Dalam proses penyusunan laporan ini, kami mendapat banyak bantuan dan dukungan dari berbagai pihak. Oleh karena itu, kami ingin menyampaikan terima kasih kepada:

1. Bapak Abdul Rezha Efrat Najaf, S.Kom, M.Kom selaku dosen pengampu mata kuliah Pemrograman Desktop yang telah memberikan arahan dan masukan sesudah presentasi sehingga kami bisa mengerjakan project ini dengan baik.
2. Pihak UMKM *Cirengku Cimolku* yang telah bersedia menjadi objek studi dalam pengembangan aplikasi ini.
3. Beberapa anggota yang telah membantu dalam penyelesaian project ini.

Kami menyadari bahwa laporan ini masih jauh dari sempurna. Oleh karena itu, kami mengharapkan kritik dan saran yang membangun untuk perbaikan di masa mendatang. Semoga laporan ini dapat bermanfaat bagi pembaca dan dapat memberikan kontribusi positif bagi perkembangan UMKM di Indonesia.

Surabaya, Desember 2024

Tim Penyusun Kelompok 8

DAFTAR ISI

KATA PENGANTAR.....	2
DAFTAR ISI.....	3
DAFTAR GAMBAR.....	5
DAFTAR TABEL.....	6
BAB I.....	7
PENDAHULUAN.....	7
1.1 Latar Belakang.....	7
1.2 Rumusan Masalah.....	7
1.3 Tujuan.....	8
1.4 Manfaat.....	8
BAB II.....	9
KEBUTUHAN SISTEM.....	9
2.1 Kebutuhan Fungsional.....	9
2.1.1 Use Case Diagram.....	9
Gambar 1. Use Case Diagram.....	10
Tabel 1. Deskripsi Use Case.....	10
2.2 Kebutuhan Non-Fungsional Availability.....	11
BAB III.....	12
PERANCANGAN SISTEM.....	12
3.1 Perancangan Arsitektur.....	12
3.2 Perancangan Antarmuka.....	12
3.2.1 Sequence Diagram.....	12
3.2.2 Class Diagram.....	30
Gambar 18. Class Diagram.....	30
3.3 Perancangan Data.....	32
3.3.1 CDM (Conceptual Data Model).....	32
Gambar 19. CDM.....	32
3.3.2 PDM (Physical Data Model).....	34
Gambar 20. PDM.....	34
3.4 Tampilan Aplikasi.....	36
Gambar 21. Tampilan Login Aplikasi.....	36
Gambar 22. Tampilan Sign Up Aplikasi.....	36
BAB III.....	39
KESIMPULAN DAN SARAN.....	39
3.1 Kesimpulan.....	39
3.2 Saran.....	39

LAMPIRAN.....	40
----------------------	-----------

DAFTAR GAMBAR

Gambar 1. Use Case Diagram.....	9
Gambar 2. Sequence Diagram Mengelola Menu.....	11
Gambar 3. Sequence Diagram Atur Harga.....	12
Gambar 4. Sequence Diagram Kelola Variant.....	13
Gambar 5. Sequence Diagram Login.....	14
Gambar 6. Logout.....	16
Gambar 7. Mengelola Stok.....	17
Gambar 8. Sequence Diagram Cek Stok.....	18
Gambar 9. Sequence Diagram Notifikasi Stok Sedikit.....	19
Gambar 10. Sequence Diagram Mengambil Data Akun.....	20
Gambar 11. Sequence Diagram Mencetak Laporan.....	21
Gambar 12. Sequence Diagram Mengelola Laporan.....	22
Gambar 13. Sequence Diagram Pilih Menu.....	23
Gambar 14. Sequence Diagram Buat Pesanan.....	24
Gambar 15. Hitung Total.....	25
Gambar 16. Proses Pembayaran.....	26
Gambar 17. Sequence Diagram Cetak Struk.....	28
Gambar 18. Class Diagram.....	29
Gambar 19. CDM.....	31
Gambar 20. PDM.....	33
Gambar 21. Tampilan Login Aplikasi.....	35
Gambar 22. Tampilan Sign Up Aplikasi.....	35
Gambar 23. Tampilan Awal Aplikasi.....	37
Gambar 24. Tampilan Transaksi.....	37
Gambar 25. Tampilan Cetak Laporan.....	38

DAFTAR TABEL

Tabel 1. Deskripsi Use Case.....	9
----------------------------------	---

BAB I

PENDAHULUAN

1.1 Latar Belakang

Usaha Mikro, Kecil, dan Menengah (UMKM) memiliki peran yang sangat signifikan dalam perekonomian Indonesia. UMKM tidak hanya berkontribusi terhadap penciptaan lapangan kerja, tetapi juga mampu menyerap tenaga kerja dalam jumlah besar dan berkontribusi pada Produk Domestik Bruto (PDB) negara. Menurut data dari Kementerian Koperasi dan UKM, pada tahun 2013, UMKM menyerap lebih dari 6 juta tenaga kerja di seluruh Indonesia. Dalam konteks ini, Cireng sebagai salah satu produk UMKM menunjukkan potensi yang besar untuk berkembang, terutama dengan meningkatnya permintaan akan makanan ringan yang unik dan berkualitas.

UMKM *Cirengku Cimolku* merupakan salah satu usaha kuliner di Surabaya yang bergerak di bidang jajanan tradisional khas Indonesia, seperti Cireng Mercon, Cireng Salju, dan Cimol. Dengan harga yang terjangkau, yakni sekitar 10 ribu rupiah per porsi, *Cirengku Cimolku* telah berhasil menarik perhatian konsumen dari berbagai kalangan. Sejak didirikan dua tahun yang lalu, UMKM ini telah berkembang pesat dengan 40 cabang tersebar di wilayah Surabaya. Setiap harinya, produksi mencapai 200 cireng menggunakan alat produksi sederhana.

Meskipun memiliki potensi pasar yang besar dan pengelolaan yang cukup berkembang, proses pencatatan transaksi penjualan di UMKM ini masih dilakukan secara manual. Sistem manual tersebut memerlukan pencatatan dalam buku atau menggunakan metode sederhana lainnya yang memakan waktu dan berisiko terhadap kesalahan pencatatan atau kehilangan data transaksi. Hal ini dapat menghambat efisiensi operasional serta proses analisis penjualan yang penting untuk pengambilan keputusan strategis, seperti pengelolaan stok bahan baku, perencanaan produksi, dan pengembangan cabang baru.

Dengan kondisi tersebut, diperlukan sebuah solusi berupa aplikasi kasir berbasis teknologi untuk mendigitalisasi proses transaksi. Aplikasi kasir ini diharapkan dapat membantu UMKM *Cirengku Cimolku* dalam mempermudah proses pencatatan penjualan, mengurangi risiko kesalahan atau kehilangan data transaksi, dan menyediakan data penjualan secara real-time yang dapat digunakan untuk analisis. Implementasi aplikasi kasir ini diharapkan dapat mendukung perkembangan UMKM *Cirengku Cimolku* menuju pengelolaan usaha yang lebih modern, efisien, dan terintegrasi, sehingga mampu bersaing di era digitalisasi saat ini.

1.2 Rumusan Masalah

1. Bagaimana membuat suatu sistem pencatatan transaksi pada UMKM *Cirengku Cimolku* yang berbasis desktop?

1.3 Tujuan

1. Mengembangkan sistem pencatatan transaksi berbasis desktop yang sederhana, efisien, dan sesuai kebutuhan operasional UMKM Cirengku Cimolku.
2. Memastikan sistem yang dikembangkan mampu memberikan kemudahan dalam pengelolaan data transaksi secara otomatis dan akurat.

1.4 Manfaat

Pengembangan sistem ini diharapkan dapat memberikan kontribusi positif bagi UMKM Cirengku Cimolku dengan meningkatkan efisiensi dalam pencatatan transaksi, mengurangi risiko kesalahan pencatatan manual, dan mendukung pengambilan keputusan yang lebih baik melalui akses data transaksi yang terstruktur dan mudah diakses. Selain itu, sistem ini diharapkan dapat meningkatkan daya saing UMKM melalui modernisasi proses bisnis berbasis teknologi informasi.

BAB II

KEBUTUHAN SISTEM

2.1 Kebutuhan Fungsional

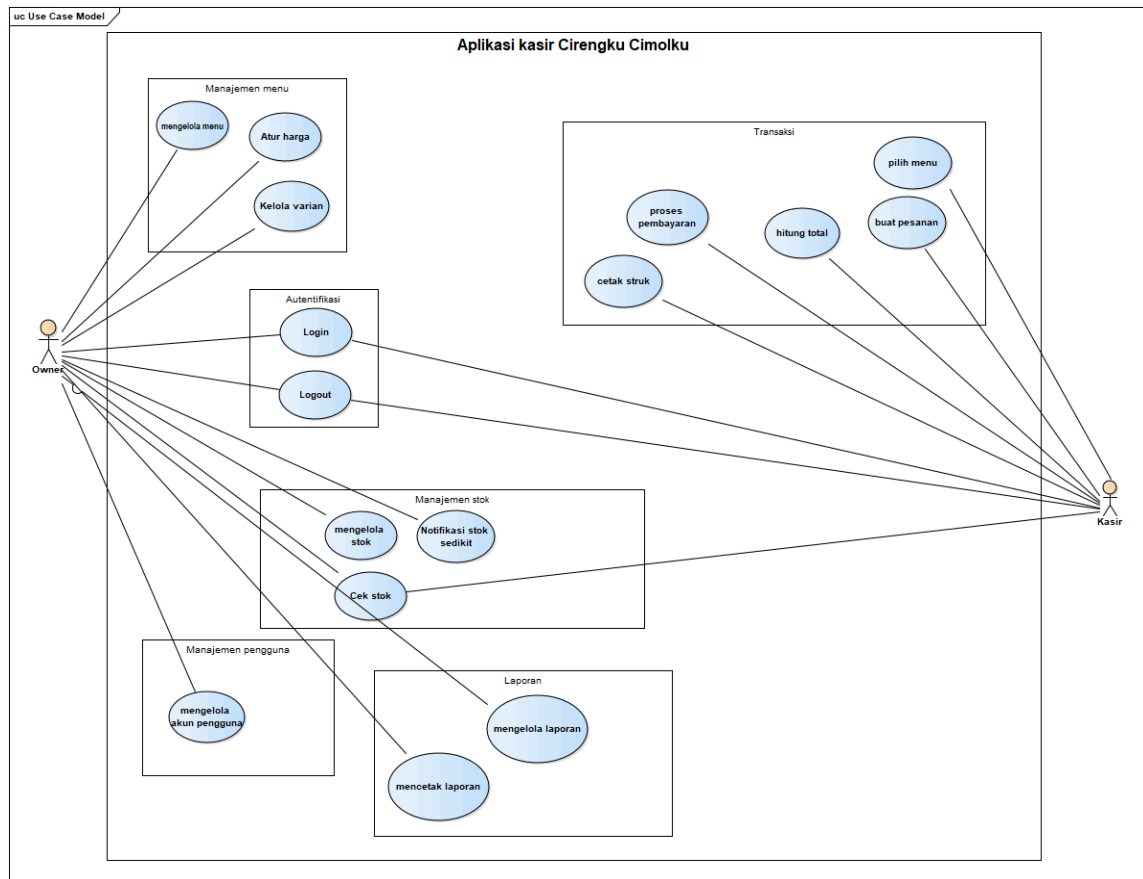
1. Manajemen Transaksi Penjualan
 - Sistem mampu mencatat data penjualan secara otomatis.
 - Sistem dapat mencetak struk transaksi setelah penjualan selesai.
 - Sistem dapat menghitung total harga dan diskon.
2. Manajemen Stok Bahan Baku
 - Sistem mencatat jumlah stok.
 - Sistem memberikan notifikasi jika stok mendekati batas minimum.
3. Manajemen Data Produk
 - Sistem memungkinkan admin untuk menambahkan, mengubah, atau menghapus data produk (misal: *Cireng Mercon*, *Cireng Salju*, *Cimol*).
 - Sistem dapat menampilkan daftar produk beserta harga per porsi.
4. Laporan Penjualan
 - Sistem dapat menghasilkan laporan penjualan.
5. Manajemen Pengguna
 - Sistem mendukung peran pengguna, seperti Owner dan kasir.
 - Admin memiliki hak akses untuk mengelola data produk, transaksi, dan laporan.
6. Pencarian dan Filter Data
 - Sistem menyediakan fitur pencarian transaksi.
 - Sistem dapat memfilter data penjualan sesuai kebutuhan.

2.1.1 Use Case Diagram

Use case diagram merupakan salah satu teknik dalam Unified Modeling Language (UML) yang digunakan untuk menggambarkan interaksi antara aktor (pengguna) dengan sistem. Diagram ini menjelaskan fungsi-fungsi atau fitur utama yang ada dalam aplikasi kasir untuk UMKM *Cirengku Cimolku*.

Pada aplikasi kasir ini, terdapat beberapa aktor utama, yaitu:

1. Kasir : Fokus pada transaksi penjualan.
2. Owner : Fokus pada manajemen.



Gambar 1. Use Case Diagram

Tabel 1. Deskripsi Use Case

Aktor	Fungsi
Owner	<ol style="list-style-type: none"> 1. Login 2. Logout 3. Manajemen menu : mengelola menu seperti menambah menu baru, mengubah detail menu, dan menghapus menu yang tidak tersedia. Atur harga, seperti menetapkan harga menu, mengubah harga menu, dan membuat diskon. Kelola Varian, seperti menambah varian menu, mengatur opsi topping, dan mengelola ukuran porsi. 4. Manajemen stok : mengelola stok, seperti input stok baru dan update jumlah stok. Notifikasi stok sedikit, seperti peringatan jika stok hampir habis.

