**LAPORAN PROJECT AKHIR PEMROGRAMAN DEKSTOP**

**Sistem Informasi Koperasi Simpan Pinjam**

Disusun untuk Memenuhi Ujian Akhir Mata Kuliah Pemrograman Dekstop

****

Dosen Pengampu:

Moh. Kautsar Sophan, S.Kom., M. MT

Disusun Oleh :

Ach. Faisol S. Arifin (180411100073)

**JURUSAN TEKNIK INFORMATIKA**

**FAKULTAS TEKNIK**

**UNIVERSITAS TRUNOJOYO MADURA**

**2019**

* 1. **Latar Belakang**

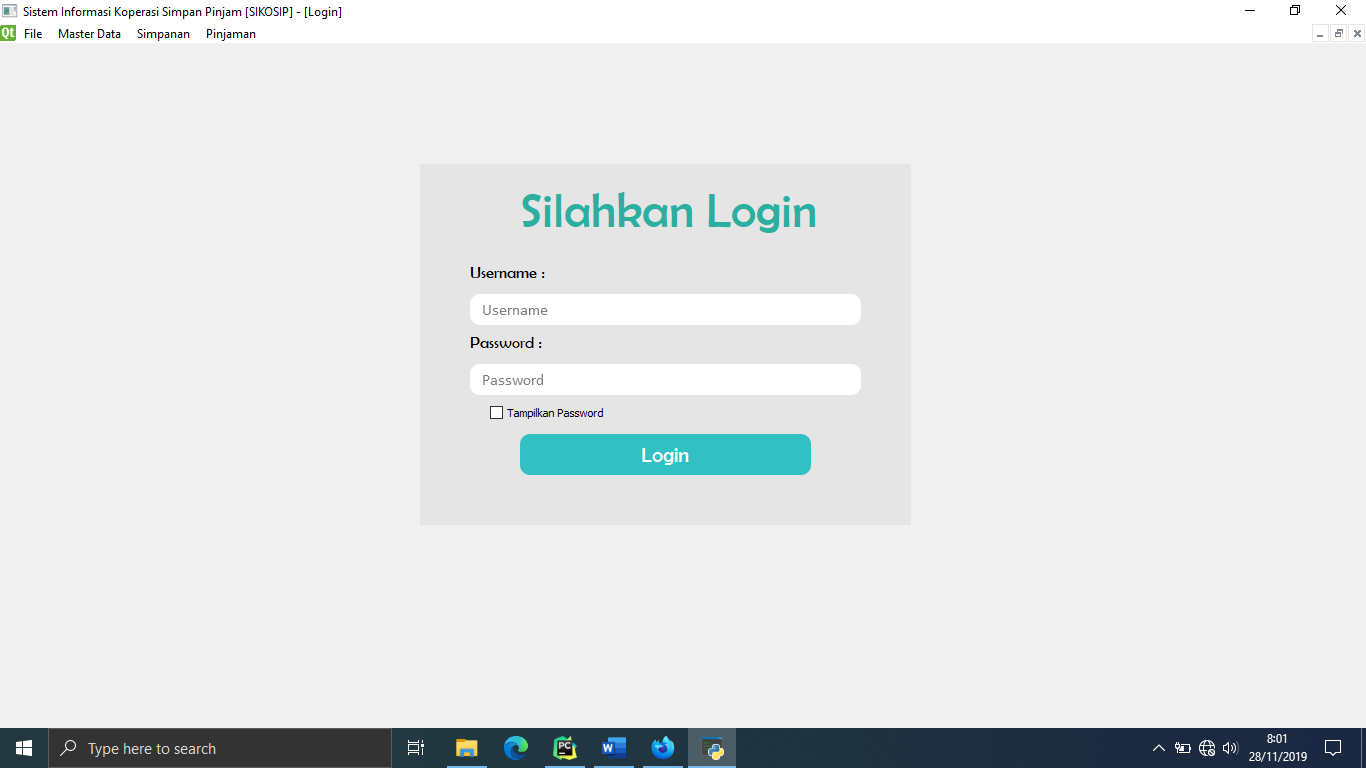
Salah satu badan usaha yang dimiliki dan dijalankan oleh anggotanya adalah koperasi. Koperasi dibentuk untuk memenuhi aspirasi dan kebutuhan bersama baik dibidang ekonomi, sosial, dan budaya. Setiap anggota koperasi akan melakukan transaksi baik menyimpanan atau peminjaman uang yang akan dilayani oleh petugas koperasi yang sedang bertugas. Petugas akan melayani para anggotanya yang kemudian akan mencatat transaksi yang dilakukan. Terkadang proses pencatatan disebuah koperasi masih di bukukan secara manual.

Dalam pencatatan transaksi yang masih dilakukan secara konvensional oleh petugas koperasi, memungkinkan terjadinya kehilangan data, atau kesalahan dalam pencatatan transaksi di sebuah koperasi. Hal tersebut dapat membuat pekerjaan menjadi tidak efisien baik dari segi waktu maupun tenaga. Dari permasalahan tersebut perlu untuk dibentuk sebuah program yang dapat melakukan pencatatan dan perhitungan secara digital yang menggunakan perangkat komputer untuk melakukan manajemen data yang kemudian disimpan pada penyimpanan komputer. Sehingga dapat mengefiensi pekerjaan para petugas koperasi dalam melayani transaksi simpanan/pinjaman yang dilakukan oleh para anggota koperasi.

