

LAPORAN PRAKTIK PEMROGRAMAN

Mobile Programming, 2024

NIM.
(2021 53 032)

UJIAN AKHIR SEMESTER
(Aplikasi CRUD Fattah
Barbershop)

Nama
(Natasya Aditiya)



Aplikasi CRUD Fattah Barbershop

1. Source Code

```
# memasukkan library flet ke aplikasi
# import flet as ft
import flet
from flet import *
import datetime
import mysql.connector

# buat koneksi ke database SQL
koneksi_db = mysql.connector.connect(host = "localhost", user = "root", password = "", database
= "fattah_crud_mobile")
cursor = koneksi_db.cursor()

class FormMember(UserControl):
    # class untuk halaman mata kuliah
    def build(member) :

        # tgl lhr member
        def ubah_tanggal_lhr(e):
            tgl_baru = member.opsi_tanggal.value
            member.inputan_tgl_lahir.value = tgl_baru.date()
            member.update()

        def opsi_tanggal_lhr_dismissed(e):
            tgl_baru = member.inputan_tgl_lahir.value
            member.inputan_tgl_lahir.value = tgl_baru
            member.update()

        member.opsi_tanggal = DatePicker(
            on_change=ubah_tanggal_lhr,
            on_dismiss=opsi_tanggal_lhr_dismissed,
            first_date=datetime.datetime(2023, 10, 1),
            last_date=datetime.datetime(2024, 10, 1),
        )

        # tgl member join
        def ubah_tanggal_lhr_member(e):
            tgl_baru = member.opsi_tanggal_member.value
            member.inputan_tgl_member.value = tgl_baru.date()
            member.update()

        def opsi_tanggal_lhr_dismissed_member(e):
            tgl_baru = member.inputan_tgl_member.value
            member.inputan_tgl_member.value = tgl_baru
            member.update()

        member.opsi_tanggal_member = DatePicker(
            on_change=ubah_tanggal_lhr_member,
```

```

        on_dismiss=opsi_tanggal_lhr_dismissed_member,
        first_date=datetime.datetime(2023, 10, 1),
        last_date=datetime.datetime(2024, 10, 1),
    )
    # buat variabel inputan
    member.inputan_id = TextField(visible = False, expand = True)
    member.inputan_nama = TextField(label = "Nama", hint_text = "Nama", expand = True)
    member.inputan_jekel = Dropdown(label = "JeKel", hint_text = "Lk or PR ", expand = True,
options=[dropdown.Option("Laki-laki"), dropdown.Option("Perempuan")],)
    member.inputan_tgl_lahir = TextField(label = "Tanggal lahir", hint_text = "Tgl Lahir", expand =
True)
    member.inputan_alamat = TextField(label = "Alamat Member", hint_text = "Alamat
Member", expand = True)
    member.inputan_telp = TextField(label = "Telepon", hint_text = "Telepon", expand = True)
    member.inputan_tgl_member = TextField(label = "Tanggal Member", hint_text = "Tanggal
Gabung Member", expand = True)
    member.snack_bar_berhasil = SnackBar( Text("Operasi berhasil"), bgcolor="green")

# memuat tabel data
def tampil_member(e):
    # Merefresh halaman & menampilkan notif
    member.data_member.rows.clear()
    cursor.execute("SELECT * FROM membership")
    result = cursor.fetchall()
    # menampilkan ulang data
    columns = [column[0] for column in cursor.description]
    rows = [dict(zip(columns,row)) for row in result]
    for row in rows:
        member.data_member.rows.append(
            DataRow(
                cells = [
                    DataCell(Text(row['id'])),
                    DataCell(Text(row['nama'])),
                    DataCell(Text(row['jekel'])),
                    DataCell(Text(row['tgl_lahir'])),
                    DataCell(Text(row['alamat'])),
                    DataCell(Text(row['telp'])),
                    DataCell(Text(row['tgl_member'])),
                    DataCell(
                        Row([
                            IconButton("EDIT_OUTLINED", icon_color = "grey", data =
row,on_click=tampil_dialog_ubah_member ),
                            IconButton("DELETE_OUTLINE_OUTLINED", icon_color = "red", data =
row,on_click=hapus_member ),
                        ])
                    )
                ],
            ),
        ]
    )