2.2 Kebutuhan Non-Fungsional Availability

1. Waktu Operasional Sistem

- Sistem harus selalu tersedia selama jam operasional UMKM, yaitu setiap hari dari pukul 08:00 hingga 22:00.
- Sistem dapat diakses kapan saja jika dibutuhkan untuk mengecek laporan atau stok.

2. Fitur Pemulihan Cepat

- Jika terjadi gangguan atau sistem mengalami crash, sistem harus dapat dipulihkan dalam waktu kurang dari 10 menit.
- Data transaksi yang tercatat sebelumnya tidak boleh hilang atau rusak.

3. Kinerja Stabil

- Sistem harus tetap stabil dan responsif ketika digunakan untuk memproses transaksi dalam jumlah besar, misalnya minimal 200 transaksi per hari.

4. Notifikasi Kesalahan

- Jika terjadi masalah dalam sistem, seperti gagal menyimpan transaksi, sistem harus memberikan notifikasi kepada pengguna untuk segera melakukan tindakan atau melapor kepada teknisi.

BAB III

PERANCANGAN SISTEM

3.1 Perancangan Arsitektur

```
Imports System.Data.Odbc
Imports System.Data.SqlClient
Imports System.Transactions
Imports MySql.Data.MySqlClient
Imports Mysqlx
Public Class Form3
    Dim nomorAntrean As Integer
    Public userRole As String ' Variabel untuk menyimpan role dari FormLogin
    Private hargaMenu As Integer = 0

    Private Sub Form3_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        Try
            Module1.koneksi()
            UpdateStatusKoneksi()
            LoadDataTransaksi()
            LoadDataDetailTransaksi()
            LoadMenuToComboBox()

            ' Inisialisasi label stok
            InitializeStockLabels()

            ' Update stok
            UpdateStok()

            ' Konfigurasi UI berdasarkan peran
            If userRole = "Kasir" Then
                btnlaporan.Enabled = False
                pnllaporan.Visible = False
            ElseIf userRole = "Owner" Then
                btnlaporan.Enabled = True
            End If
        Catch ex As Exception
            MessageBox.Show("Terjadi kesalahan saat inisialisasi: " & ex.Message, "Error",
                MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Sub
End Class
```

End Sub

Private Sub InitializeStockLabels()

' Pastikan nama-nama label ini sesuai dengan yang ada di form designer

lblStokMercon = CType(Controls.Find("lblStokMercon", True)(0), Label)

lblstokcheese = CType(Controls.Find("lblStokCheese", True)(0), Label)

lblstoksalju = CType(Controls.Find("lblStokSalju", True)(0), Label)

lblstokcimol = CType(Controls.Find("lblStokCimol", True)(0), Label)

lblstokmix = CType(Controls.Find("lblStokMix", True)(0), Label)

End Sub

Private Sub UpdateStok()

Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")

Try

conn.Open()

Dim query As String = "SELECT id_menu, nama_menu, stok_menu FROM
tbl_menu"

Using cmd As New MySqlCommand(query, conn)

Using reader As MySqlDataReader = cmd.ExecuteReader()

While reader.Read()

Dim idMenu As Integer = reader.GetInt32("id_menu")

Dim stok As String = reader.GetString("stok_menu")

Dim namaMenu As String = reader.GetString("nama_menu")

' Update label stok sesuai dengan menu

Select Case namaMenu

Case "Cireng Mercon"

lblstokmercon.Text = "Stok: " & stok

Case "Cireng Cheese"

lblstokcheese.Text = "Stok: " & stok

Case "Cireng Salju"

lblstoksalju.Text = "Stok: " & stok

Case "Cimol Bubuk"

lblstokcimol.Text = "Stok: " & stok

Case "Paket Mix"

lblstokmix.Text = "Stok: " & stok

End Select

End While

End Using

```

        End Using
    Catch ex As Exception
        MessageBox.Show("Gagal memuat data stok: " & ex.Message, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Using
End Sub

Private Sub UpdateStokSetelahTransaksi(idMenu As Integer, jumlahDibeli As Integer)
    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            ' Update stok di database
            Dim query As String = "UPDATE tbl_menu SET stok_menu = stok_menu - @jumlah
WHERE id_menu = @id_menu"
            Using cmd As New MySqlCommand(query, conn)
                cmd.Parameters.AddWithValue("@jumlah", jumlahDibeli)
                cmd.Parameters.AddWithValue("@id_menu", idMenu)
                cmd.ExecuteNonQuery()
            End Using

            ' Refresh tampilan stok
            UpdateStok()
        Catch ex As Exception
            MessageBox.Show("Gagal update stok: " & ex.Message, "Error",
            MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Using
End Sub

Private Function CekStokMencukupi(idMenu As Integer, jumlahDiminta As Integer) As
Boolean
    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            Dim query As String = "SELECT stok_menu FROM tbl_menu WHERE id_menu =
@id_menu"
            Using cmd As New MySqlCommand(query, conn)

```

```

        cmd.Parameters.AddWithValue("@id_menu", idMenu)
        Dim stokTersedia As Integer = Convert.ToInt32(cmd.ExecuteScalar())
        Return stokTersedia >= jumlahDiminta
    End Using
    Catch ex As Exception
        MessageBox.Show("Gagal mengecek stok: " & ex.Message, "Error",
        MessageBoxButtons.OK, MessageBoxIcon.Error)
        Return False
    End Try
End Using
End Function

```

```

Private Sub UpdateStatusKoneksi()
    If Module1.IsConnected() Then
        StatusKoneksi.Text = "Connected"
        StatusKoneksi.ForeColor = Color.Green
    Else
        StatusKoneksi.Text = "Not Connected"
        StatusKoneksi.ForeColor = Color.Red
    End If
End Sub

```

```

Private Sub LoadDataTransaksi()
    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            Dim query As String = "SELECT * FROM tbl_transaksi"
            Dim adapter As New MySqlDataAdapter(query, conn)
            Dim table As New DataTable()
            adapter.Fill(table)
            dgTransaksi.DataSource = table

            Catch ex As Exception
                MessageBox.Show("Gagal memuat data: " & ex.Message)
            End Try
        End Using
    End Sub

```

```

Private Sub LoadDataDetailTransaksi()

```

```

Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
Try
    conn.Open()
    Dim query As String = "SELECT * FROM tbl_detail_transaksi"
    Dim adapter As New MySqlDataAdapter(query, conn)
    Dim table As New DataTable()
    adapter.Fill(table)
    dgDetail.DataSource = table

Catch ex As Exception
    MessageBox.Show("Gagal memuat data detail transaksi: " & ex.Message)
End Try
End Using
End Sub

```

```

Private Sub LoadMenuToComboBox()
    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
    Try
        conn.Open()
        Dim query As String = "SELECT nama_menu FROM tbl_menu"
        Dim cmd As New MySqlCommand(query, conn)
        Dim reader As MySqlDataReader = cmd.ExecuteReader()

        cmbMenu.Items.Clear()
        While reader.Read()
            cmbMenu.Items.Add(reader("nama_menu"))
        End While

    Catch ex As Exception
        MsgBox("Gagal memuat menu: " & ex.Message)
    End Try
End Using
End Sub

```

```

Private Sub cmbMenu_SelectedIndexChanged(sender As Object, e As EventArgs) Handles
cmbMenu.SelectedIndexChanged
    Try
        ' Pastikan koneksi aktif
    End Try

```

koneksi()

' Query untuk mengambil ID menu dan harga menu berdasarkan nama menu yang dipilih
Dim query As String = "SELECT id_menu, harga_menu FROM tbl_menu WHERE
nama_menu = @nama_menu"

Using cmd As New MySqlCommand(query, conn)
cmd.Parameters.AddWithValue("@nama_menu", cmbMenu.Text) ' Menggunakan
nama menu yang dipilih dari ComboBox

Using reader As MySqlDataReader = cmd.ExecuteReader()
If reader.Read() Then
txtidmenu.Text = reader("id_menu").ToString()
hargaMenu = Convert.ToInt32(reader("harga_menu"))
End If
End Using
End Using

Catch ex As Exception
MsgBox("Terjadi kesalahan saat mengambil data menu: " & ex.Message)
Finally
' Tutup koneksi
closeConnection()
End Try
End Sub

Private Sub txtJumlah_TextChanged(sender As Object, e As EventArgs) Handles
txtJumlah.TextChanged
Dim jumlah As Integer
If Integer.TryParse(txtJumlah.Text, jumlah) Then
Dim subtotal As Integer = jumlah * hargaMenu
txtSubTotal.Text = subtotal.ToString() ' Gunakan string sederhana tanpa format "N0"
Else
txtSubTotal.Text = "0"
End If
End Sub

Private Sub HitungTotal()
Dim total As Integer = 0
For Each row As DataGridViewRow In dgDetail.Rows


```

    Dim subtotal As Integer
    If Integer.TryParse(row.Cells("subtotal").Value.ToString(), subtotal) Then
        total += subtotal
    End If
Next
txtTotal.Text = total.ToString() ' Tampilkan total di TextBox
End Sub

Private Sub SimpanTransaksi()
    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            Dim queryTransaksi As String = "INSERT INTO tbl_transaksi (total_harga,
tanggal_transaksi, id_pelanggan) VALUES (@total_harga, @tanggal_transaksi,
@id_pelanggan)"

            Using cmdTransaksi As New MySqlCommand(queryTransaksi, conn)
                cmdTransaksi.Parameters.AddWithValue("@total_harga",
Convert.ToInt32(txtTotal.Text))
                cmdTransaksi.Parameters.AddWithValue("@tanggal_transaksi",
DateTimePicker1.Value)
                cmdTransaksi.Parameters.AddWithValue("@id_pelanggan", txtIdPelanggan.Text)
                cmdTransaksi.ExecuteNonQuery()

                ' Ambil no_antrian yang baru dibuat
                Dim lastInsertQuery As String = "SELECT LAST_INSERT_ID()"
                Using cmdLastInsert As New MySqlCommand(lastInsertQuery, conn)
                    nomorAntrean = Convert.ToInt32(cmdLastInsert.ExecuteScalar())
                    txtNoAntrian.Text = nomorAntrean.ToString()
                End Using
            End Using

            Catch ex As Exception
                MessageBox.Show("Gagal menyimpan transaksi: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
                Throw ' Lempar exception untuk ditangkap di level atas
            End Try
        End Using
    End Sub

```

```

' Sub untuk menyimpan ke tbl_detail_transaksi
Private Sub SimpanDetailTransaksi()
    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            Dim queryDetail As String = "INSERT INTO tbl_detail_transaksi (no_antrian,
id_menu, jumlah_transaksi, subtotal) VALUES (@no_antrian, @id_menu, @jumlah_transaksi,
@subtotal)"

            Using cmdDetail As New MySqlCommand(queryDetail, conn)
                cmdDetail.Parameters.AddWithValue("@no_antrian", nomorAntrean)
                cmdDetail.Parameters.AddWithValue("@id_menu", txtidmenu.Text)
                cmdDetail.Parameters.AddWithValue("@jumlah_transaksi",
Convert.ToInt32(txtJumlah.Text))
                cmdDetail.Parameters.AddWithValue("@subtotal",
Convert.ToInt32(txtSubTotal.Text))
                cmdDetail.ExecuteNonQuery()
            End Using

            Catch ex As Exception
                MessageBox.Show("Gagal menyimpan detail transaksi: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
                Throw ' Lempar exception untuk ditangkap di level atas
            End Try
        End Using
    End Sub

' Button Simpan yang baru
Private Sub btnsimpan_Click(sender As Object, e As EventArgs) Handles btnsimpan.Click
    ' Validasi input
    If String.IsNullOrEmpty(txtIdPelanggan.Text) OrElse
String.IsNullOrEmpty(txtidmenu.Text) OrElse
String.IsNullOrEmpty(txtJumlah.Text) OrElse
String.IsNullOrEmpty(txtSubTotal.Text) Then
        MessageBox.Show("Pastikan semua data telah diisi dengan benar.", "Peringatan",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Return
    End If

```

```

Try
    ' Jika txtNoAntrian kosong, berarti ini transaksi baru
    If String.IsNullOrEmpty(txtNoAntrian.Text) Then
        SimpanTransaksi()
    End If

    ' Simpan detail transaksi
    SimpanDetailTransaksi()

    ' Refresh DataGridView
    LoadDataTransaksi()
    LoadDataDetailTransaksi()

    ' Reset form detail (tapi tidak reset no_antrian dan total)
    ResetFormDetail()

    MessageBox.Show("Item berhasil ditambahkan ke transaksi.", "Sukses",
    MessageBoxButtons.OK, MessageBoxIcon.Information)

    Catch ex As Exception
        MessageBox.Show("Terjadi kesalahan: " & ex.Message, "Error",
    MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Sub

' Sub untuk reset form detail saja
Private Sub ResetFormDetail()
    cmbMenu.SelectedIndex = -1
    txtidmenu.Text = ""
    txtJumlah.Text = ""
    txtSubTotal.Text = ""
End Sub

Private Sub btnTambah_Click(sender As Object, e As EventArgs) Handles btnTambah.Click
    txtNoAntrian.Text = ""
    txtIdPelanggan.Text = ""
    txtTotal.Text = "0"
    ResetFormDetail()
    nomorAntrean = 0