* 1. **Penjelasan Umum Program**

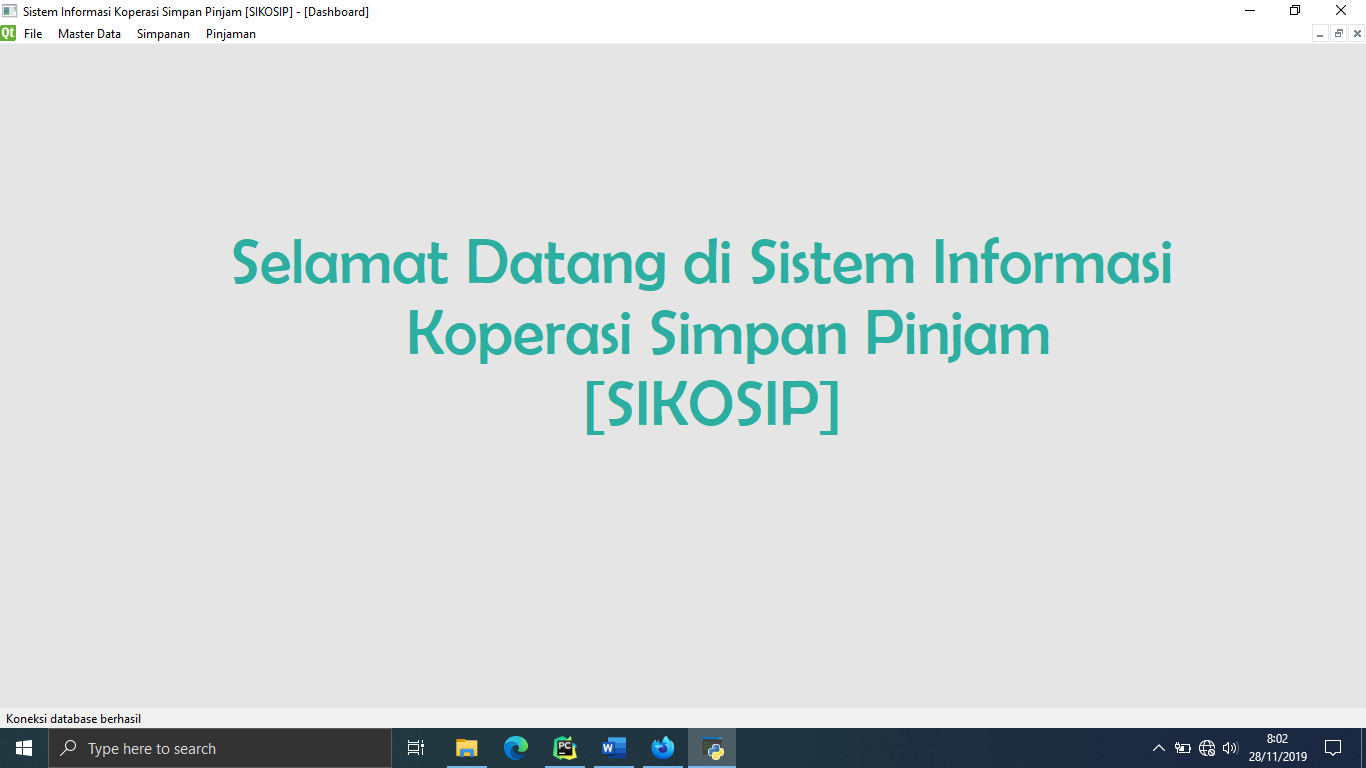
Pembentukan program untuk menyesaikan permasalah yang ter-urai diatas, progam dapat menyimpan data-data anggota berdasarkan KTP, transaksi simpanan, transaksi penarikan, pinjaman, angsuran. dan detailnya program ini berisi beberapa menu, yaitu:

1. Data anggota, dimana setiap anggota anggota yang akan berkontrobusi dengan koperasi perlu untuk mendaftarkan dirinya, yang kemudian petugas akan mencatat identitas sesuai dengan KTP dari setiap anggota yang akan bergabung.
2. Data petugas, seluruh data petugas dapat disimpan. Petugas yang akan menggunakan aplikasi tersebut, perlu untuk login terlebih dahulu agar tidak ada intevensi dari pihak yang tidak berkepentingan. Dimana dalam data petugas telah meliputi akun yang berisi, username dan password.
3. Data Simpanan, dimana menu tersebut dapat digunakan untuk mencatat transaksi simpanan dengan memasukkan NIK sebagai kode dari anggota, jenis simpanan, dan jumlah simpanan yang diberikan.
4. Penarikan, dimana menu tersebut untuk mencatat tansaksi penarikan masing-masing anggota.
5. Daftar Simpanan, dalam menu tersebut berisi jumlah simpanan yang dimiliki setiap anggota.
6. Pinjaman, menu tersebut untuk mencatat pinjaman uang setiap anggota.
7. Anguran, di menu tersebut untuk mencatat angsuran peminjaman dari setiap anggota.
   1. **Tampilan Aplikasi**
8. Tampilan Halaman Login