```

fungsi menampilkan dialog form entri

```

def tampil_dialog_member(e):
    member.inputan_id.value = ""
    member.inputan_nama.value = ""
    member.inputan_jekel.value = ""
    member.inputan_tgl_lahir.value = ""
    member.inputan_alamat.value = ""
    member.inputan_telp.value = ""
    member.inputan_tgl_member.value = ""
    member.dialog.open = True
    member.update()

def tampil_dialog_ubah_member(e):
    member.inputan_id.value = e.control.data['id']
    member.inputan_nama.value = e.control.data['nama']
    member.inputan_jekel.value = e.control.data['jekel']
    member.inputan_tgl_lahir.value = e.control.data['tgl_lahir']
    member.inputan_alamat.value = e.control.data['alamat']
    member.inputan_telp.value = e.control.data['telp']
    member.inputan_tgl_member.value = e.control.data['tgl_member']
    member.dialog.open = True
    member.update()

# fungsi simpan data
def simpan_member(e):
    try:
        if (member.inputan_id.value == "") :
            sql = "INSERT INTO membership (id, nama, jekel, tgl_lahir, alamat, telp, tgl_member)
VALUES(%s, %s, %s, %s, %s, %s, %s)"
            val = (member.inputan_id.value, member.inputan_nama.value,
member.inputan_jekel.value, member.inputan_tgl_lahir.value, member.inputan_alamat.value,
member.inputan_telp.value, member.inputan_tgl_member.value)
        else :
            sql = "UPDATE membership SET nama = %s, jekel = %s, tgl_lahir = %s, alamat = %s, telp
= %s, tgl_member = %s WHERE id = %s"
            val = (member.inputan_nama.value, member.inputan_jekel.value,
member.inputan_tgl_lahir.value, member.inputan_alamat.value, member.inputan_telp.value,
member.inputan_tgl_member.value, member.inputan_id.value)

        cursor.execute(sql, val)
        koneksi_db.commit()
        print(cursor.rowcount, "Data di simpan!")

        tampil_member(e)
        member.dialog.open = False
        member.snack_bar_berhasil.open = True
        member.update()
    except Exception as e:
        print(e)
        print("Ada yang error!")

```

```

# fungsi hapus data
def hapus_member(e):
    try:
        sql = "DELETE FROM membership WHERE id = %s"
        val = (e.control.data['id'],)
        cursor.execute(sql, val)
        koneksi_db.commit()
        print(cursor.rowcount, "data di hapus!")
        member.data_member.rows.clear()

        tampil_member(e)
        member.dialog.open = False
        member.snack_bar_berhasil.open = True
        member.update()
    except Exception as e:
        print(e)
        print("Ada yang error!")

# menampilkan semua data ke dalam tabel
cursor.execute("SELECT * FROM membership")
result = cursor.fetchall()
columns = [column[0] for column in cursor.description]
rows = [dict(zip(columns,row)) for row in result]
member.data_member = DataTable(
    columns = [
        DataColumn(Text("ID")),
        DataColumn(Text("Nama")),
        DataColumn(Text("Jenis Kelamin")),
        DataColumn(Text("Tgl Lahir")),
        DataColumn(Text("Alamat")),
        DataColumn(Text("Telp")),
        DataColumn(Text("Tgl Member")),
        DataColumn(Text("Ops")),
    ],
)
for row in rows:
    member.data_member.rows.append(
        DataRow(
            cells = [
                DataCell(Text(row['id'])),
                DataCell(Text(row['nama'])),
                DataCell(Text(row['jkel'])),
                DataCell(Text(row['tgl_lahir'])),
                DataCell(Text(row['alamat'])),
                DataCell(Text(row['telp'])),
                DataCell(Text(row['tgl_member'])),
                DataCell(
                    Row([
                        IconButton("EDIT_OUTLINED", icon_color = "grey", data = row, on_click =
tampil_dialog_ubah_member),
                        IconButton("DELETE_OUTLINE_OUTLINED", icon_color = "red", data = row,

```

```

on_click = hapus_member ),
    ])
    ),
]
)
)

# buat variabel utk layout data rekapan
member.layout_data = Column()

# buat form dialog untuk form entri data
member.dialog = BottomSheet(
    Container(
        Column(
            [
                Text("Form Entri Data Member", weight = FontWeight.BOLD),
                Row([ member.inputan_id ]),
                Row([ member.inputan_nama ]),
                Row([ member.inputan_jekel ]),
                Row([ member.inputan_alamat ]),
                Row([ member.inputan_tgl_lahir,
FloatingActionButton(icon=icons.CALENDAR_MONTH, on_click=lambda _ :
member.opsi_tanggal.pick_date()) ]),
                Row([ member.inputan_telp ]),
                Row([ member.inputan_tgl_member,
FloatingActionButton(icon=icons.CALENDAR_MONTH, on_click=lambda _ :
member.opsi_tanggal_member.pick_date()) ]),
                Row([
                    #tombol tambah data
                    ElevatedButton(
                        "Simpan Data",
                        icon = "SAVE_AS",
                        icon_color = "white",
                        color = "white",
                        bgcolor = "blue",
                        width = 250,
                        height = 50,
                        on_click = simpan_member,
                    )
                ]),
            ],
            horizontal_alignment = CrossAxisAlignment.CENTER,
            height = 500,
            scroll= ScrollMode.ALWAYS,
            #tight = True,

        ),
        padding = 40,
        width = 378,
        height = 500