```

End Sub

```
Private Sub btnedit_Click(sender As Object, e As EventArgs) Handles btnedit.Click
    If String.IsNullOrEmpty(txtNoAntrian.Text) OrElse
        String.IsNullOrEmpty(txtidmenu.Text) OrElse
        String.IsNullOrEmpty(txtJumlah.Text) OrElse
        String.IsNullOrEmpty(txtSubTotal.Text) Then
        MessageBox.Show("Pastikan semua data telah diisi dengan benar.", "Peringatan",
        MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Return
    End If

    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            ' Update detail transaksi
            Dim queryDetail As String = "UPDATE tbl_detail_transaksi SET " &
                "jumlah_transaksi = @jumlah_transaksi, " &
                "subtotal = @subtotal " &
                "WHERE no_antrian = @no_antrian AND id_menu = @id_menu"

            Using cmdDetail As New MySqlCommand(queryDetail, conn)
                cmdDetail.Parameters.AddWithValue("@no_antrian",
                Convert.ToInt32(txtNoAntrian.Text))
                cmdDetail.Parameters.AddWithValue("@id_menu",
                Convert.ToInt32(txtidmenu.Text))
                cmdDetail.Parameters.AddWithValue("@jumlah_transaksi",
                Convert.ToInt32(txtJumlah.Text))
                cmdDetail.Parameters.AddWithValue("@subtotal",
                Convert.ToInt32(txtSubTotal.Text))
                cmdDetail.ExecuteNonQuery()
            End Using

            ' Update total di tabel transaksi
            Dim queryUpdateTotal As String = "UPDATE tbl_transaksi SET " &
                "total_harga = @total_harga " &
                "WHERE no_antrian = @no_antrian"

            Using cmdUpdateTotal As New MySqlCommand(queryUpdateTotal, conn)
```

```

        cmdUpdateTotal.Parameters.AddWithValue("@total_harga",
Convert.ToInt32(txtTotal.Text))
        cmdUpdateTotal.Parameters.AddWithValue("@no_antrian",
Convert.ToInt32(txtNoAntrian.Text))
        cmdUpdateTotal.ExecuteNonQuery()
    End Using

    MessageBox.Show("Data berhasil diperbarui.", "Sukses", MessageBoxButtons.OK,
MessageBoxIcon.Information)

    ' Refresh data
    LoadDataTransaksi()
    LoadDataDetailTransaksi()

    Catch ex As Exception
        MessageBox.Show("Gagal memperbarui data: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
    End Try
End Using
End Sub

Private Sub btnhapus_Click(sender As Object, e As EventArgs) Handles btnhapus.Click
    If String.IsNullOrEmpty(txtNoAntrian.Text) Then
        MessageBox.Show("Pilih transaksi yang ingin dihapus terlebih dahulu.", "Peringatan",
MessageBoxButtons.OK, MessageBoxIcon.Warning)
        Return
    End If

    Dim result As DialogResult = MessageBox.Show("Apakah Anda yakin ingin menghapus
transaksi ini?", "Konfirmasi", MessageBoxButtons.YesNo, MessageBoxIcon.Question)
    If result = DialogResult.No Then Return

    Using conn As New MySqlConnection("server=127.0.0.1;user
id=root;password=;database=db_cireng2")
        Try
            conn.Open()
            ' Detail transaksi akan terhapus otomatis karena ada CASCADE
            Dim queryTransaksi As String = "DELETE FROM tbl_transaksi WHERE no_antrian
= @no_antrian"

```

```

        Using cmdTransaksi As New MySqlCommand(queryTransaksi, conn)
            cmdTransaksi.Parameters.AddWithValue("@no_antrian",
Convert.ToInt32(txtNoAntrian.Text))
            cmdTransaksi.ExecuteNonQuery()
        End Using

        MessageBox.Show("Transaksi berhasil dihapus.", "Sukses", MessageBoxButtons.OK,
MessageBoxIcon.Information)

        ' Reset form dan refresh data
        txtNoAntrian.Text = ""
        txtIdPelanggan.Text = ""
        txtTotal.Text = "0"
        ResetFormDetail()
        LoadDataTransaksi()
        LoadDataDetailTransaksi()

        Catch ex As Exception
            MessageBox.Show("Gagal menghapus transaksi: " & ex.Message, "Error",
MessageBoxButtons.OK, MessageBoxIcon.Error)
        End Try
    End Using
End Sub

```

```
Private Sub FormKasir_FormClosing(sender As Object, e As FormClosingEventArgs)
Handles MyBase.FormClosing
    If conn IsNot Nothing Then conn.Close()
End Sub
```

```
Private Sub btnmenu_Click(sender As Object, e As EventArgs) Handles btnmenu.Click
    pnlon.Height = btnmenu.Height
    pnlon.Top = btnmenu.Top
    pnlmenu.Visible = True
    pnltransaksi.Visible = False
    pnllaporan.Visible = False
End Sub
```

```
Private Sub btntransaksi_Click(sender As Object, e As EventArgs) Handles btntransaksi.Click
    pnlon.Height = btntransaksi.Height
    pnlon.Top = btntransaksi.Top
    pnlmenu.Visible = False
    pnltransaksi.Visible = True
    pnllaporan.Visible = False
End Sub
```

```
Private Sub btnlaporan_Click(sender As Object, e As EventArgs) Handles btnlaporan.Click
    pnlon.Height = btnlaporan.Height
    pnlon.Top = btnlaporan.Top
    pnlmenu.Visible = False
    pnltransaksi.Visible = False
    pnllaporan.Visible = True
```

```
    If userRole = "Owner" Then
        pnllaporan.Visible = True
    Else
        MessageBox.Show("Anda tidak memiliki akses ke laporan. Silakan login sebagai
Owner.", "Akses Ditolak", MessageBoxButtons.OK, MessageBoxIcon.Warning)
        pnllaporan.Visible = False ' Panel tetap tersembunyi
    End If
End Sub
```

```
Private Sub Guna2CircleButton1_Click(sender As Object, e As EventArgs) Handles
Guna2CircleButton1.Click
    Me.Hide()
```

```
Form2.Show()  
End Sub
```

```
Private Sub Guna2CircleButton2_Click(sender As Object, e As EventArgs) Handles  
Guna2CircleButton2.Click  
    Me.WindowState = WindowState.Minimized  
End Sub
```

```
Private Sub Guna2CircleButton3_Click(sender As Object, e As EventArgs) Handles  
Guna2CircleButton3.Click  
    If Me.WindowState = WindowState.Normal Then  
        Me.WindowState = WindowState.Maximized  
    ElseIf Me.WindowState = WindowState.Maximized Then  
        Me.WindowState = WindowState.Normal  
    End If  
End Sub
```

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click  
    Dim cetaklaporan As New form4()  
    cetaklaporan.Show()  
    Me.Hide()  
End Sub
```

```
Private Sub dgDetail_CellContentClick(sender As Object, e As DataGridViewCellEventArgs)  
Handles dgDetail.CellContentClick  
    If e.RowIndex >= 0 Then  
        Dim row As DataGridViewRow = dgDetail.Rows(e.RowIndex)  
  
        txtNoAntrian.Text = row.Cells("no_antrian").Value.ToString()  
        txtidmenu.Text = row.Cells("id_menu").Value.ToString()  
        txtJumlah.Text = row.Cells("jumlah_transaksi").Value.ToString()  
        txtSubTotal.Text = row.Cells("subtotal").Value.ToString()  
  
        ' Load menu name based on id_menu  
        Using conn As New MySqlConnection("server=127.0.0.1;user  
id=root;password=;database=db_cireng2")  
            Try  
                conn.Open()  
                Dim query As String = "SELECT nama_menu FROM tbl_menu WHERE id_menu  
= @id_menu"
```