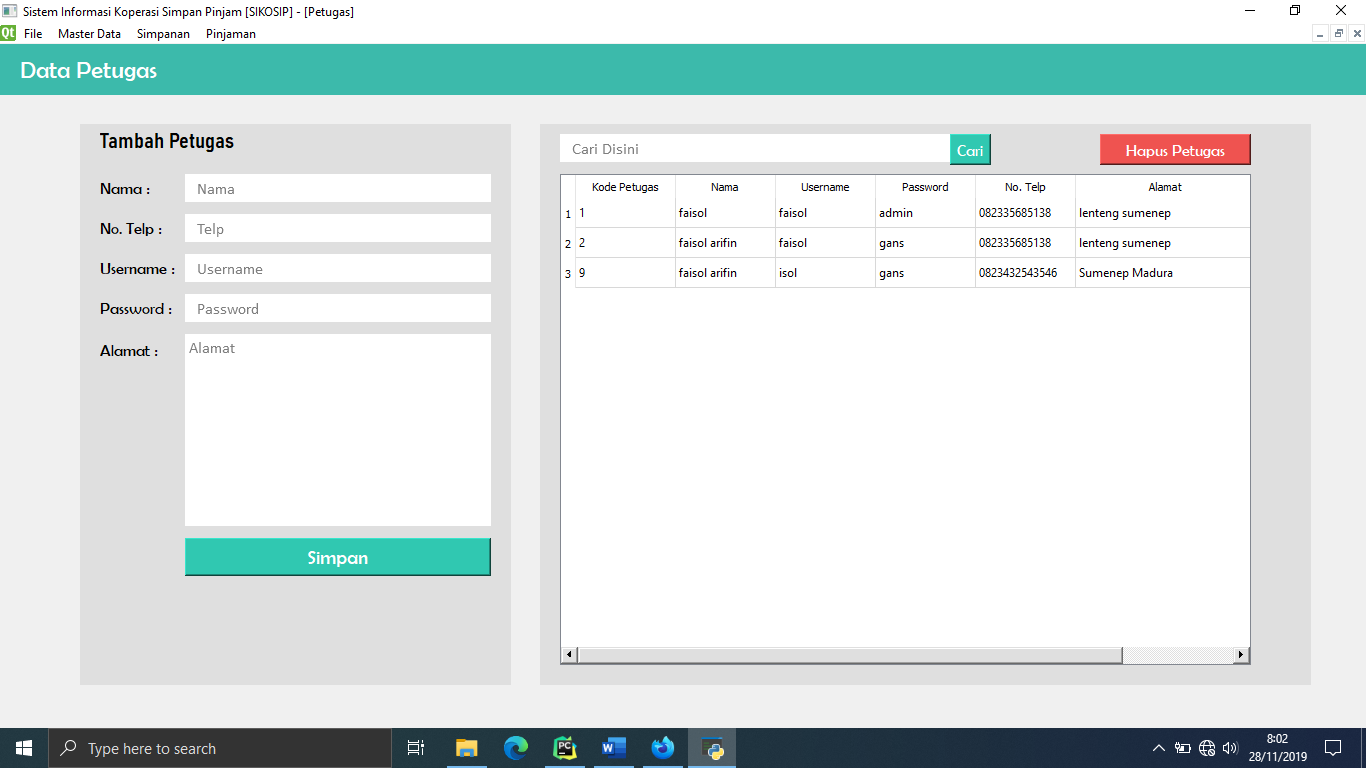
Halaman login tersebut merupakan halaman yang pertama kali akan disugukan, ketika aplikasi tersebut dijalankan. Setiap petugas yang akan menggunakan aplikasi harus melakukan autentikasi terlebih dahulu agar mengurangi intervensi dari pihak yang tidak berkepentingan dalam sistem ini.

1. Tampilan Halaman Utama.



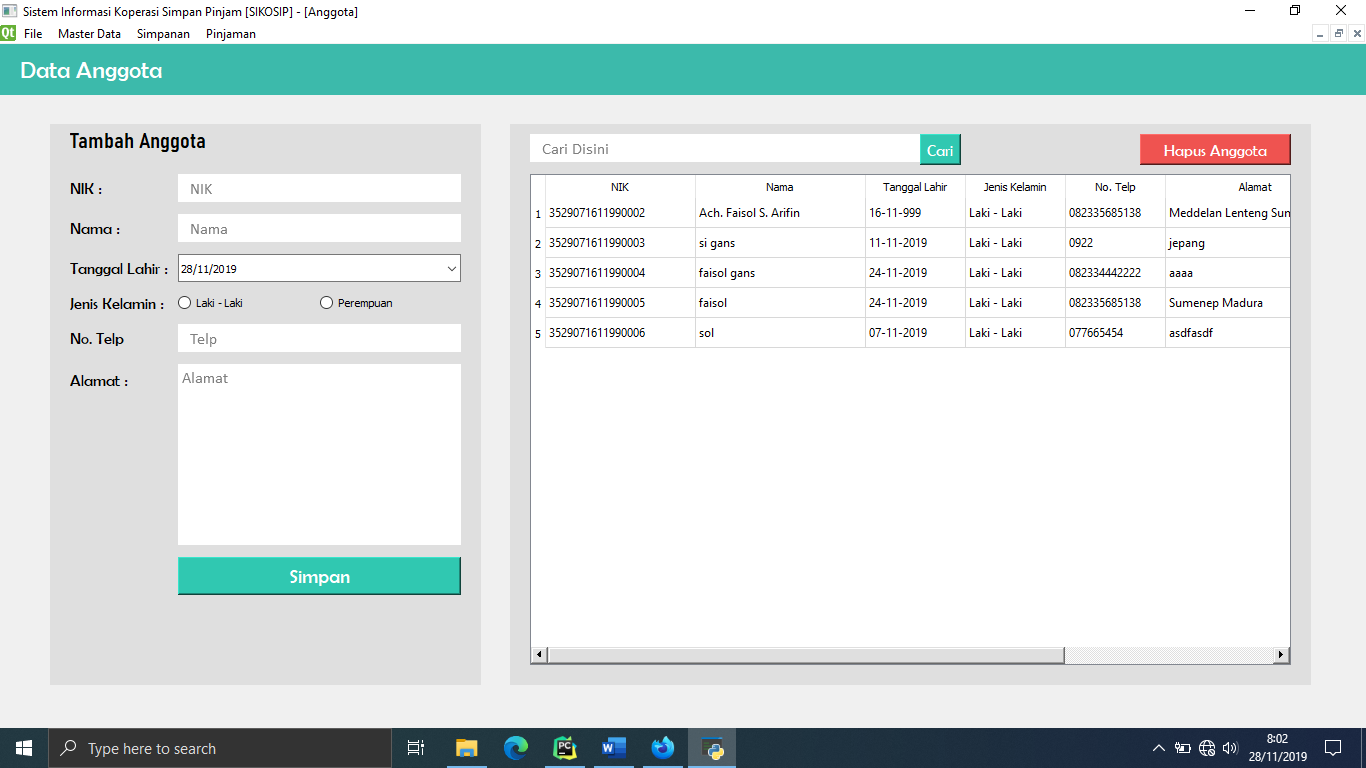
Saat authentikasi berhasil, kemudia akan dilanjutkan tampilan tersebut sebagai tampilan utama untuk menyambut user yang berhasil melakukan login.