```

```

    ),
    open = False,
    #on_dismiss=bs_dismissed,
)

# buat variabel tampilan layout utama
member.layout_utama = Column(
    [
        Container(
            Text(
                "Rekap Data Member",
                size = 25,
                # color = "white",
                weight = FontWeight.BOLD,
            ),
            alignment = alignment.center,
            padding = 30,
        ),
        Container(
            ElevatedButton(
                "Tambah Data",
                icon = "ADD",
                icon_color = "white",
                color = "white",
                bgcolor = "blue",
                width = 200,
                on_click = tampil_dialog_member,
            ),
            alignment = alignment.center,
            padding = 10,
        ),
        Row(
            [member.data_member], scroll=ScrollMode.ALWAYS
        ),
        member.layout_data,
        member.snack_bar_berhasil,
        member.dialog,
        member.opsi_tanggal,
        member.opsi_tanggal_member
    ]
)

return member.layout_utama

```

```

class Reservasi(UserControl):
    # class untuk halaman mata kuliah
    def build(reservasi) :

        def format_rupiah(angka):
            return f"Rp {angka:,.0f}".replace('.', ',')

```

```

# tgl_lhr reservasi
def ubah_tanggal_lhr(e):
    tgl_baru = reservasi.opsi_tanggal.value
    reservasi.inputan_tanggal_reservasi.value = tgl_baru.date()
    reservasi.update()

def opsi_tanggal_lhr_dismissed(e):
    tgl_baru = reservasi.inputan_tanggal_reservasi.value
    reservasi.inputan_tanggal_reservasi.value = tgl_baru
    reservasi.update()

reservasi.opsi_tanggal = DatePicker(
    on_change=ubah_tanggal_lhr,
    on_dismiss=opsi_tanggal_lhr_dismissed,
    first_date=datetime.datetime(2023, 10, 1),
    last_date=datetime.datetime(2024, 10, 1),
)

#input jam
def ubah_jam(e):
    jam_baru = reservasi.opsi_jam.value
    reservasi.inputan_waktu_reservasi.value = jam_baru.strftime('%H:%M')
    reservasi.update()

def ubah_jam_dismissed(e):
    jam_baru = reservasi.inputan_waktu_reservasi.value
    reservasi.inputan_waktu_reservasi.value = jam_baru
    reservasi.update()

reservasi.opsi_jam = TimePicker(
    confirm_text="Confirm",
    error_invalid_text="Time out of range",
    help_text="Pick your time slot",
    on_change=ubah_jam,
    on_dismiss=ubah_jam_dismissed,
)

# buat variabel inputan
reservasi.inputan_id_reservasi= TextField(visible = False, expand = True)

cursor.execute("SELECT * FROM membership")
reservasi.inputan_id_pelanggan = Dropdown(label = "Nama Pelanggan", hint_text = "Nama Pelanggan", expand = True, options=[dropdown.Option(row[0],row[1] + " - " + row[2]) for row in cursor.fetchall()])
reservasi.inputan_tanggal_reservasi = TextField(label = "Tanggal Reservasi", hint_text = "Tanggal Reservasi", expand = True)
reservasi.inputan_waktu_reservasi = TextField(label = "Waktu Reservasi", hint_text = "Waktu Reservasi", expand = True)

cursor.execute("SELECT * FROM layanan")

```

```

    reservasi.inputan_jenis_layanan= Dropdown(label = "Jenis Layanan", hint_text = "Jenis",
expand = True, options=[dropdown.Option(row[0], f"{row[1]} - {row[2]}") for row in
cursor.fetchall()],)
    reservasi.snack_bar_berhasil = SnackBar( Text("Operasi berhasil"), bgcolor="green")

# memuat tabel data
def tampil_reservasi(e):
    # Merefresh halaman & menampilkan notif
    reservasi.data_reservasi.rows.clear()
    cursor.execute("""
        SELECT reservasi.id_reservasi, reservasi.id_pelanggan, membership.nama AS
nama_pelanggan,
        reservasi.tanggal_reservasi, reservasi.waktu_reservasi,
        reservasi.jns_layanan, layanan.jns_layanan AS nama_layanan,
        layanan.hrg_layanan AS harga_layanan
        FROM reservasi
        JOIN membership ON reservasi.id_pelanggan = membership.id
        JOIN layanan ON reservasi.jns_layanan = layanan.id_layanan
        """)
    result = cursor.fetchall()
    # menampilkan ulang data
    columns = [column[0] for column in cursor.description]
    rows = [dict(zip(columns,row)) for row in result]
    for row in rows:
        reservasi.data_reservasi.rows.append(
            DataRow(
                cells = [
                    DataCell(Text(row['id_reservasi'])),
                    DataCell(Text(f"{row['id_pelanggan']} - ({row['nama_pelanggan']})")),
                    DataCell(Text(row['tanggal_reservasi'])),
                    DataCell(Text(row['waktu_reservasi'])),
                    DataCell(Text(f"{row['jns_layanan']} - ({row['nama_layanan']})")),
                    DataCell(Text(format_rupiah(row['harga_layanan']))),
                    DataCell(
                        Row([
                            IconButton("EDIT_OUTLINED", icon_color = "grey", data = row, on_click =
tampil_dialog_ubah_reservasi ),
                            IconButton("DELETE_OUTLINE_OUTLINED", icon_color = "red", data = row,
on_click = hapus_reservasi ),
                        ])),
                ]
            )
        )

# fungsi menampilkan dialog form entri
def tampil_dialog_reservasi(e):
    reservasi.inputan_id_reservasi.value = ""
    reservasi.inputan_id_pelanggan.value = ""
    reservasi.inputan_tanggal_reservasi.value = ""
    reservasi.inputan_waktu_reservasi.value = ""
    reservasi.inputan_jenis_layanan.value = ""