```

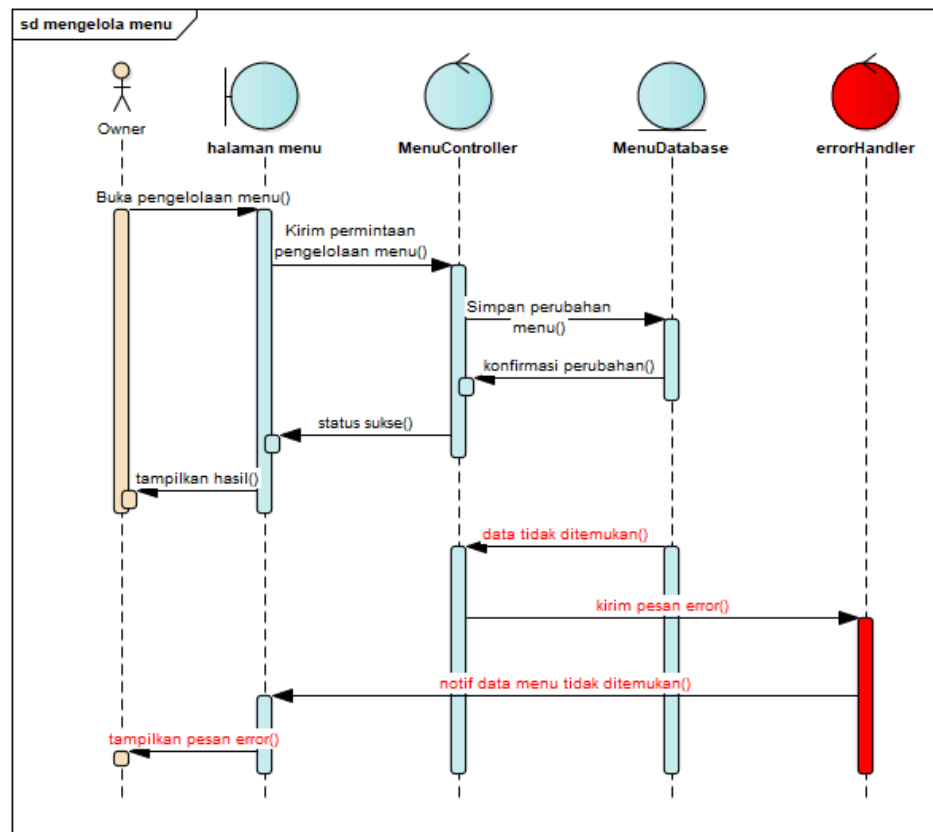
Using cmd As New MySqlCommand(query, conn)
    cmd.Parameters.AddWithValue("@id_menu", Convert.ToInt32(txtidmenu.Text))
    Dim menuName As Object = cmd.ExecuteScalar()
    If menuName IsNot Nothing Then
        cmbMenu.Text = menuName.ToString()
    End If
End Using
Catch ex As Exception
    MessageBox.Show("Error loading menu name: " & ex.Message)
End Try
End Using
End If
End Sub
End Class

```

3.2 Perancangan Antarmuka

3.2.1 Sequence Diagram

A. Mengelola Menu



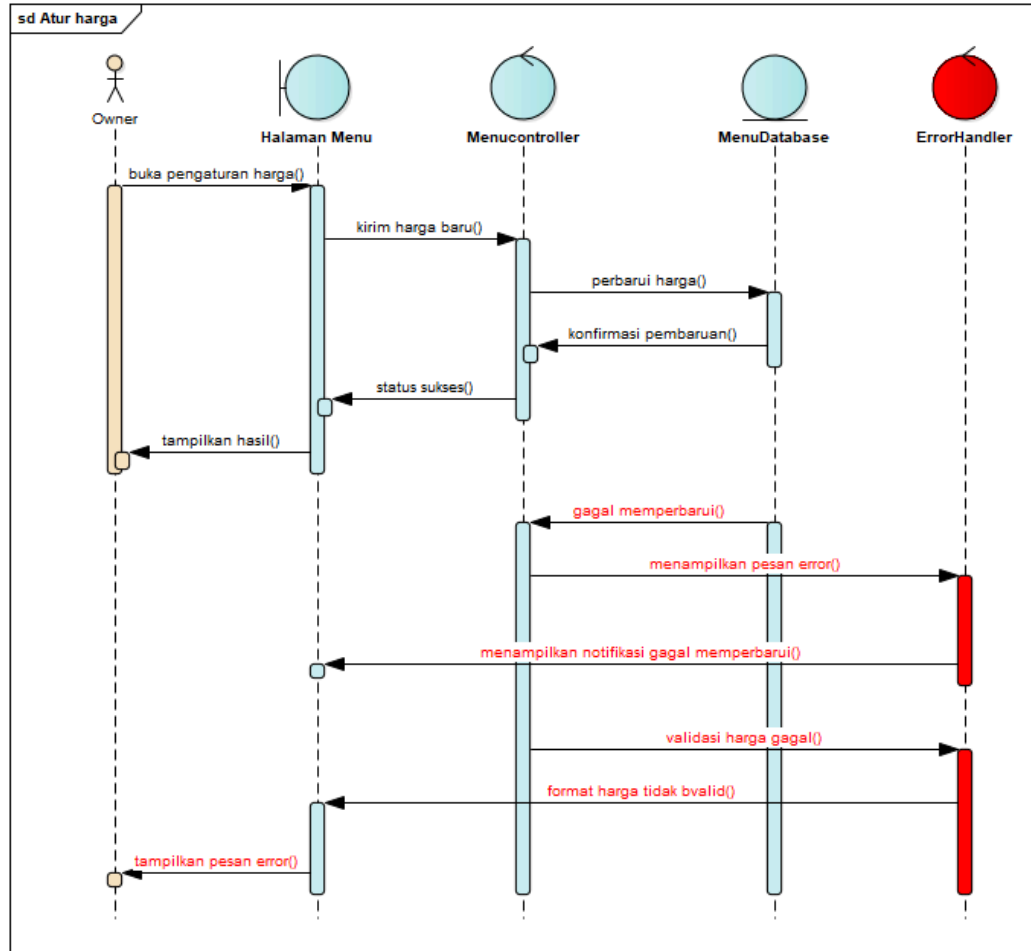
Gambar 2. Sequence Diagram Mengelola Menu

1. Owner membuka antarmuka untuk mengelola menu.
2. Owner memilih untuk menambahkan, mengedit, atau menghapus menu.
3. Owner mengisi data menu (nama, harga, varian).
4. Sistem memvalidasi data input.
5. Sistem menyimpan perubahan ke dalam database.
6. Sistem memberikan status sukses atau error kepada Owner.

Alternate Course

- A1: Admin mengirimkan data tidak valid.
 1. Sistem menampilkan pesan kesalahan dan meminta Admin untuk memperbaiki data.
 2. Proses kembali ke langkah 3.
- A2: Gagal menyimpan ke database (misalnya, koneksi terputus).
 1. Sistem menampilkan pesan kesalahan.
 2. Sistem memberikan opsi untuk mencoba lagi atau membatalkan proses.

B. Atur Harga



Gambar 3. Sequence Diagram Atur Harga

Basic Course

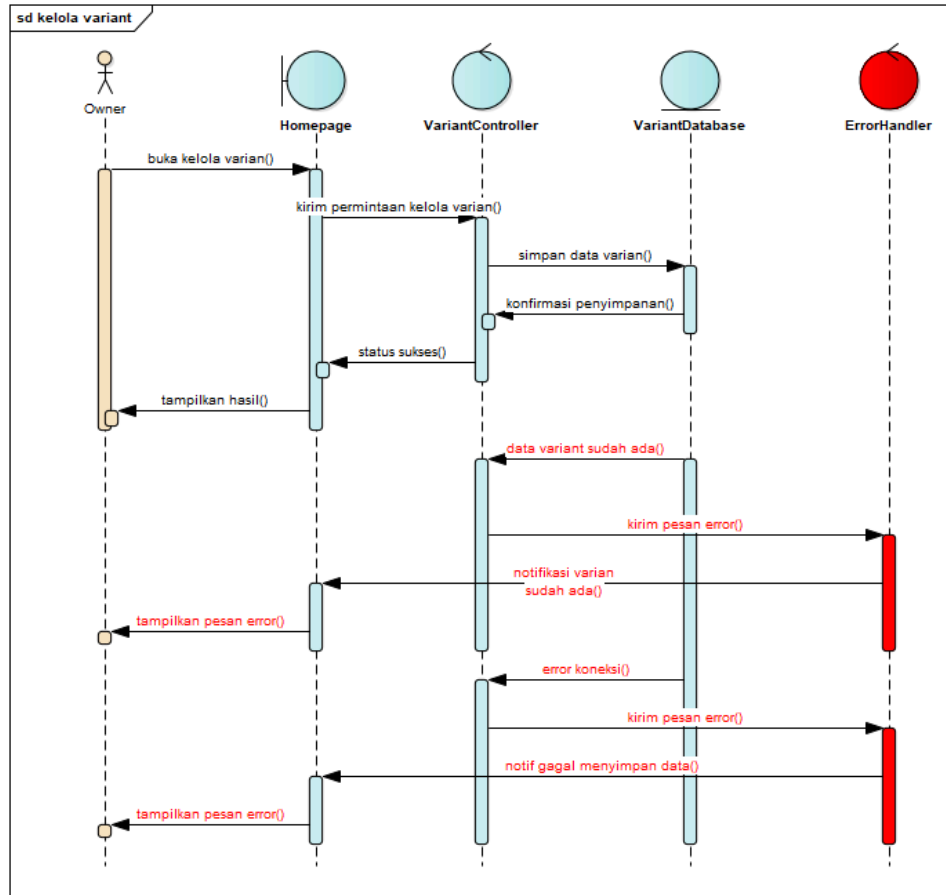
1. Owner membuka antarmuka untuk pengaturan harga.
2. Owner memilih menu yang ingin diatur harganya.
3. Owner memasukkan harga baru.
4. Sistem memvalidasi data input.
5. Sistem menyimpan harga baru ke dalam database.
6. Sistem memberikan status sukses kepada owner.

Alternate Course

- A1: Harga yang dimasukkan tidak valid (contoh: format angka salah).
 - Sistem menampilkan pesan kesalahan.
 - Sistem meminta Owner untuk memperbaiki data.
 - Proses kembali ke langkah 3.
- A2: Gagal menyimpan ke database (misalnya, koneksi database terputus).
 - Sistem menampilkan pesan kesalahan.

- Sistem memberikan opsi untuk mencoba lagi atau membatalkan proses.

C. Kelola Variant



Gambar 4. Sequence Diagram Kelola Variant

Basic Course

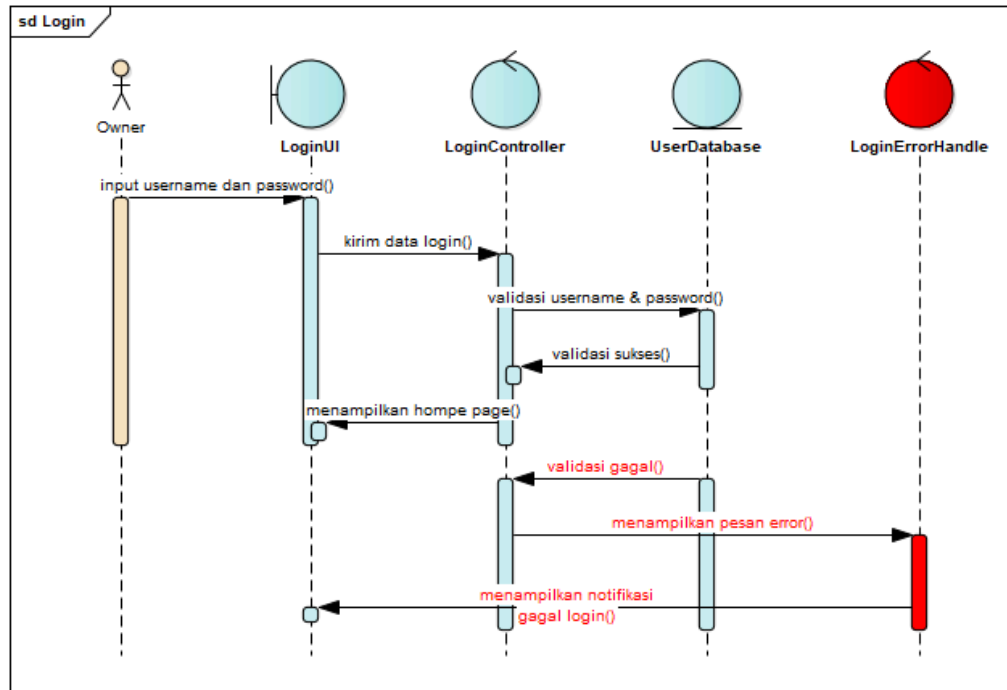
1. Owner membuka antarmuka untuk kelola varian.
2. Owner memilih untuk menambahkan, mengedit, atau menghapus varian.
3. Owner memasukkan data varian (contoh: nama varian dan harga tambahan).
4. Sistem memvalidasi data input.
5. Sistem menyimpan perubahan ke dalam database.
6. Sistem memberikan status sukses kepada Owner.

Alternate Course

- A1: Data varian yang dimasukkan tidak valid (contoh: nama kosong).

- Sistem menampilkan pesan kesalahan.
 - Sistem meminta Owner untuk memperbaiki data.
 - Proses kembali ke langkah 3.
- A2: Gagal menyimpan ke database (contoh: server error).
 - Sistem menampilkan pesan kesalahan.
 - Sistem memberikan opsi untuk mencoba lagi atau membatalkan proses.

D. Login



Gambar 5. Sequence Diagram Login

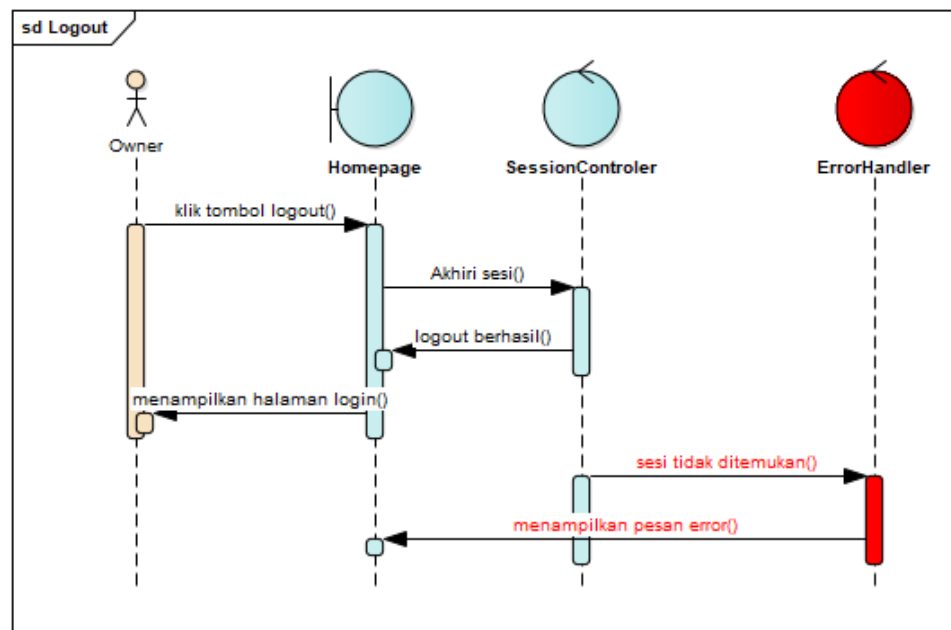
Basic Course

1. Owner memasukkan username dan password melalui LoginUI.
2. LoginUI mengirimkan data login ke LoginController.
3. LoginController memvalidasi data login melalui UserDatabase.
4. UserDatabase mengkonfirmasi validasi berhasil.
5. LoginController mengirimkan status sukses ke LoginUI.
6. LoginUI menampilkan halaman utama kepada Owner.

Alternate Course

- Owner memasukkan username dan password melalui AdminUI.
- LoginUI mengirimkan data login ke LoginController.
- LoginController mencoba memvalidasi data
 - Kemungkinan 1: Username/password salah.
 - LoginController mengirimkan pesan error ke LoginUI: "Username atau password salah".
 - Kemungkinan 2: Akun terkunci.
 - LoginController mengirimkan pesan error ke LoginUI: "Akun terkunci, hubungi owner sistem".
- LoginUI menampilkan pesan error kepada Owner.

E. Logout



Gambar 6. Logout

Basic Course

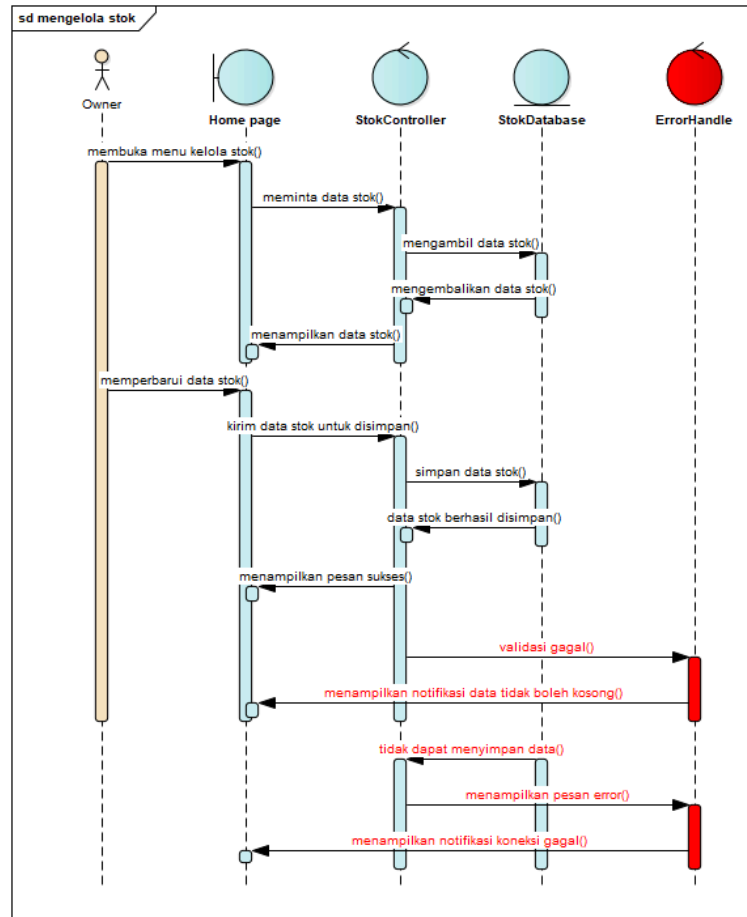
1. Owner menekan tombol logout di Homepage.
2. Homepage mengirimkan permintaan logout ke SessionController.
3. SessionController menghapus sesi login dari sistem.
4. SessionController mengirimkan status sukses ke Homepage.
5. Homepage mengarahkan kembali ke halaman login.

Alternate Course (Gagal):

- Owner menekan tombol logout di Homepage.
- Homepage mengirimkan permintaan logout ke SessionController.

- SessionController mencoba menghapus sesi login:
 - Kemungkinan: Gagal menghapus sesi karena error sistem.