1. Tampilan Data petugas



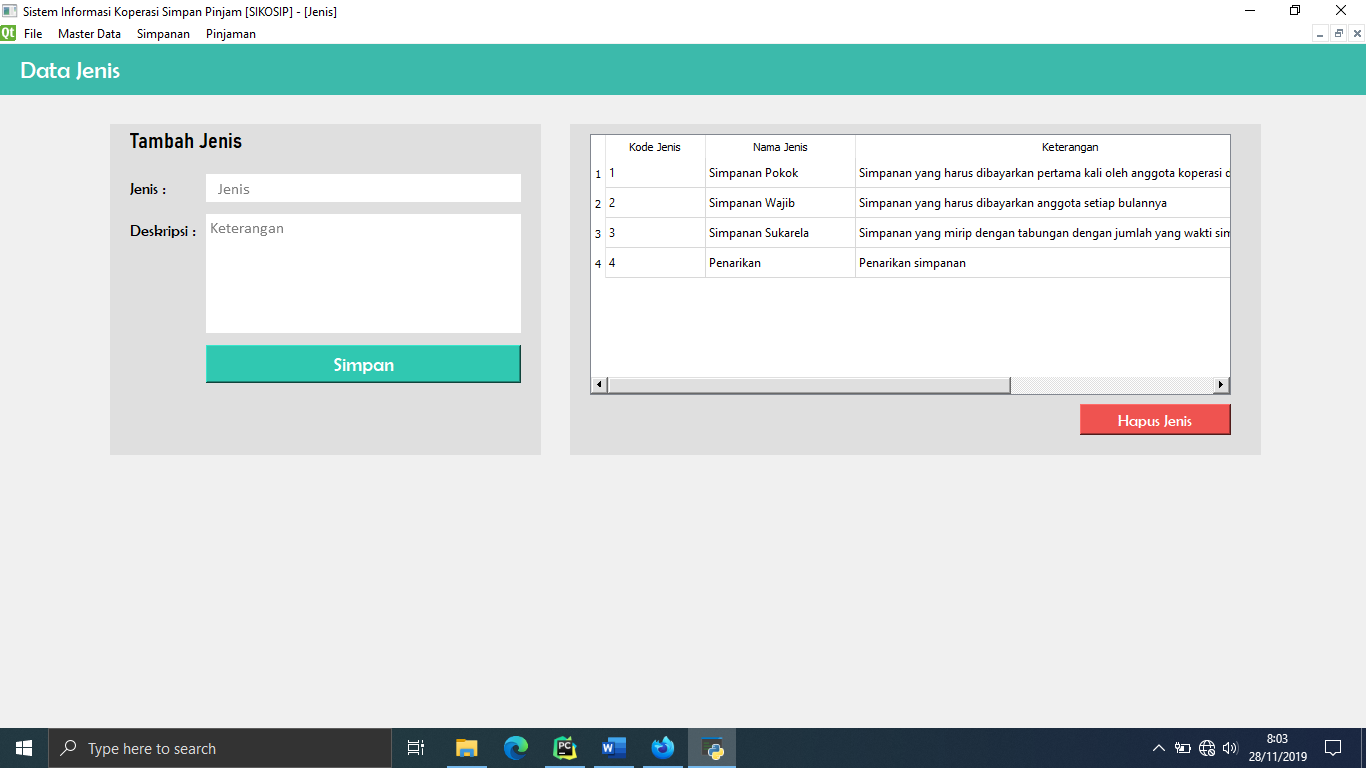
Tampilan data petugas merupakan sub menu dari master data, dimana dalam tampilan tersebut dapat melakukan tambah, pencarian, dan hapus data petugas. Tampilan tersebut akan menyinmpan dan menampilkan data petugas termasuk akun username dan password login.

1. Tampilan Halaman Data Anggota



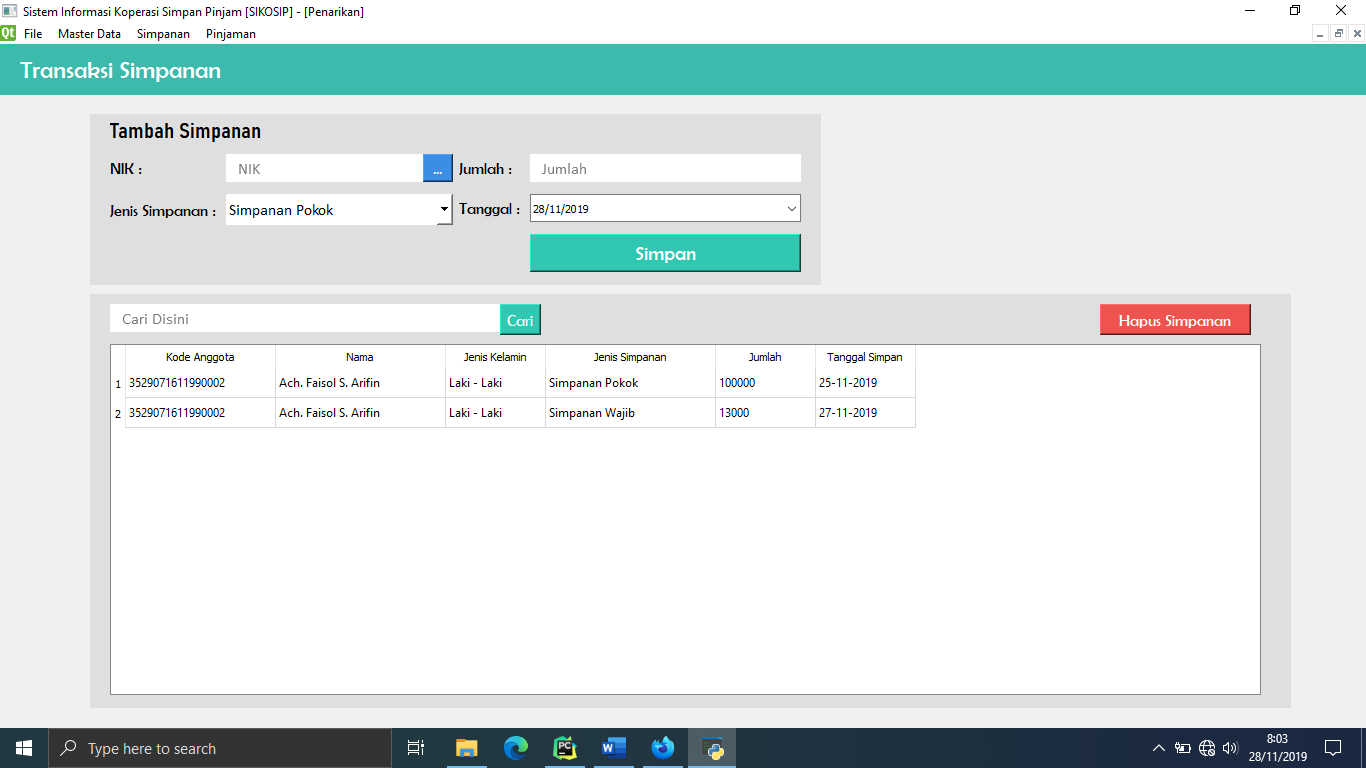
Tampilan data anggota merupakan sub menu dari master data, dimana tampilann tersebut dapat menambah, mencari, dan menghapus data-data anggota. Setiap datang anggota baru maka mereka diwajibkan untuk mendaftarkan diri dengan menyerahkan ktp, dan petugas dapat menambahkan pada menu tersebut.

