|  |  |
| --- | --- |
| **LAPORAN PRAKTIK PEMROGRAMAN**  **UJIAN AKHIR SEMESTER**  **(Aplikasi CRUD Fattah Barbershop)** | Mobile Programming, 2024  NIM.  (2021 53 032)  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("Opsi")),  ],  )  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 ),  ])  ),  ]  )  )  # 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  layanan.inputan\_id\_layanan= TextField(visible = False, expand = True)  layanan.inputan\_jenis\_layanan= TextField(label = "Jenis Layanan", hint\_text = "Jenis", expand = True )  layanan.inputan\_harga\_layanan = TextField(label = "Harga Layanan", hint\_text = "Harga Layanan", expand = True)  layanan.snack\_bar\_berhasil = SnackBar( Text("Operasi berhasil"), bgcolor="green")  # memuat tabel data  def tampil\_layanan(e):  # Merefresh halaman & menampilkan notif  layanan.data\_layanan.rows.clear()  cursor.execute("SELECT \* FROM 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:  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 ),  ])  ),  ]  )  )  # fungsi menampilkan dialog form entri  def tampil\_dialog\_layanan(e):  layanan.inputan\_id\_layanan.value = ''  layanan.inputan\_jenis\_layanan.value = ''  layanan.inputan\_harga\_layanan.value = ''  layanan.dialog.open = True  layanan.update()  def tampil\_dialog\_ubah\_layanan(e):  layanan.inputan\_id\_layanan.value = e.control.data['id\_layanan']  layanan.inputan\_jenis\_layanan.value = e.control.data['jns\_layanan']  layanan.inputan\_harga\_layanan.value = e.control.data['hrg\_layanan']  layanan.dialog.open = True  layanan.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) |
| 1. Screenshot Aplikasi / Tampilan / UI |
|  |
| 1. Screenshot Relasi Tabel |
|  |