 - SessionController mengirimkan pesan error ke Homepage: "Gagal logout, coba lagi".
- Homepage menampilkan pesan error kepada Owner.

F. Mengelola Stok



Gambar 7. Mengelola Stok

Basic Course

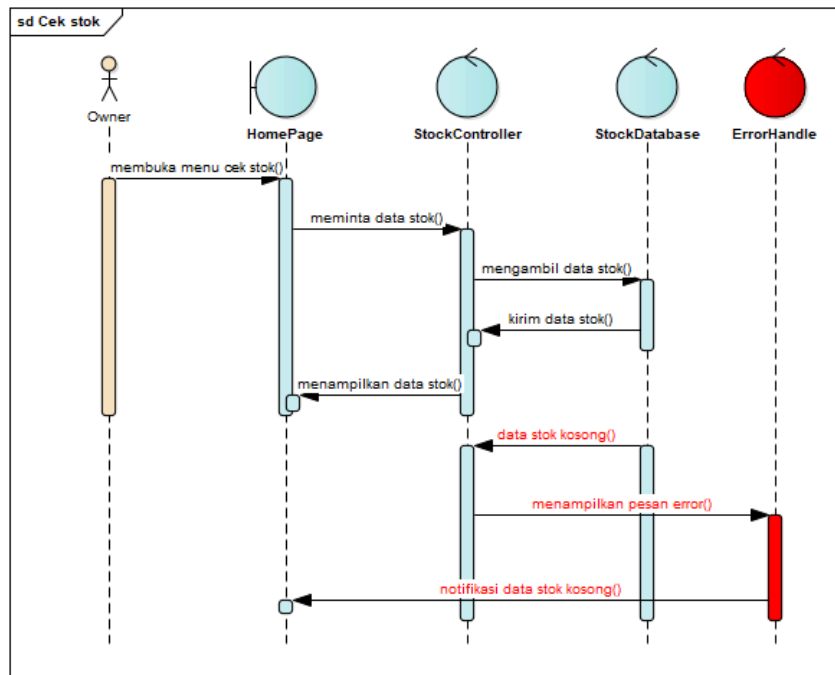
1. Owner membuka halaman pengelolaan stok melalui Homepage.
2. Homepage mengirimkan data stok baru ke StockController.
3. StockController memperbarui data stok di StockDatabase.
4. StockDatabase mengonfirmasi pembaruan berhasil.
5. StockController mengirimkan status sukses ke Homepage.
6. Homepage menampilkan hasil kepada Admin.

Alternate Course (Gagal):

- Owner membuka halaman pengelolaan stok melalui Homepage.
- Homepage mengirimkan data stok baru ke StockController.
- StockController mencoba memperbarui data:
 - Kemungkinan 1: Data tidak valid.

- StockController mengirimkan pesan error ke Homepage: "Data stok tidak valid".
 - Kemungkinan 2: Error koneksi ke StockDatabase.
 - StockController mengirimkan pesan error ke Homepage: "Gagal terhubung ke database".
- Homepage menampilkan pesan error kepada Owner.

G. Cek stok



Gambar 8. Sequence Diagram Cek Stok

Basic Course

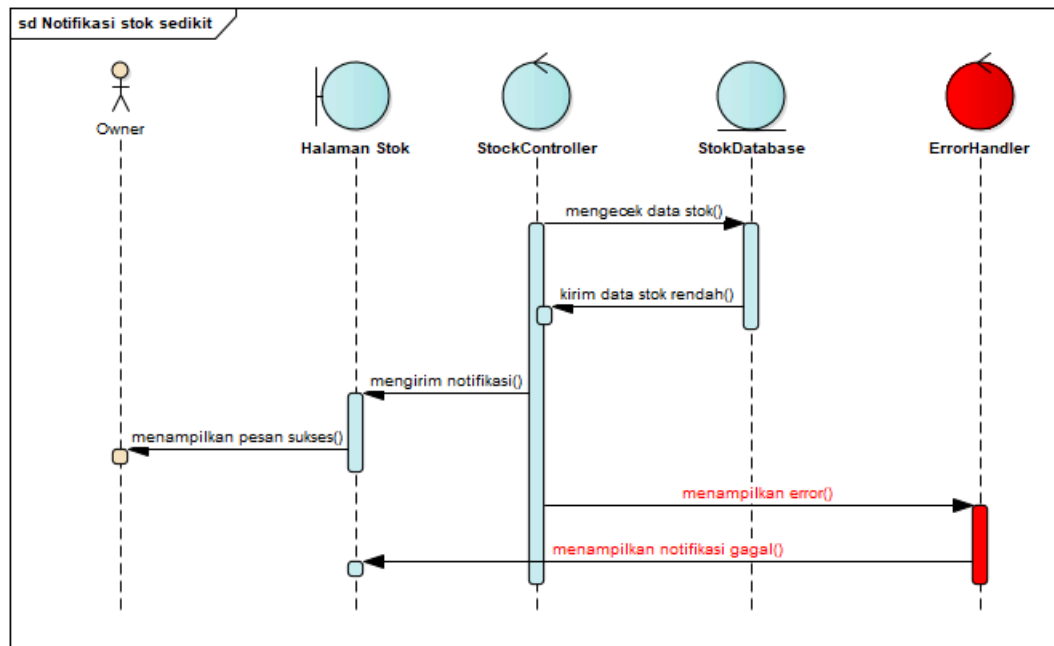
1. Owner membuka fitur cek stok melalui Homepage.
2. Homepage mengirimkan permintaan ke StockController.
3. StockController mengambil data stok dari StockDatabase.
4. StockDatabase mengirimkan data stok ke StockController.
5. StockController mengirimkan data ke Homepage.
6. Homepage menampilkan data stok kepada owner.

Alternate Course

- Owner membuka fitur cek stok melalui Homepage.
- Homepage mengirimkan permintaan ke StockController.

- StockController mencoba mengambil data dari StockDatabase.
 - Kemungkinan: Gagal mendapatkan data.
 - StockDatabase mengirimkan pesan error: "Data stok tidak ditemukan".
- Homepage menampilkan pesan error kepada Owner.

H. Notifikasi Stok Sedikit



Gambar 9. Sequence Diagram Notifikasi Stok Sedikit

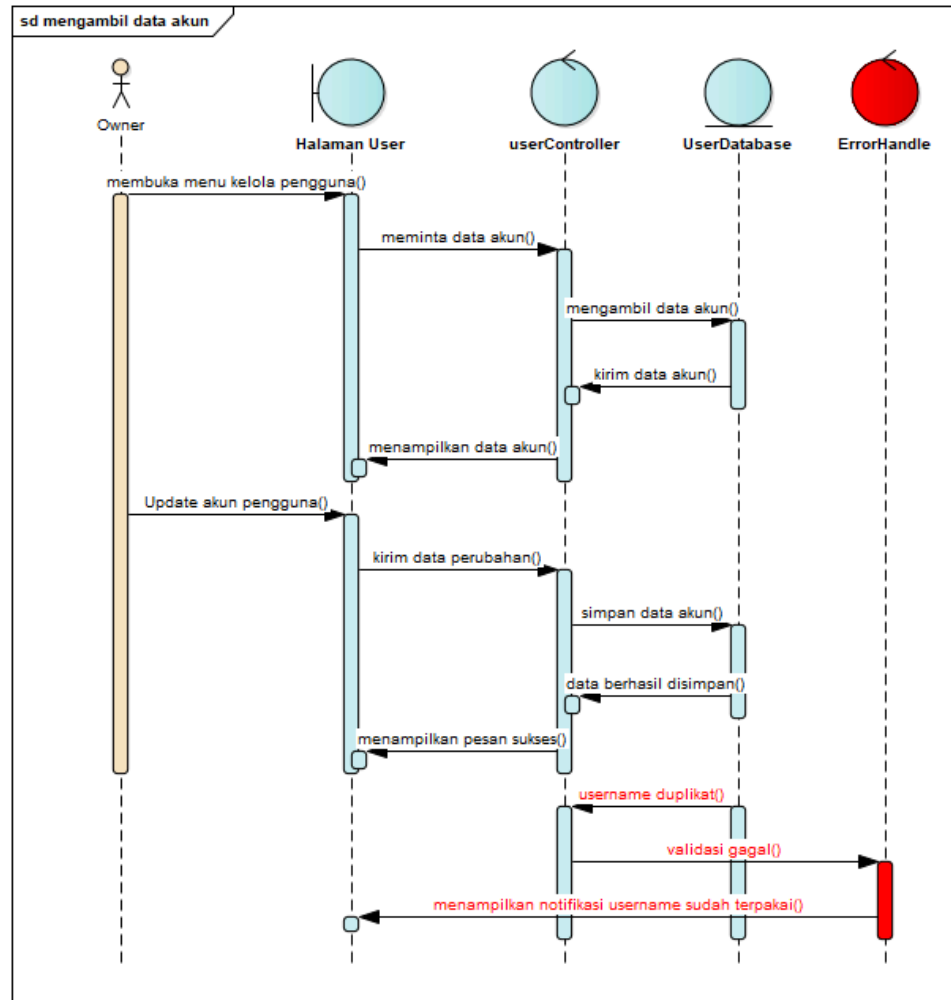
Basic Course

1. Sistem secara otomatis memeriksa stok barang di StockDatabase.
2. Jika ada barang dengan stok sedikit, sistem mengirimkan notifikasi melalui Homepage.
3. Owner menerima notifikasi stok barang yang hampir habis.

Alternate Course

- Sistem mencoba memeriksa stok barang.
 - Kemungkinan: Gagal mendapatkan data dari StockDatabase.
 - Sistem menampilkan pesan error: "Gagal memuat data stok".

I. Mengelola Akun Pengguna



Gambar 10. Sequence Diagram Mengambil Data Akun

Basic Course

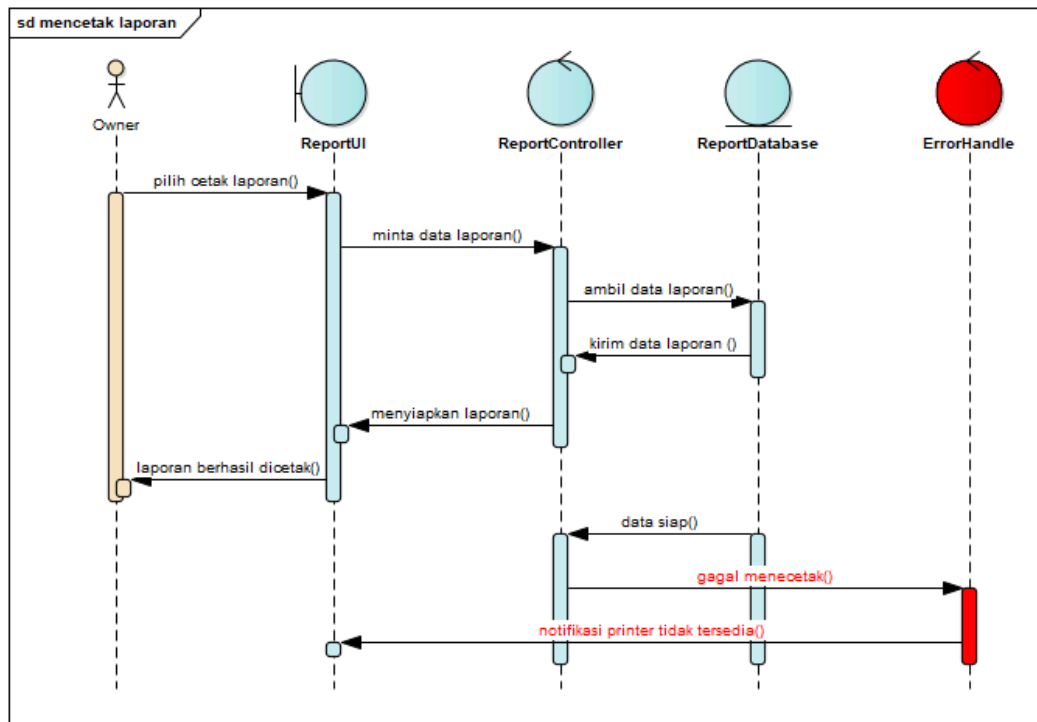
1. Owner membuka fitur kelola akun melalui Halaman User.
2. Halaman user mengirimkan data akun baru ke userController.
3. userController menyimpan data ke userDatabase.
4. userDatabase mengkonfirmasi penyimpanan berhasil.
5. userController mengirimkan status sukses ke Halaman User.
6. HalamanUser menampilkan hasil kepada Owner.

Alternate Course

1. Owner membuka fitur kelola akun melalui Halaman User.
2. Halaman user mengirimkan data akun baru ke userController.

3. userController mencoba menyimpan data:
 - Kemungkinan: Data akun tidak valid.
 - AccountController mengirimkan pesan error ke halaman user: "Data akun tidak valid".
4. Halaman user menampilkan pesan error kepada Owner.

J. Mencetak Laporan



Gambar 11. Sequence Diagram Mencetak Laporan

Basic Course

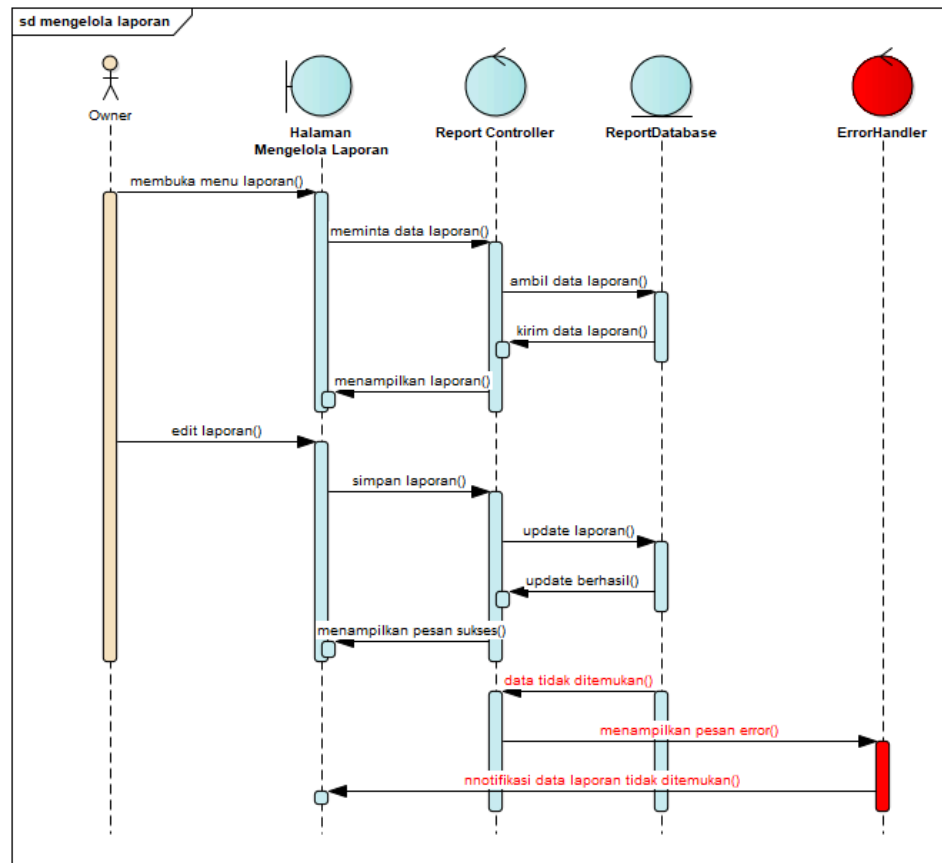
1. Owner memilih opsi cetak laporan melalui reportUI.
2. ReportUI mengirimkan permintaan cetak ke ReportController.
3. ReportController mengambil data dari ReportDatabase.
4. ReportDatabase mengirimkan data ke ReportController.
5. ReportController mengirimkan data ke printer.
6. Laporan berhasil dicetak.

Alternate Course

- Owner memilih opsi cetak laporan melalui reportUI.
- reportUI mengirimkan permintaan cetak ke ReportController.

- ReportController mencoba mengambil data dari ReportDatabase.
 - Kemungkinan: Data laporan tidak tersedia.
 - Sistem menampilkan pesan error: "Laporan tidak tersedia".

K. Mengelola Laporan



Gambar 12. Sequence Diagram Mengelola Laporan

Basic Course

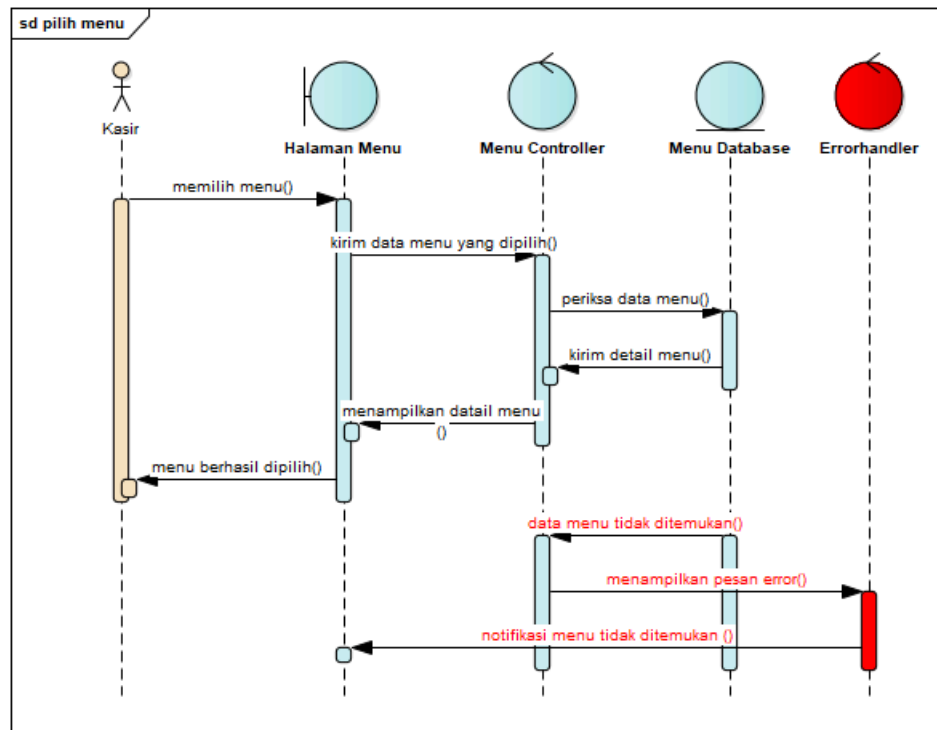
1. Owner membuka fitur kelola laporan melalui Halaman mengelola laporan.
2. Halaman mengelola laporan mengirimkan data laporan baru ke ReportController.
3. ReportController menyimpan data laporan ke ReportDatabase.
4. ReportDatabase mengonfirmasi penyimpanan berhasil.