1. Tampilan Halaman Data Jenis



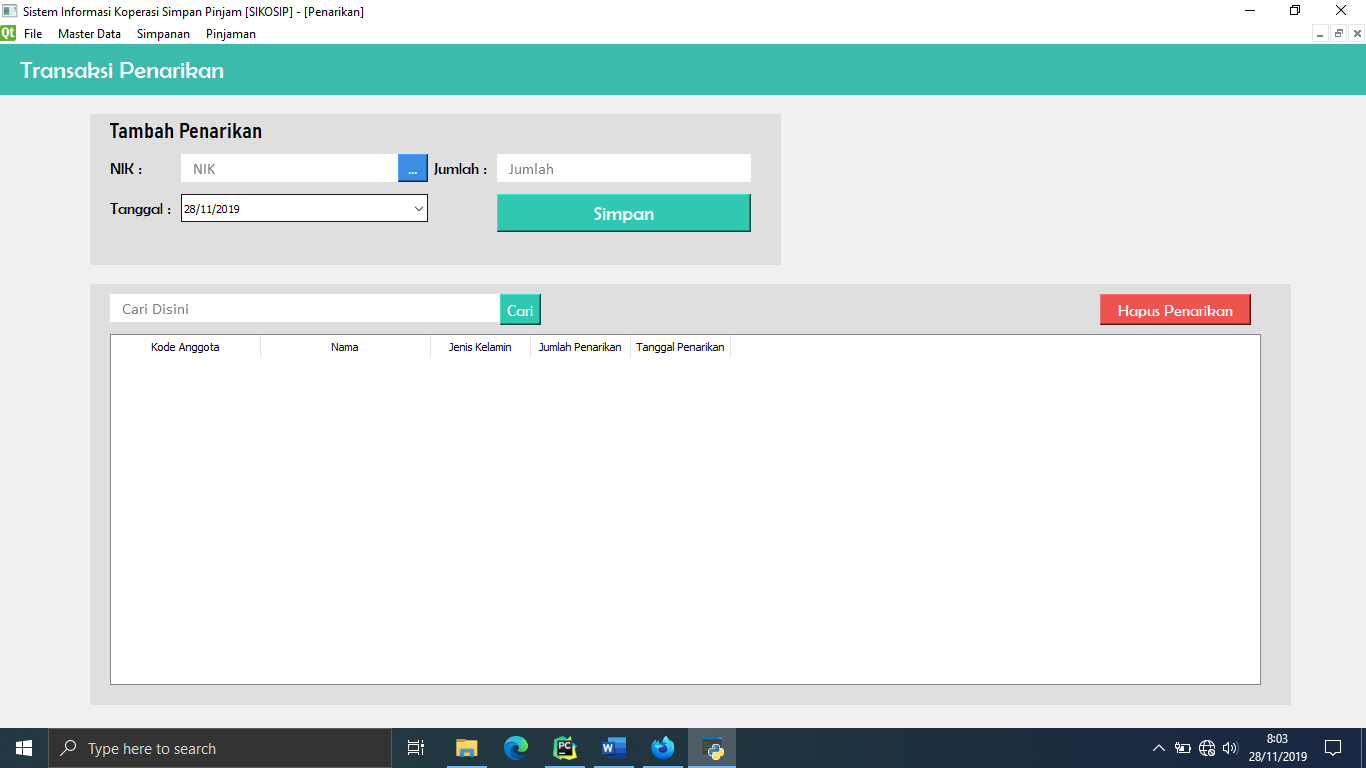
Tampilan Data jenis merupakan sub menu dari master data, yang akan menampilkan data jenis-jenis dari simpanan dan penarikan.