```

```

reservasi.dialog.open = True
reservasi.update()

def tampil_dialog_ubah_reservasi(e):
    reservasi.inputan_id_reservasi.value = e.control.data['id_reservasi']
    reservasi.inputan_id_pelanggan.value = e.control.data['id_pelanggan']
    reservasi.inputan_tanggal_reservasi.value = e.control.data['tanggal_reservasi']
    reservasi.inputan_waktu_reservasi.value = e.control.data['waktu_reservasi']
    reservasi.inputan_jenis_layanan.value = e.control.data['jns_layanan']
    reservasi.dialog.open = True
    reservasi.update()

# fungsi simpan data
def simpan_reservasi(e):
    try:
        if (reservasi.inputan_id_reservasi.value == '') :
            sql = "INSERT INTO reservasi (id_reservasi, id_pelanggan, tanggal_reservasi,
waktu_reservasi, jns_layanan) VALUES(%s, %s, %s, %s, %s)"
            val = (reservasi.inputan_id_reservasi.value, reservasi.inputan_id_pelanggan.value,
reservasi.inputan_tanggal_reservasi.value, reservasi.inputan_waktu_reservasi.value,
reservasi.inputan_jenis_layanan.value)
        else :
            sql = "UPDATE reservasi SET id_pelanggan = %s, tanggal_reservasi = %s,
waktu_reservasi = %s, jns_layanan = %s WHERE id_reservasi = %s"
            val = (reservasi.inputan_id_pelanggan.value,
reservasi.inputan_tanggal_reservasi.value, reservasi.inputan_waktu_reservasi.value,
reservasi.inputan_jenis_layanan.value, reservasi.inputan_id_reservasi.value)

        cursor.execute(sql, val)
        koneksi_db.commit()
        print(cursor.rowcount, "Data di simpan!")

        tampil_reservasi(e)
        reservasi.dialog.open = False
        reservasi.snack_bar_berhasil.open = True
        reservasi.update()
    except Exception as e:
        print(e)
        print("Ada yang error!")

# fungsi hapus data
def hapus_reservasi(e):
    try:
        sql = "DELETE FROM reservasi WHERE id_reservasi = %s"
        val = (e.control.data['id_reservasi'],)
        cursor.execute(sql, val)
        koneksi_db.commit()
        print(cursor.rowcount, "data di hapus!")
        reservasi.data_reservasi.rows.clear()

```

```

        tampil_reservasi(e)
        reservasi.dialog.open = False
        reservasi.snack_bar_berhasil.open = True
        reservasi.update()
    except Exception as e:
        print(e)
        print("Ada yang error!")

# menampilkan semua data ke dalam tabel
cursor.execute("""
    SELECT reservasi.id_reservasi, reservasi.id_pelanggan, membership.nama AS
nama_pelanggan,
        reservasi.tanggal_reservasi, reservasi.waktu_reservasi,
        reservasi.jns_layanan, layanan.jns_layanan AS nama_layanan,
        layanan.hrg_layanan AS harga_layanan
    FROM reservasi
    JOIN membership ON reservasi.id_pelanggan = membership.id
    JOIN layanan ON reservasi.jns_layanan = layanan.id_layanan
""")
result = cursor.fetchall()
columns = [column[0] for column in cursor.description]
rows = [dict(zip(columns,row)) for row in result]
reservasi.data_reservasi = DataTable(
    columns = [
        DataColumn(Text("ID Reservasi")),
        DataColumn(Text("ID Pelanggan")),
        DataColumn(Text("Tanggal Reservasi")),
        DataColumn(Text("Waktu Reservasi")),
        DataColumn(Text("Jenis Layanan")),
        DataColumn(Text("Total Harga")),
        DataColumn(Text("Opsi")),
    ],
)
for row in rows:
    reservasi.data_reservasi.rows.append(
        DataRow(
            cells = [
                DataCell(Text(row['id_reservasi'])),
                DataCell(Text(f"{row['id_pelanggan']} - ({row['nama_pelanggan']})")),
                DataCell(Text(row['tanggal_reservasi'])),
                DataCell(Text(row['waktu_reservasi'])),
                DataCell(Text(f"{row['jns_layanan']} - ({row['nama_layanan']})")),
                DataCell(Text(format_rupiah(row['harga_layanan']))),
                DataCell(
                    Row([
                        IconButton("EDIT_OUTLINED", icon_color = "grey", data = row, on_click =
tampil_dialog_ubah_reservasi ),
                        IconButton("DELETE_OUTLINE_OUTLINED", icon_color = "red", data = row,
on_click = hapus_reservasi ),
                    ])
                )
            ]
        )
    )