5. ReportController mengirimkan status sukses ke Halaman mengelola laporan.
6. Halaman mengelola laporan menampilkan hasil kepada Owner.

Alternate Course

- Owner membuka fitur kelola laporan melalui Halaman mengelola laporan.

- Halaman mengelola laporan mengirimkan data laporan baru ke ReportController.
- ReportController mencoba menyimpan data:
 - Kemungkinan: Error validasi data.
 - Sistem menampilkan pesan error: "Data laporan tidak valid".

L. Pilih Menu



Gambar 13. Sequence Diagram Pilih Menu

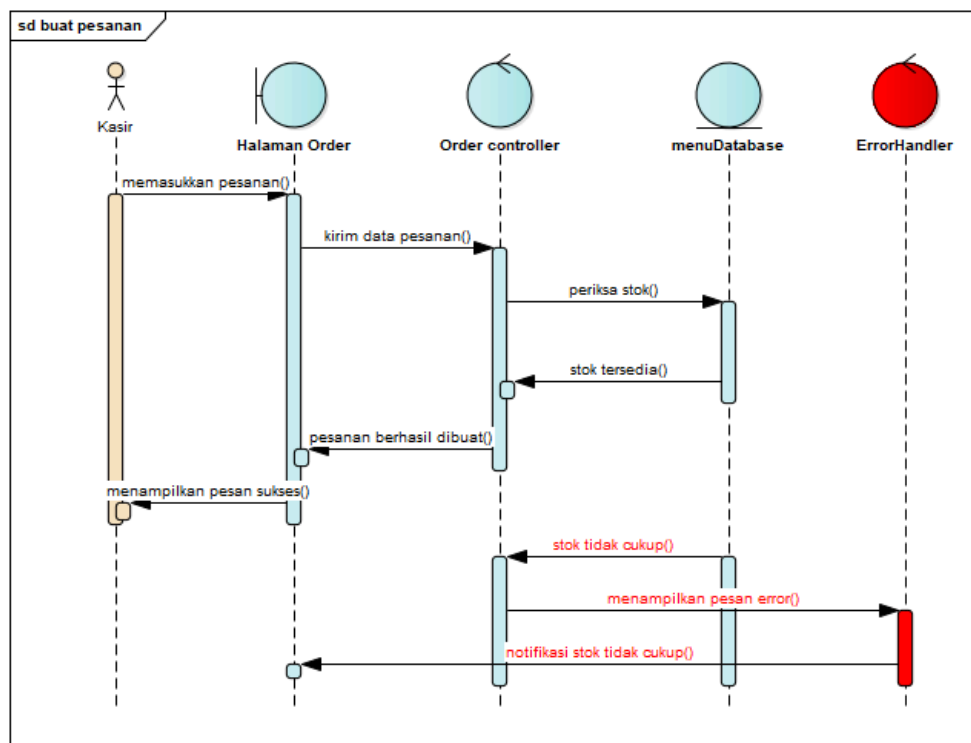
Basic Course

1. Kasir membuka daftar menu melalui HalamanMenu.
2. HalamanMenu meminta data menu dari MenuController.
3. MenuController mengambil data menu dari MenuDatabase.
4. MenuDatabase mengirimkan data menu ke MenuController.
5. MenuController mengirimkan data menu ke HalamanMenu.
6. HalamanMenu menampilkan daftar menu kepada Kasir.
7. Kasir memilih item dari daftar menu.
8. HalamanMenu menambahkan item yang dipilih ke daftar pesanan.

Alternate Course

- Kasir membuka daftar menu melalui HalamanMenu.
- HalamanMenu meminta data menu dari MenuController.
- MenuController mencoba mengambil data dari MenuDatabase:
 - Kemungkinan: Data menu tidak tersedia.
 - Sistem menampilkan pesan error: "Daftar menu tidak dapat dimuat".
- Kasir menghubungi Owner untuk memperbaiki data menu.

M. Buat Pesanan



Gambar 14. Sequence Diagram Buat Pesanan

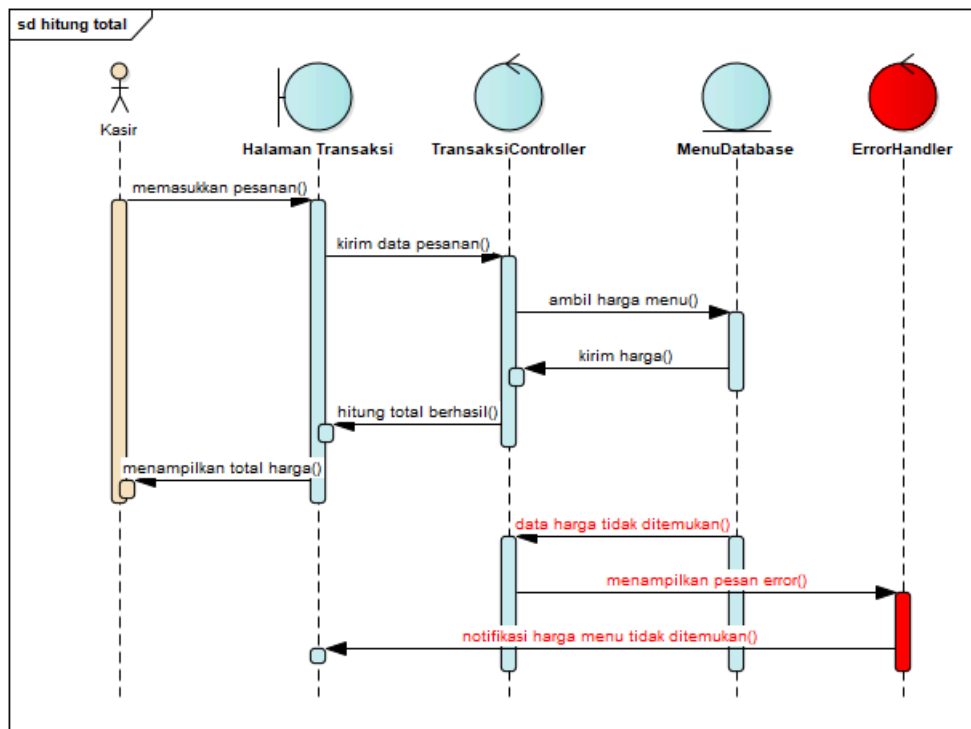
Basic Course

1. Kasir memilih opsi buat pesanan di HalamanOrder.
2. HalamanOrder mengirimkan data pesanan ke OrderController.
3. OrderController menyimpan data pesanan ke menuDatabase.
4. menuDatabase mengonfirmasi penyimpanan berhasil.
5. OrderController mengirimkan status sukses ke HalamanOrder.
6. HalamanOrder menampilkan notifikasi sukses kepada Kasir.

Alternate Course

- Kasir memilih opsi buat pesanan di HalamanOrder.
- HalamanOrder mengirimkan data pesanan ke OrderController.
- OrderController mencoba menyimpan data:
 - Kemungkinan 1: Data pesanan tidak valid (contoh: jumlah negatif).
 - Sistem menampilkan pesan error: "Data pesanan tidak valid".
 - Kemungkinan 2: Sistem gagal menyimpan data ke menuDatabase.
 - Sistem menampilkan pesan error: "Gagal menyimpan pesanan, coba lagi".
- Kasir memperbaiki data atau mencoba lagi.

N. Hitung Total



Gambar 15. Hitung Total

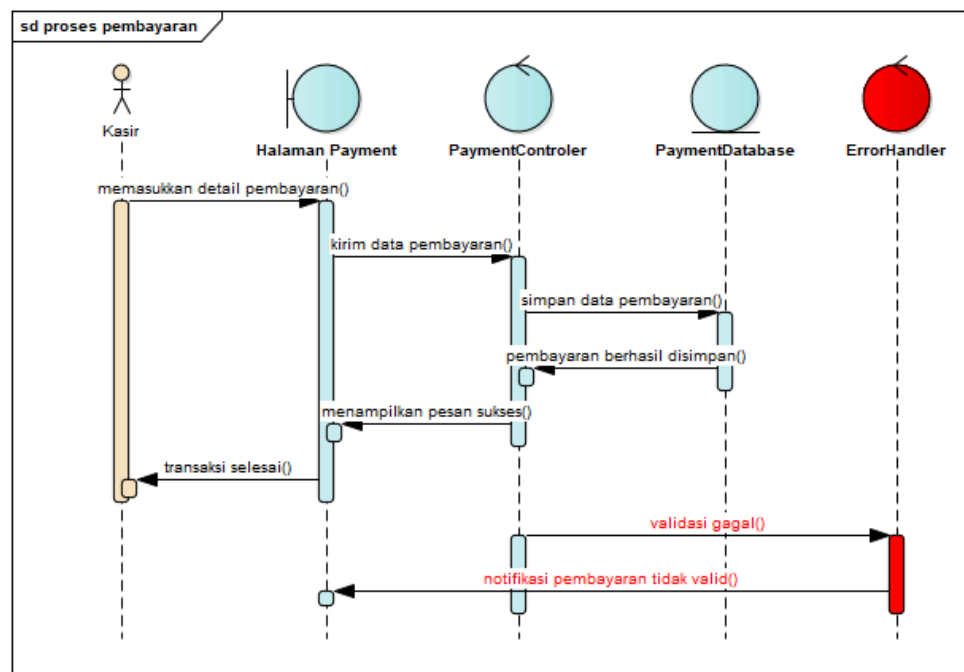
Basic Course

1. Kasir memilih opsi hitung total di Halaman Transaksi.
2. Halaman Transaksi mengirimkan data pesanan ke TransaksiController.
3. TransaksiController menghitung total pesanan berdasarkan harga di MenuDatabase.
4. TransaksiController mengirimkan total pesanan ke Halaman transaksi.
5. Halaman Transaksi menampilkan total kepada Kasir.

Alternate Course

- Kasir memilih opsi hitung total di Halaman transaksi.
- Halaman transaksi mengirimkan data pesanan ke TransaksiController.
- TransaksiController mencoba menghitung total:
 - Kemungkinan: Data harga tidak tersedia di MenuDatabase.
 - Sistem menampilkan pesan error: "Data harga tidak tersedia untuk beberapa item".
- Kasir memperbaiki data pesanan atau menghubungi Owner.

O. Proses Pembayaran



Gambar 16. Proses Pembayaran

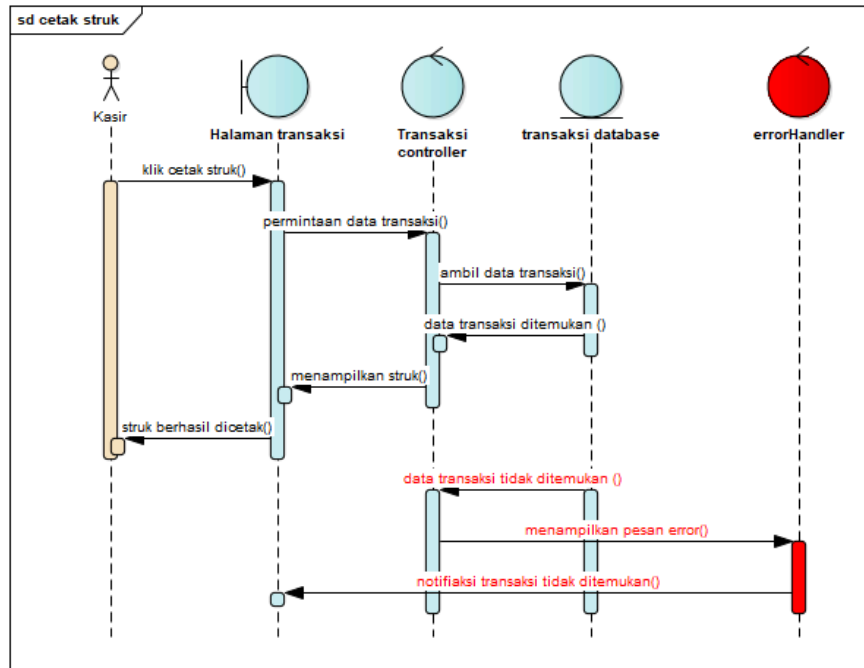
Basic Course

1. Kasir memilih opsi proses pembayaran di Halaman Payment.
2. Halaman Payment mengirimkan data pembayaran ke PaymentController.
3. PaymentController memvalidasi pembayaran (contoh: cek uang yang diterima vs total transaksi).
4. Jika valid, PaymentController menyimpan data pembayaran ke TransactionDatabase.
5. PaymentController mengirimkan status sukses ke Halaman Payment.
6. Halaman Payment menampilkan notifikasi sukses kepada Kasir.

Alternate Course

- Kasir memilih opsi proses pembayaran di Halaman Payment.
- Halaman Payment mengirimkan data pembayaran ke PaymentController.
- PaymentController memvalidasi pembayaran:
 - Kemungkinan 1: Jumlah uang yang diterima kurang.
 - Sistem menampilkan pesan error: "Pembayaran kurang, mohon tambahkan uang".
 - Kemungkinan 2: Sistem gagal menyimpan data ke TransactionDatabase.
 - Sistem menampilkan pesan error: "Gagal menyimpan data transaksi, coba lagi".
- Kasir memperbaiki data atau mencoba lagi.

P. Cetak Struk



Gambar 17. Sequence Diagram Cetak Struk

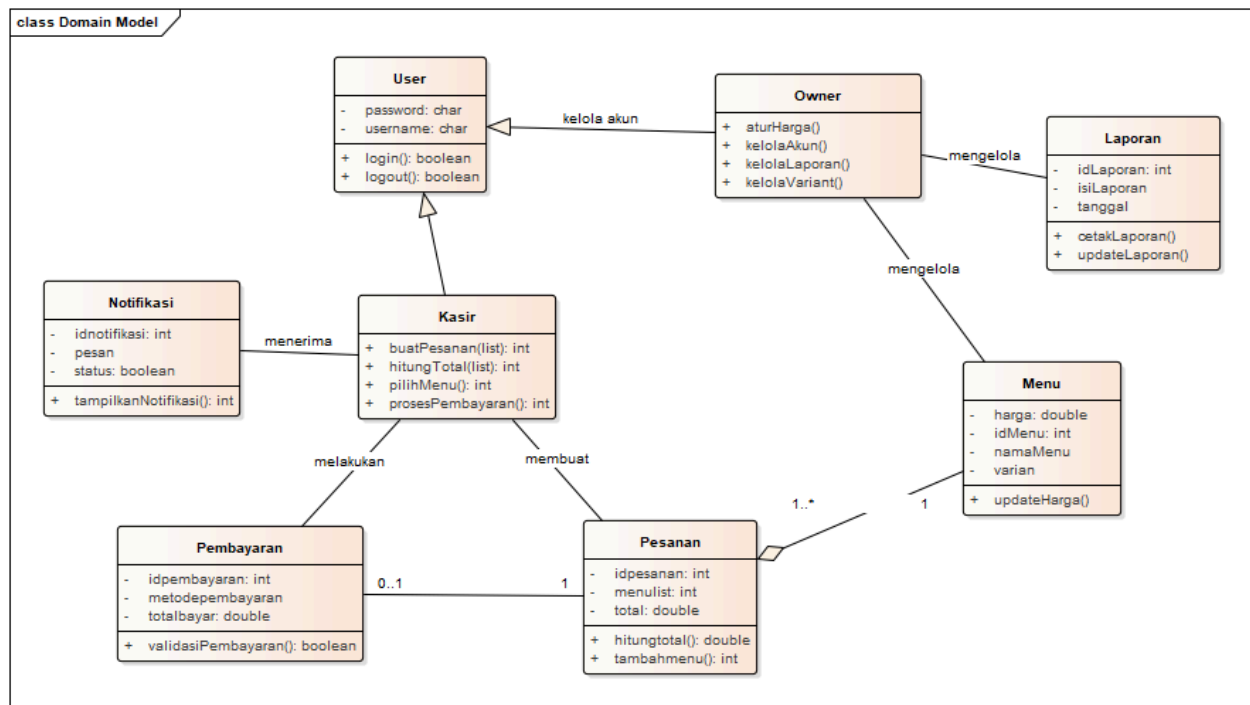
Basic Course

1. Kasir memilih opsi cetak struk di Halaman transaksi.