1. Halaman Transaksi Simpanan



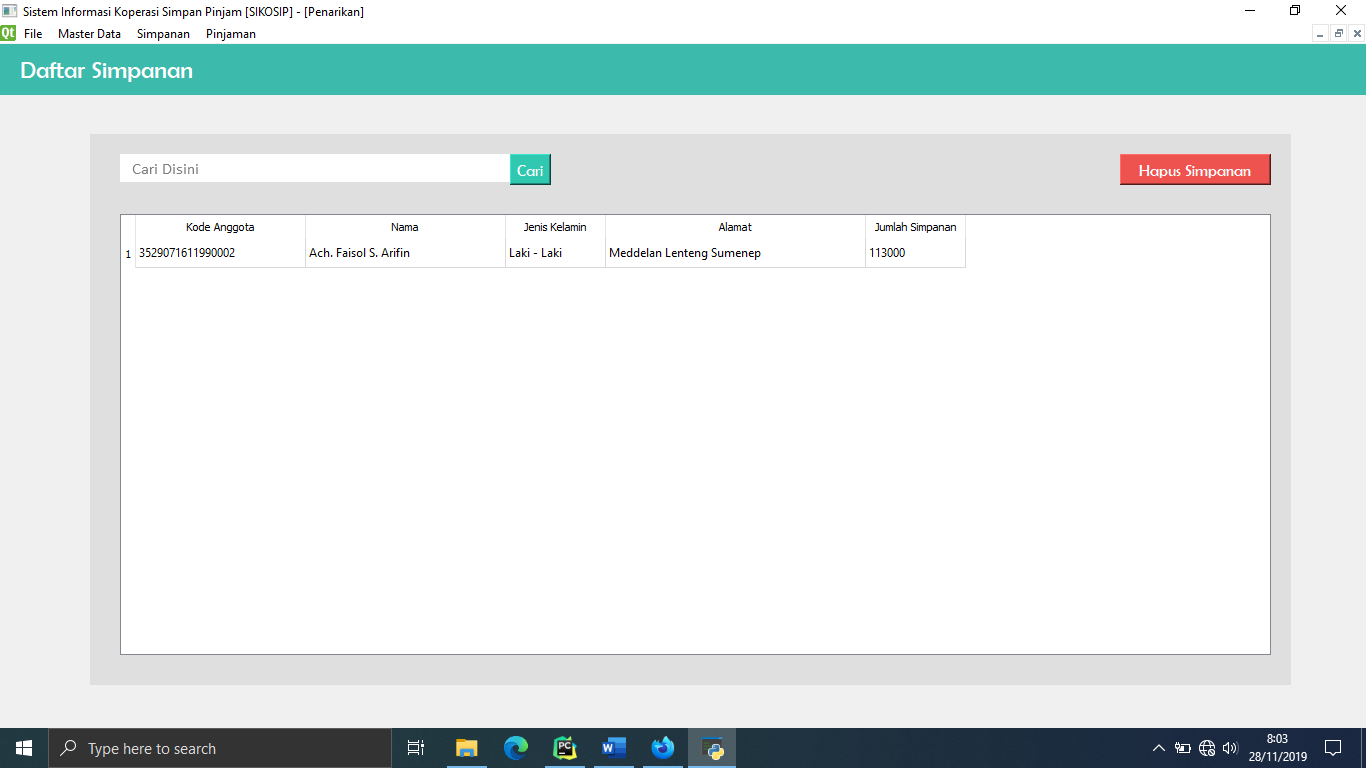
Tampilan transaksi simpanan digunakan untuk mencatat transaksi simpanan yang dilakukan anggota. ketika anggota pertama kali bergabung, maka anggota harus membayar simpanan pokok, dan membayar simpanan wajib setiap bulan, dan membayar simpanan sukarela setiap waktu.

1. Halaman Transaksi Penarikan



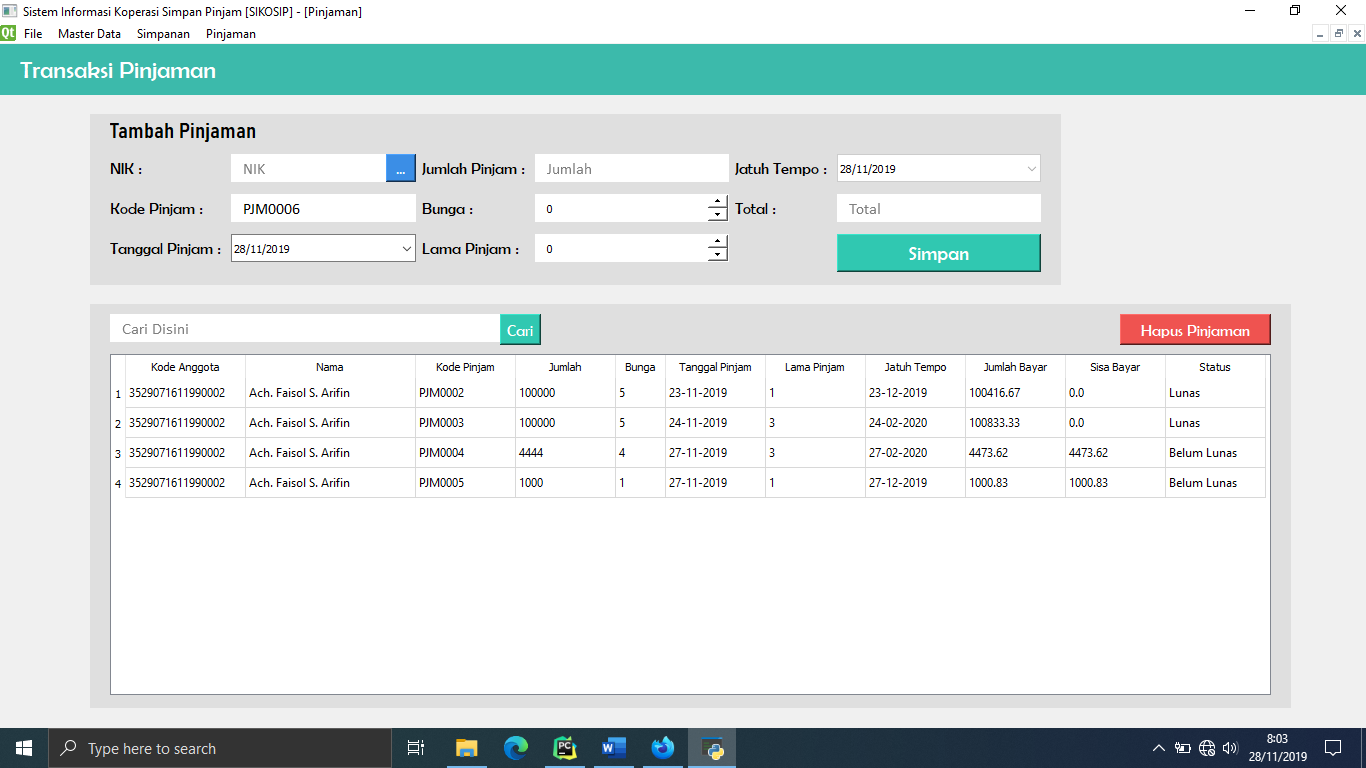
Tampilan transaksi penarikan digunakan untuk mencatat transaksi penarikan. Setiap anggota yang melakukan transaksi penarikan dapat di catat pada halaman tersebut yang secara otomatis akan mengurangi jumlah simpanannya.