```

```

        ),
    ],
)
)

# buat variabel utk layout data rekapan
reservasi.layout_data = Column()

# buat form dialog untuk form entri data
reservasi.dialog = BottomSheet(
    Container(
        Column(
            [
                Text("Form Entri Data Reservasi", weight = FontWeight.BOLD),
                Row([ reservasi.inputan_id_reservasi ]),
                Row([ reservasi.inputan_id_pelanggan ]),
                Row([ reservasi.inputan_tanggal_reservasi,
FloatingActionButton(icon=icons.CALENDAR_MONTH, on_click=lambda _:
reservasi.opsi_tanggal.pick_date()) ]),
                Row([
reservasi.inputan_waktu_reservasi,FloatingActionButton(icon=icons.ACCESS_TIME,
on_click=lambda _: reservasi.opsi_jam.pick_time()) ]),
                Row([ reservasi.inputan_jenis_layanan ]),
                Row([
                    #tombol tambah data
                    ElevatedButton(
                        "Simpan Data",
                        icon = "SAVE_AS",
                        icon_color = "white",
                        color = "white",
                        bgcolor = "blue",
                        width = 250,
                        height = 50,
                        on_click = simpan_reservasi,
                    )
                ]),
            ],
            horizontal_alignment = CrossAxisAlignment.CENTER,
            height = 500,
            scroll= ScrollMode.ALWAYS,
            #tight = True,

        ),
        padding = 40,
        width = 378,
        height = 500
    ),
    open = False,
    #on_dismiss=bs_dismissed,

```

)

```
# buat variabel tampilan layout utama
reservasi.layout_utama = Column(
    [
        Container(
            Text(
                "Rekap Data Reservasi",
                size = 23,
                # color = "white",
                weight = FontWeight.BOLD,
            ),
            alignment = alignment.center,
            padding = 30,
        ),
        Container(
            ElevatedButton(
                "Tambah Data",
                icon = "ADD",
                icon_color = "white",
                color = "white",
                bgcolor = "blue",
                width = 200,
                on_click = tampil_dialog_reservasi,
            ),
            alignment = alignment.center,
            padding = 10,
        ),

        Row(
            [reservasi.data_reservasi], scroll=ScrollMode.ALWAYS
        ),
        reservasi.layout_data,
        reservasi.opsi_tanggal,
        reservasi.opsi_jam,
        reservasi.snack_bar_berhasil,
        reservasi.dialog,
    ]
)

return reservasi.layout_utama
```

```
class Layanan(UserControl):
    # class untuk halaman mata kuliah
    def build(layanan) :

        def format_rupiah(angka):
            return f"Rp {angka:,.0f}".replace(',', '')
```

```

# buat variabel inputan
layanannya.inputan_id_layanan= TextField(visible = False, expand = True)
layanannya.inputan_jenis_layanan= TextField(label = "Jenis Layanan", hint_text = "Jenis", expand
= True )
layanannya.inputan_harga_layanan = TextField(label = "Harga Layanan", hint_text = "Harga
Layanan", expand = True)
layanannya.snack_bar_berhasil = Snackbar( Text("Operasi berhasil"), bgcolor="green")

# memuat tabel data
def tampil_layanan(e):
    # Merefresh halaman & menampilkan notif
    layanannya.data_layanan.rows.clear()
    cursor.execute("SELECT * FROM layanannya")
    result = cursor.fetchall()
    # menampilkan ulang data
    columns = [column[0] for column in cursor.description]
    rows = [dict(zip(columns,row)) for row in result]
    for row in rows:
        layanannya.data_layanan.rows.append(
            DataRow(
                cells = [
                    DataCell(Text(row['id_layanan'])),
                    DataCell(Text(row['jns_layanan'])),
                    DataCell(Text(format_rupiah(row['hrg_layanan']))),
                    DataCell(
                        Row([
                            IconButton("EDIT_OUTLINED", icon_color = "grey", data = row,
on_click=tampil_dialog_ubah_layanan ),
                            IconButton("DELETE_OUTLINE_OUTLINED", icon_color = "red", data = row,
on_click=hapus_layanan ),
                        ])
                    ),
                ]
            )
        )

# fungsi menampilkan dialog form entri
def tampil_dialog_layanan(e):
    layanannya.inputan_id_layanan.value = ""
    layanannya.inputan_jenis_layanan.value = ""
    layanannya.inputan_harga_layanan.value = ""
    layanannya.dialog.open = True
    layanannya.update()

def tampil_dialog_ubah_layanan(e):
    layanannya.inputan_id_layanan.value = e.control.data['id_layanan']
    layanannya.inputan_jenis_layanan.value = e.control.data['jns_layanan']
    layanannya.inputan_harga_layanan.value = e.control.data['hrg_layanan']
    layanannya.dialog.open = True
    layanannya.update()