2. Halaman transaksi mengirimkan permintaan cetak ke transaksiController.
3. TransaksiController mengambil data transaksi dari TransactionDatabase.
4. TransaksiController mengirimkan data ke printer.
5. Struk berhasil dicetak.

Alternate Course

1. Kasir memilih opsi cetak struk di Halaman transaksi.
2. TransaksiController gagal mendapatkan data dari TransactionDatabase.
 - Sistem menampilkan pesan error: "Gagal mencetak struk".

3.2.2 Class Diagram



Gambar 18. Class Diagram

1. User

- Merupakan superclass yang digunakan untuk Owner dan Kasir.
- Memiliki atribut username dan password untuk autentikasi.
- Metode: `login()` dan `logout()` untuk mengelola sesi pengguna.

2. Admin

- Mewarisi User.
- Memiliki fungsi spesifik seperti:
 - `aturHarga()`: Mengubah harga menu.
 - `kelolaVarian()`: Menambahkan/mengubah varian menu.
 - `kelolaAkun()`: Menambah atau menghapus akun pengguna.
 - `kelolaLaporan()`: Mengelola laporan operasional.

3. Kasir

- Mewarisi User.
- Tugas utamanya melibatkan transaksi dan pesanan:
 - `pilihMenu()`: Memilih menu untuk pelanggan.
 - `buatPesanan()`: Membuat daftar pesanan berdasarkan pilihan menu.
 - `hitungTotal()`: Menghitung total dari pesanan.
 - `prosesPembayaran()`: Memvalidasi pembayaran dan menyelesaikan transaksi.

4. Menu

- Merepresentasikan item menu.
- Atribut: idMenu, namaMenu, harga, dan varian.
- Metode:
 - updateHarga(): Untuk mengubah harga menu.

5. Pesanan

- Representasi daftar pesanan pelanggan.
- Atribut:
 - menuList: Berisi daftar menu yang dipesan.
 - total: Total harga pesanan.
- Metode:
 - hitungTotal(): Menghitung total pesanan.
 - tambahMenu(): Menambahkan item ke pesanan.

6. Pembayaran

- Merepresentasikan proses pembayaran.
- Atribut:
 - idPembayaran, totalBayar, dan metodePembayaran.
- Metode:
 - validasiPembayaran(): Memastikan pembayaran valid.

7. Laporan

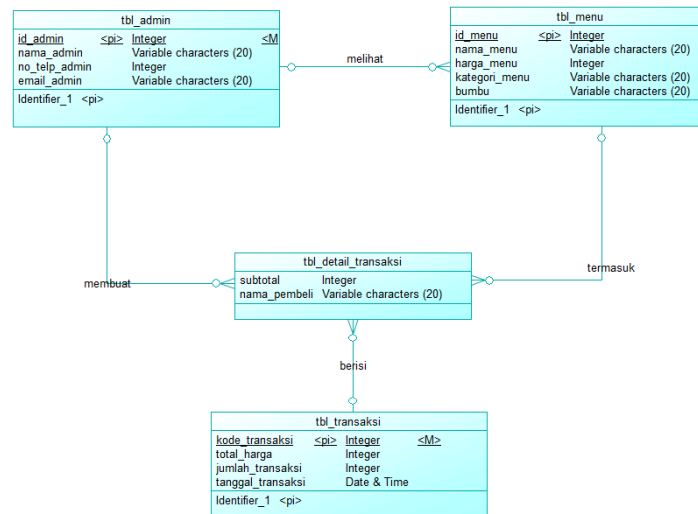
- Digunakan oleh Admin untuk mengelola laporan.
- Atribut: idLaporan, tanggal, dan isiLaporan.
- Metode:
 - cetakLaporan(): Mencetak laporan.
 - updateLaporan(): Memperbarui isi laporan.

8. Notifikasi

- Digunakan untuk memberikan pemberitahuan stok sedikit atau error lainnya.
- Atribut: idNotifikasi, pesan, dan status.
- Metode:
 - tampilkanNotifikasi(): Menampilkan pesan kepada pengguna.

3.3 Perancangan Data

3.3.1 CDM (Conceptual Data Model)



Gambar 19. CDM

CDM adalah model data tingkat tinggi yang menggambarkan struktur data secara konseptual tanpa memperhatikan implementasi fisik. CDM digunakan untuk memahami hubungan antar entitas, atribut, dan keterhubungannya di dalam sistem.

Berdasarkan CDM di atas, terdapat beberapa entitas dengan atributnya masing-masing, serta relasi di antara entitas tersebut:

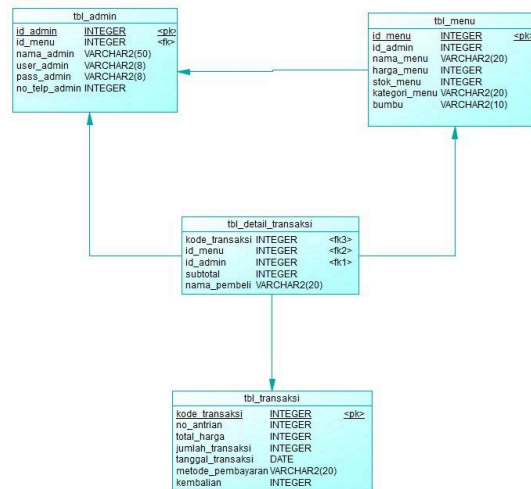
1. Entitas **tbl_admin**
 - Atribut:
 - **id_admin** : Primary key (Identifier)
 - **nama_admin** : Variabel karakter (20)
 - **no_hp_admin** : Variabel karakter (20)
 - Fungsi: Menyimpan informasi admin dalam sistem.
2. Entitas **tbl_menu**
 - Atribut:
 - **id_menu** : Primary key (Identifier)
 - **harga_menu** : Integer
 - **nama_menu** : Variabel karakter (20)
 - **button** : Variabel karakter (20)
 - Fungsi: Menyimpan data menu dan harga dalam sistem.

3. Entitas `tbl_detail_transaksi`
 - Atribut:
 - `subtotal` : Integer
 - `nama_pembeli` : Variabel karakter (20)
 - Fungsi: Menyimpan rincian transaksi.
4. Entitas `tbl_transaksi`
 - Atribut:
 - `kode_transaksi` : Primary key (Identifier)
 - `total_harga` : Integer
 - `tgl_transaksi` : Date
 - `lama_transaksi` : Date & Time
 - Fungsi: Menyimpan data transaksi utama.

Relasi antar Entitas:

- Relasi `tbl_admin` ke `tbl_menu`: Relasi "melihat" yang menunjukkan admin dapat melihat data menu.
- Relasi `tbl_menu` ke `tbl_detail_transaksi`: Relasi "termasuk" menunjukkan menu menjadi bagian dari detail transaksi.
- Relasi `tbl_detail_transaksi` ke `tbl_transaksi`: Relasi "berisi" yang menunjukkan detail transaksi adalah bagian dari transaksi utama.

3.3.2 PDM (Physical Data Model)



Gambar 20. PDM

PDM adalah model data yang lebih rinci dibanding CDM karena sudah mendefinisikan tipe data, panjang karakter, serta atribut lainnya yang spesifik untuk implementasi database. PDM digunakan sebagai acuan langsung untuk pembuatan tabel dalam sistem database.

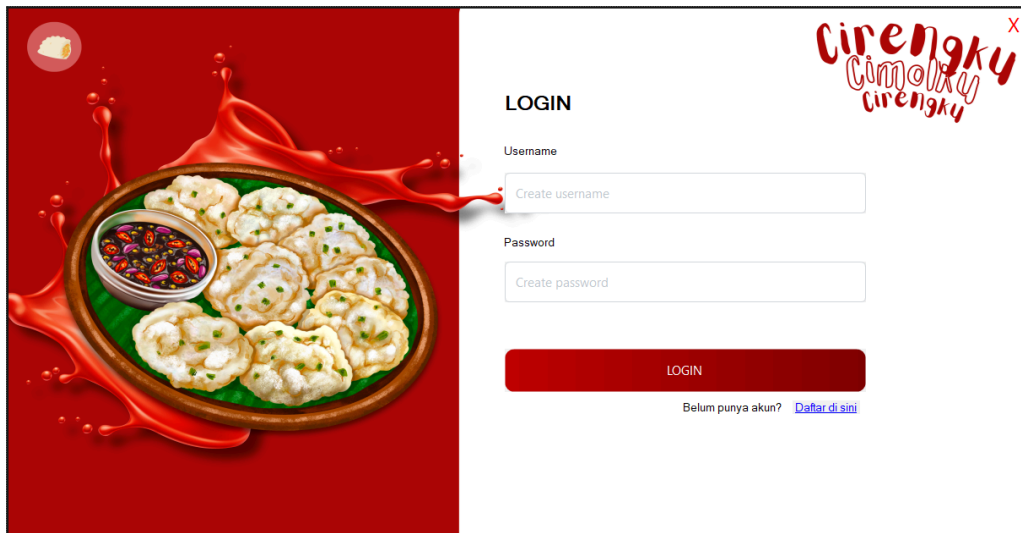