1. Halaman Daftar Simpanan

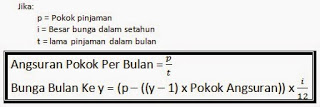


Tampilan daftar simpanan akan menampilkan informasi jumlah simpanan yang dimiliki dari masing-masing anggota. Ditampilan tersebut terdapat pencarian, dan tombol untuk menghapus data, ketika tombol hapus ditekan maka akan menghapus simpanan dan seluruh data transaksi yang dilakukan.

1. Halaman Transaksi Pinjaman

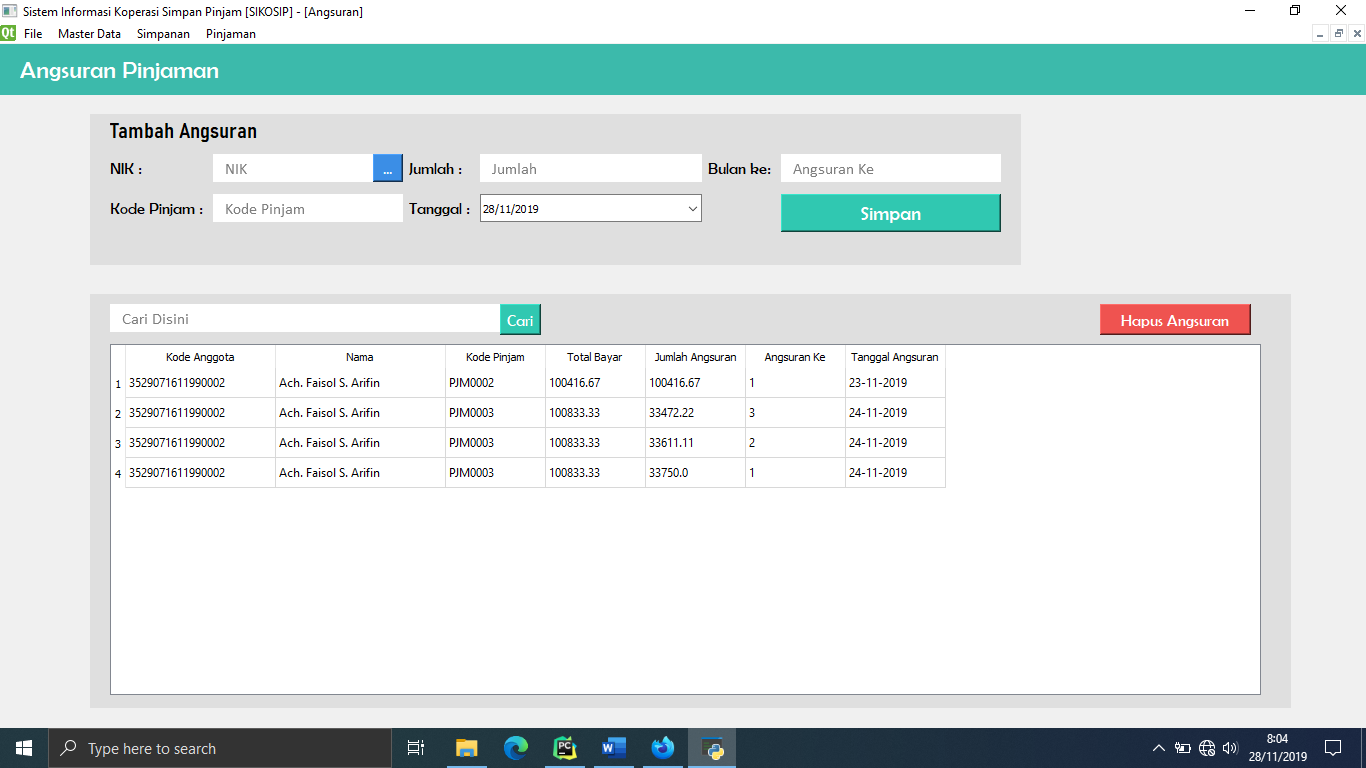


Tampilan transaksu pinjaman digunakan untuk menyimpan data transakti peminjaman uang yang dilakukan anggota. Petugas data memasukan NIK anggota, besar pinjaman, bunga, dan lama pinjam. Bersar bunga ditentukan menggunakan rumus berikut:



Dimana dalam total pembayaran akan dihitung sebanyak lama peminjaman.

1. Halaman Transaksi Angsuran



Tampilan transaksi angsuran digunakan untuk mencatat angusuran bulanan dari pinjaman anggota. Jumlah anguran perbulan telah ditentukan oleh sistem menggunakan rumus diatas. Petugas hanya bisa memasukkan NIK.

* 1. **Listing Program**

Dalam program tersebut saya pecah menjadi beberapa modul, yang kemudian diakses dari main program. dan program tersebut menggunakan tampilan MDI sehingga dapat memuat banyak modul dalam satu window.