```

```

# fungsi simpan data
def simpan_layanan(e):
    try:
        if (layanan.inputan_id_layanan.value == '') :
            sql = "INSERT INTO layanan (id_layanan, jns_layanan, hrg_layanan ) VALUES(%s, %s,
%s)"
            val = (layanan.inputan_id_layanan.value, layanan.inputan_jenis_layanan.value,
layanan.inputan_harga_layanan.value)
        else :
            sql = "UPDATE layanan SET jns_layanan = %s, hrg_layanan = %s WHERE id_layanan =
%s"
            val = (layanan.inputan_jenis_layanan.value, layanan.inputan_harga_layanan.value,
layanan.inputan_id_layanan.value)

        cursor.execute(sql, val)
        koneksi_db.commit()
        print(cursor.rowcount, "Data di simpan!")

        tampil_layanan(e)
        layanan.dialog.open = False
        layanan.snack_bar_berhasil.open = True
        layanan.update()
    except Exception as e:
        print(e)
        print("Ada yang error!")

# fungsi hapus data
def hapus_layanan(e):
    try:
        sql = "DELETE FROM layanan WHERE id_layanan = %s"
        val = (e.control.data['id_layanan'],)
        cursor.execute(sql, val)
        koneksi_db.commit()
        print(cursor.rowcount, "data di hapus!")
        layanan.data_layanan.rows.clear()

        tampil_layanan(e)
        layanan.dialog.open = False
        layanan.snack_bar_berhasil.open = True
        layanan.update()
    except Exception as e:
        print(e)
        print("Ada yang error!")

# menampilkan semua data ke dalam tabel
cursor.execute("SELECT * FROM layanan")
result = cursor.fetchall()
columns = [column[0] for column in cursor.description]
rows = [dict(zip(columns,row)) for row in result]
layanan.data_layanan = DataTable(

```

```

        columns = [
            DataColumn(Text("ID Layanan")),
            DataColumn(Text("Jenis Layanan")),
            DataColumn(Text("Harga Layanan")),
            DataColumn(Text("Opsi")),
        ],
    )
    for row in rows:
        layanan.data_layanan.rows.append(
            DataRow(
                cells = [
                    DataCell(Text(row['id_layanan'])),
                    DataCell(Text(row['jns_layanan'])),
                    DataCell(Text(format_rupiah(row['hrg_layanan']))),
                    DataCell(
                        Row([
                            IconButton("EDIT_OUTLINED", icon_color = "grey", data = row, on_click =
tampil_dialog_ubah_layanan),
                            IconButton("DELETE_OUTLINE_OUTLINED", icon_color = "red", data = row,
on_click = hapus_layanan ),
                        ])
                    ),
                ]
            )
        )

# buat variabel utk layout data rekapan
layanan.layout_data = Column()

# buat form dialog untuk form entri data
layanan.dialog = BottomSheet(
    Container(
        Column(
            [
                Text("Form Entri Data Layanan", weight = FontWeight.BOLD),
                Row([ layanan.inputan_id_layanan ]),
                Row([ layanan.inputan_jenis_layanan ]),
                Row([ layanan.inputan_harga_layanan ]),
                Row([
                    #tombol tambah data
                    ElevatedButton(
                        "Simpan Data",
                        icon = "SAVE_AS",
                        icon_color = "white",
                        color = "white",
                        bgcolor = "blue",
                        width = 250,
                        height = 50,
                        on_click = simpan_layanan,
                    )
                ])
            ]
        ),
    ),
)

```

```

        ],
        horizontal_alignment = CrossAxisAlignment.CENTER,
        height = 500,
        scroll= ScrollMode.ALWAYS,
        #tight = True,

    ),
    padding = 40,
    width = 378,
    height = 500
),
open = False,
#on_dismiss=bs_dismissed,
)

# buat variabel tampilan layout utama
layanan.layout_utama = Column(
[
    Container(
        Text(
            "Rekap Data Layanan",
            size = 25,
            # color = "White",
            weight = FontWeight.BOLD,
        ),
        alignment = alignment.center,
        padding = 30,
    ),
    Container(
        ElevatedButton(
            "Tambah Data",
            icon = "ADD",
            icon_color = "white",
            color = "white",
            bgcolor = "blue",
            width = 200,
            on_click = tampil_dialog_layanan,
        ),
        alignment = alignment.center,
        padding = 10,
    ),
    Row(
        [layanan.data_layanan], scroll=ScrollMode.ALWAYS
    ),
    layanan.layout_data,
    layanan.snack_bar_berhasil,
    layanan.dialog,
]
)