Berdasarkan PDM di atas, struktur data diubah menjadi tabel-tabel dengan tipe data spesifik:

1. Tabel **tbl_admin**
 - Atribut:
 - **id_admin** : INTEGER (Primary Key)
 - **nama_admin** : VARCHAR(20)
 - **no_hp_admin** : VARCHAR(20)
 - Deskripsi: Menyimpan data admin dengan tipe data spesifik.
2. Tabel **tbl_menu**
 - Atribut:
 - **id_menu** : INTEGER (Primary Key)
 - **harga_menu** : INTEGER
 - **nama_menu** : VARCHAR(20)
 - **button** : VARCHAR(20)
 - Deskripsi: Menyimpan data menu yang disertai harga.
3. Tabel **tbl_detail_transaksi**
 - Atribut:
 - **id_transaksi** : INTEGER (Foreign Key)
 - **id_menu** : INTEGER (Foreign Key)
 - **nama_pembeli** : VARCHAR(20)
 - **subtotal** : INTEGER

- Deskripsi: Menyimpan rincian transaksi dengan keterhubungan ke tabel transaksi dan menu.
- 4. Tabel tbl_transaksi
 - Atribut:
 - kode_transaksi : INTEGER (Primary Key)
 - total_harga : INTEGER
 - tgl_transaksi : DATE
 - lama_transaksi : DATETIME
 - Deskripsi: Menyimpan data transaksi utama yang berkaitan dengan detail transaksi.

3.4 Tampilan Aplikasi

3.4.1 Tampilan Login



Cirengky
Cimolky
Cirengky

LOGIN

Username

Create username

Password

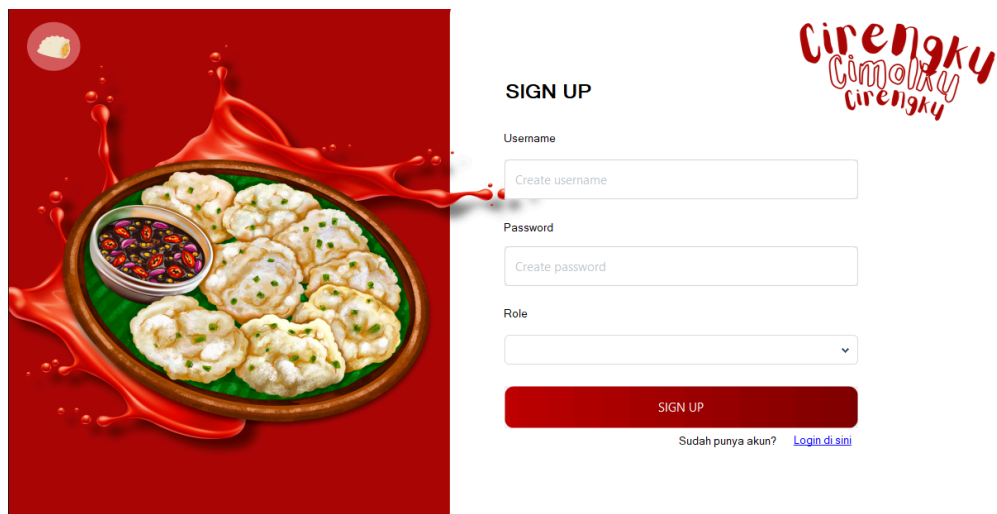
Create password

LOGIN

Belum punya akun? [Daftar di sini](#)

Gambar 21. Tampilan Login Aplikasi

3.4.2 Tampilan Sign Up



Cirengky
Cimolky
Cirengky

SIGN UP

Username

Create username

Password

Create password

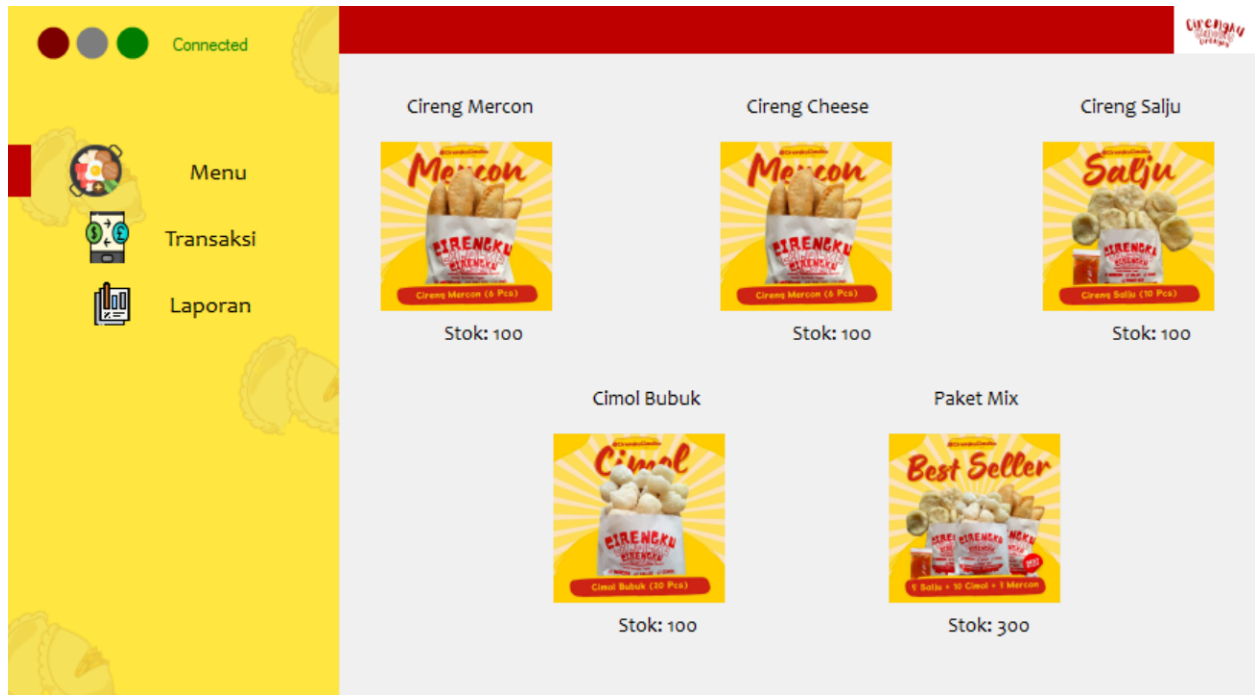
Role

SIGN UP

Sudah punya akun? [Login di sini](#)

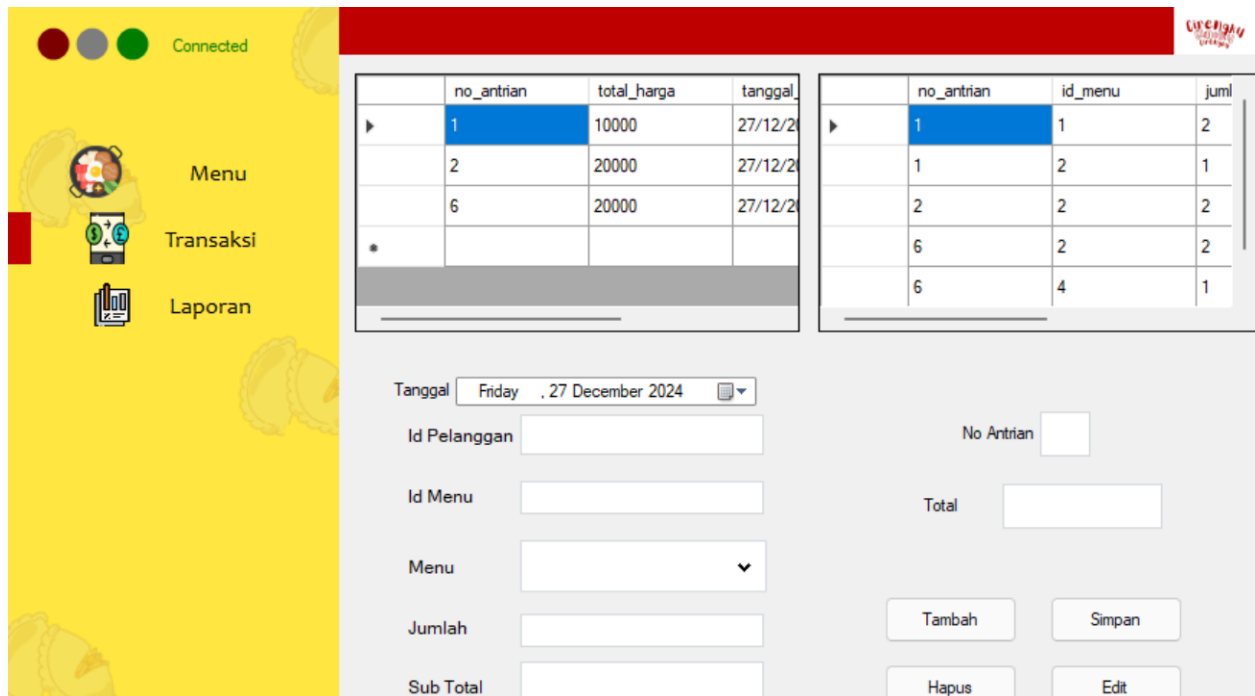
Gambar 22. Tampilan Sign Up Aplikasi

3.4.3 Tampilan Awal (Panel Menu)



Gambar 23. Tampilan Awal Aplikasi

3.4.5 Tampilan Panel Transaksi



Gambar 24. Tampilan Transaksi

3.4.6 Tampilan Panel Laporan



Gambar 25. Tampilan Cetak Laporan

BAB III

KESIMPULAN DAN SARAN

3.1 Kesimpulan

Aplikasi kasir yang dikembangkan untuk usaha kecil dan menengah (UMKM) *Cirengku Cimolku* telah berhasil mencapai tujuan utamanya, yaitu membangun sistem pencatatan transaksi yang lebih terorganisir dan efektif. Proses pengelolaan transaksi penjualan yang sebelumnya dilakukan secara manual telah diubah menjadi lebih modern dan sistematis berkat penggunaan sistem berbasis desktop ini. Beberapa fitur penting yang ditawarkan oleh aplikasi ini termasuk manajemen menu, pengelolaan stok, pencatatan transaksi otomatis, dan pembuatan laporan yang terintegrasi. Sistem ini juga berhasil menangani berbagai kebutuhan operasional pada level manajemen dan operasional dengan membagi tugas antara pemilik dan kasir. Dengan notifikasi stok dan pemantauan langsung, pemilik membantu membuat keputusan strategis tentang pengelolaan persediaan. Secara keseluruhan, aplikasi ini telah membantu UMKM *Cirengku Cimolku* meningkatkan efisiensi operasional dan menghadapi tantangan di era digitalisasi.

3.2 Saran

Hasil dari pengembangan dan pelaksanaan aplikasi kasir untuk UMKM *Cirengku Cimolku* menghasilkan beberapa rekomendasi yang dapat digunakan untuk meningkatkan sistem di masa mendatang. Dengan membangun fitur analisis penjualan yang lebih mendalam, pemilik bisnis dapat lebih memahami apa yang dikonsumsi pelanggan dan membuat pilihan strategis yang lebih baik. Pengembangan aplikasi berbasis ponsel akan sangat membantu pemilik bisnis untuk memantau penjualan dari manapun mereka berada mengingat tingkat mobilitas yang meningkat. Untuk membuat transaksi lebih mudah bagi pelanggan, diperlukan penggabungan berbagai metode pembayaran digital untuk mengikuti tren pembayaran digital yang semakin berkembang.

Link Demo : <https://youtu.be/Dg8Nf83E0fw>



Gambar Dokumentasi Wawancara

KANG		WONG		A910		TOTAL	
3UTAMA	TOTAL	TSI	UTAMA	PUTAMA	TOTAL		
600.000	750.000	200.000	200.000	2.700.000	2.700.000		
100.000	150.000			1.200.000	1.200.000		
	0			1.500.000	1.500.000		
800.000	1.500.000	1/2	800.000	1.700.000	1.700.000		
600.000	800.000	2/3		450.000	450.000		
600.000	300.000			600.000	600.000		
	6.000.000			1.000.000	1.000.000		
200.000	900.000			1.200.000	1.200.000		
100.000	180.000			1.000.000	1.000.000		
	1.350.000	11.000		1.200.000	1.200.000		
	1.050.000		200.000	1.500.000	1.500.000		
300.000	550.000			1.500.000	1.500.000		
500.000	250.000			1.700.000	1.700.000		
300.000	0		1.250.000	1.500.000	1.500.000		
200.000	0			800.000	800.000		
500.000	1.000.000			800.000	800.000		
				0	0		

Gambar Pencatatan Manual Cirengku Cimolku