```

```
return layanan.layout_utama
```

```
# fungsi utama
def main (page : Page):
    # mengatur halaman
    page.title = "Fattah Barbershop"
    page.window_width = 350
    page.window_height = 700
    page.window_resizable = False
    page.window_maximizable = False
    page.window_minimizable = True
    page.scroll = "adaptive"
    #page.theme_mode = "light"
    page.theme_mode = ThemeMode.DARK

# fungsi untuk mode halaman dark/light
def mode_tema(e):
    page.theme_mode = "dark" if page.theme_mode == "light" else "light"
    page.update()

# fungsi untuk routing / pembagian halaman
def route_change(route):
    page.views.clear()
    page.views.append(
        View("/",
            [
                AppBar(
                    title = Text("Aplikasi CRUD Fattah Barbershop", size = 18, weight = FontWeight.BOLD,
color = colors.WHITE),
                    bgcolor = colors.BLUE_800,
                    center_title = True,
                ),
                Column(
                    [
                        Image(src="reference/fattahbarbershop.png",width=180 ),
                        # Icon(name = icons.CAST_FOR_EDUCATION, color = colors.BLUE, size = 180),
                        Column(
                            controls = [
                                ElevatedButton("Menu Reservasi", icon = icons.TABLE_ROWS, on_click =
lambda _: page.go("/reservasi"), width=205),
                                ElevatedButton("Menu Membership", icon = icons.PEOPLE_ROUNDED,
on_click = lambda _: page.go("/member"), width=205 ),
                                ElevatedButton("Menu Layanan", icon = icons.PEOPLE_ROUNDED, on_click =
lambda _: page.go("/layanan"), width=205 ),
                                ElevatedButton("Menu Jadwal Kuliah", icon = icons.SCHEDULE_ROUNDED,
on_click = lambda _: page.go("/jadwalkuliah"), disabled=True, width=205, visible=False ),
                            ],
                            width = 375,
                            horizontal_alignment = CrossAxisAlignment.CENTER,
```

```

        ),
        ],
        height = 500,
        width = 375,
        alignment = MainAxisAlignment.SPACE_AROUND,
        horizontal_alignment = CrossAxisAlignment.CENTER,
    ),
    Column(
      [
        Column(
          controls = [
            ElevatedButton("Mode Warna", icon = icons.WB_SUNNY_OUTLINED, on_click
= mode_tema),
          ],
          width = 375,
          horizontal_alignment = CrossAxisAlignment.CENTER,
        ),
        Text('Mobile Programming - Fattah Barbershop @2024', size = 12)
      ],
      horizontal_alignment = CrossAxisAlignment.CENTER,
    ),
  ],
)
)
if page.route == "/member":
  page.views.append(
    View("/member",
      [
        AppBar(title = Text("Menu Membership", size = 14, weight = FontWeight.BOLD),
bgcolor = colors.SURFACE_VARIANT ),
        FormMember()
      ],
    )
  )
if page.route == "/reservasi":
  page.views.append(
    View("/reservasi",
      [
        AppBar(title = Text("Menu Reservasi", size = 14, weight = FontWeight.BOLD), bgcolor
= colors.SURFACE_VARIANT ),
        Reservasi()
      ],
    )
  )
if page.route == "/layanan":
  page.views.append(
    View("/layanan",
      [
        AppBar(title = Text("Menu Layanan", size = 14, weight = FontWeight.BOLD), bgcolor =

```

```

colors.SURFACE_VARIANT),
    Layanan()
  ],
)
)

page.update()

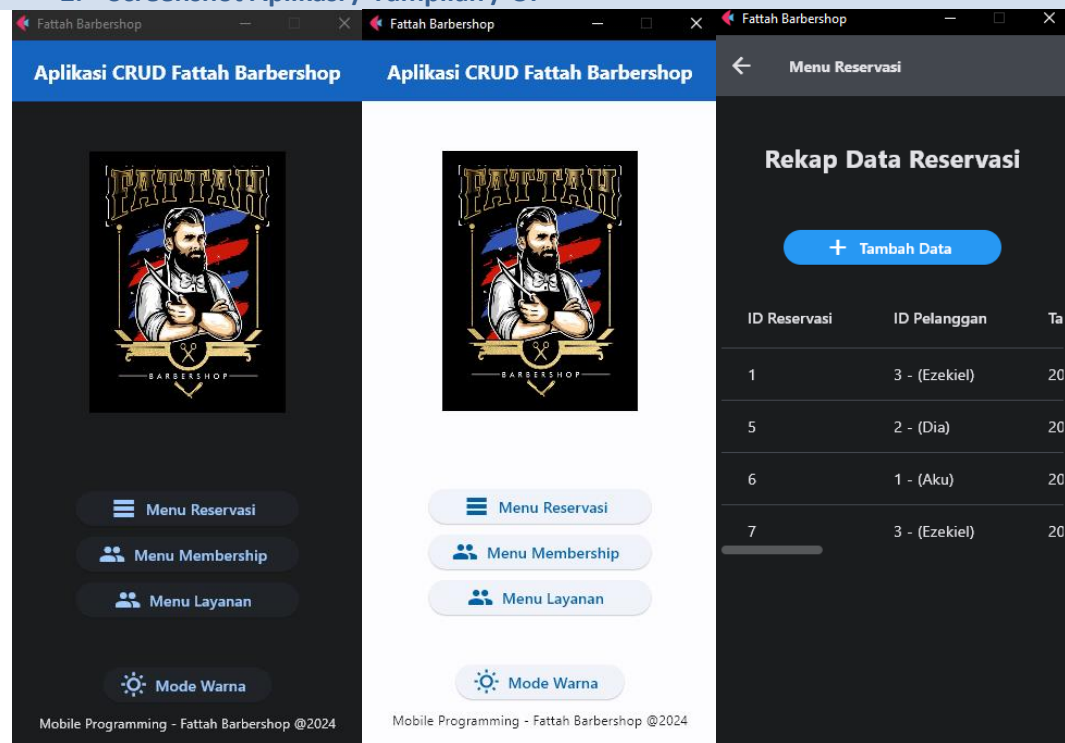
# fungsi untuk pop up halaman
def view_pop(view):
    page.views.pop()
    top_view = page.views[-1]
    page.go(top_view.route)

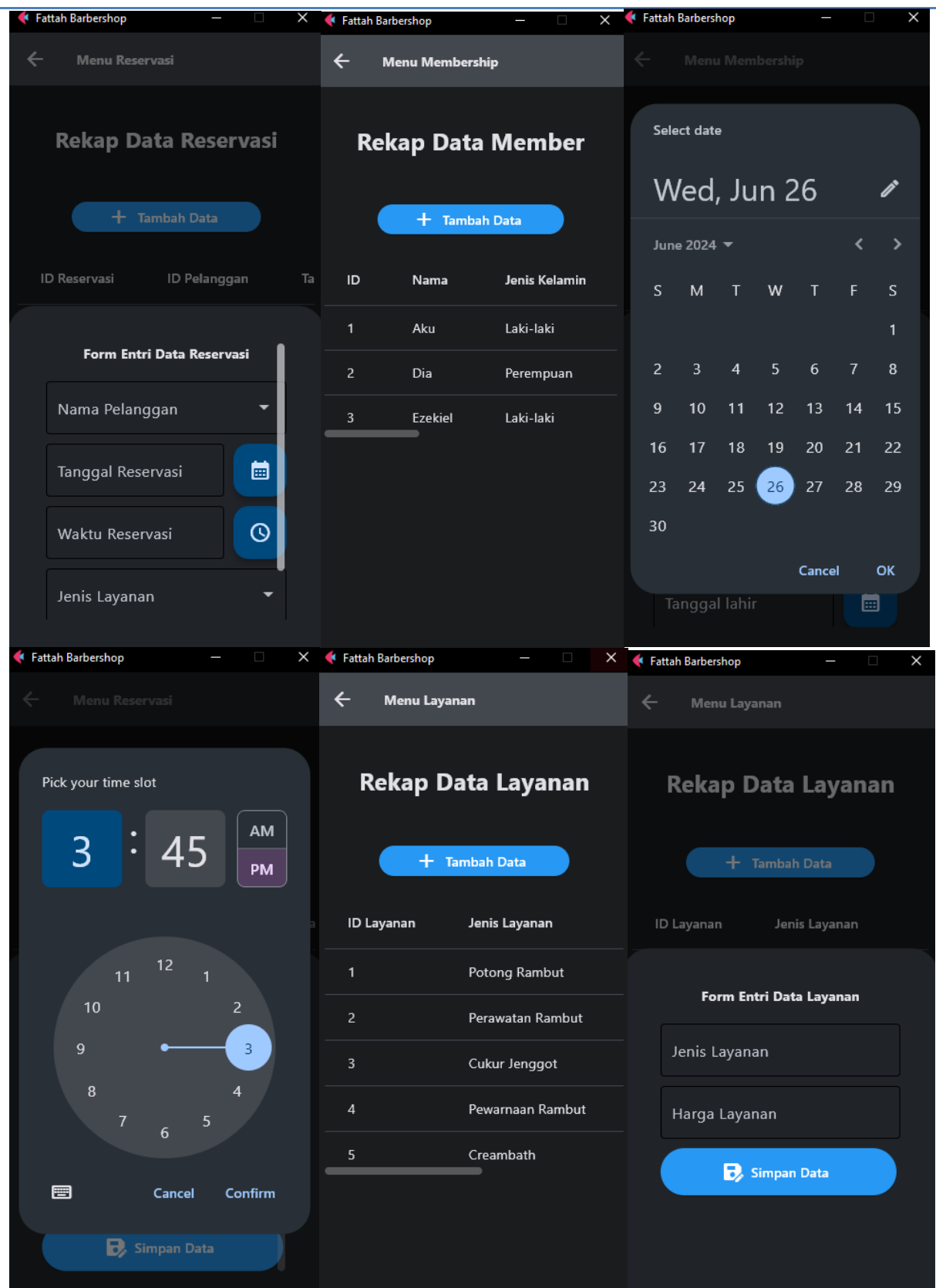
page.on_route_change = route_change
page.on_view_pop = view_pop
page.go(page.route)

# mengatur output aplikasi
flet.app(target = main)

```

2. Screenshot Aplikasi / Tampilan / UI





3. Screenshot Relasi Tabel

fattah_crud_mobile layanan	
id_layanan	int(12)
jns_layanan	varchar(255)
hrg_layanan	int(255)

fattah_crud_mobile membership	
id	int(11)
nama	varchar(50)
jenis	varchar(255)
tgl_lahir	varchar(50)
alamat	varchar(50)
telp	varchar(100)
tgl_member	varchar(50)

fattah_crud_mobile reservasi	
id_reservasi	int(10)
id_pelanggan	int(11)
tanggal_reservasi	date
waktu_reservasi	varchar(222)
jns_layanan	varchar(120)