1. Listing Program MainProgram

|  |
| --- |
| import sys  from PyQt5 import QtSql from PyQt5.uic import loadUi from PyQt5.QtWidgets import QApplication, QMainWindow  import dashboard, login, petugas, anggota, jenis, simpanan, penarikan, dfsimpan, pinjaman, angsuran  class MainWindow(QMainWindow):   def \_\_init\_\_(self):  super().\_\_init\_\_()  loadUi('./ui/main.ui', self)  self.Koneksi()   self.setCentralWidget(self.mdi)  self.menu\_file.triggered.connect(self.windowaction)  self.menu\_master.triggered.connect(self.windowaction)  self.menu\_simpanan.triggered.connect(self.windowaction)  self.menu\_pinjaman.triggered.connect(self.windowaction)   self.Petugas.setVisible(False)  self.Anggota.setVisible(False)  self.Jenis.setVisible(False)  self.Simpanan.setVisible(False)  self.Penarikan.setVisible(False)  self.Daftar\_Simpanan.setVisible(False)  self.Pinjaman.setVisible(False)  self.Angsuran.setVisible(False)   self.login = login.Login(self)  self.dash = dashboard.Dashboard()  self.petugas = petugas.Petugas()  self.anggota = anggota.Anggota(self)  self.jenis = jenis.Jenis()  self.simpanan = simpanan.Transaksi(self)  self.penarikan = penarikan.Penarikan(self)  self.dfsimpan = dfsimpan.DaftarSimpanan(self)  self.pinjaman = pinjaman.Pinjaman(self)  self.angsuran = angsuran.Angsuran(self)   self.mdi.addSubWindow(self.login)  self.login.showMaximized()  self.setWindowTitle("Sistem Informasi Koperasi Simpan Pinjam [SIKOSIP]")   def Koneksi(self):  db = QtSql.QSqlDatabase.addDatabase('QSQLITE')  db.setDatabaseName('./db/sikosip.db')  if not db.open():  self.statusBar().showMessage("Galat !! Database connection failed")  else:  self.statusBar().showMessage("Koneksi database berhasil")   def hitungBunga(self,pinjaman,bunga,lama,bulan=False):  p = pinjaman  i = bunga / 100  t = lama  angsurantotal = 0  angsuranpokok = round(p / t, 2)  if bulan is False:  for j in range(1, t + 1):  bungabulan = round((p - ((j - 1) \* angsuranpokok)) \* i / 12, 2)  totalangsuran = angsuranpokok + bungabulan  angsurantotal += totalangsuran  else:  bungabulan = round((p - ((bulan - 1) \* angsuranpokok)) \* i / 12, 2)  angsurantotal = angsuranpokok + bungabulan   return angsurantotal   def windowaction(self, m):  if m.text() == "Login":  if self.login.isActiveWindow():  self.login.showMaximized()  else:  self.mdi.addSubWindow(self.login)  self.login.showMaximized()  elif m.text() == "Data Petugas":  if self.petugas.isActiveWindow():  self.petugas.showMaximized()  else:  self.mdi.addSubWindow(self.petugas)  self.petugas.showMaximized()  elif m.text() == "Data Anggota":  if self.anggota.isActiveWindow():  self.anggota.showMaximized()  else:  self.mdi.addSubWindow(self.anggota)  self.anggota.showMaximized()  elif m.text() == "Data Jenis":  if self.jenis.isActiveWindow():  self.jenis.showMaximized()  else:  self.mdi.addSubWindow(self.jenis)  self.jenis.showMaximized()  elif m.text() == "Simpanan":  if self.simpanan.isActiveWindow():  self.simpanan.showMaximized()  else:  self.mdi.addSubWindow(self.simpanan)  self.simpanan.showMaximized()  elif m.text() == "Penarikan":  if self.penarikan.isActiveWindow():  self.penarikan.showMaximized()  else:  self.mdi.addSubWindow(self.penarikan)  self.penarikan.showMaximized()  elif m.text() == "Daftar Simpanan":  if self.dfsimpan.isActiveWindow():  self.dfsimpan.showMaximized()  else:  self.mdi.addSubWindow(self.dfsimpan)  self.dfsimpan.showMaximized()  elif m.text() == "Pinjaman":  if self.pinjaman.isActiveWindow():  self.pinjaman.showMaximized()  else:  self.mdi.addSubWindow(self.pinjaman)  self.pinjaman.showMaximized()  elif m.text() == "Angsuran":  if self.angsuran.isActiveWindow():  self.angsuran.showMaximized()  else:  self.mdi.addSubWindow(self.angsuran)  self.angsuran.showMaximized()  if \_\_name\_\_ == '\_\_main\_\_':  app = QApplication(sys.argv)  ex = MainWindow()  ex.showMaximized()  sys.exit(app.exec\_()) |

Penjelasan listing program :

Pada baris paling atas seluruh modul di import termasuk dengan sub-sub tampilan yang ada. Kemudian membuat class main program yang merupakan turunan dari qmainwindow. Kemudian dibaris 16, untuk menyatakan central mainwindow sebagai tampilan mdi.dibaris 17-20 digunakan untuk menangani event yang terjadi pada menu yang kemudian diarahkan pada method *windowaction*. Dibaris 22-29 dingunakan untuk memanipulasi menu ketika program pertama kali dijalankan agar tidak tampil. Dibaris 31-40 untuk menginstansiasi dari masing-masing modul sub window yang di import. Dibaris 46-52 membuat method koneksi dengan database sqlite yang kemudia dipanggil pada construkor mainwindow. Dibaris 54-69 digunakan untuk menghitung bunga dan menentukan bungan setiap bulanya secara otomasis, yang akan dipanggu di sub window pinjaman dan angsuran.dibaris 71-128 merupakan method untuk menangani event yang terjadi pada menu, yang kemudian akan memuat sub window yang berkaitan.

1. Listing Program Login

|  |
| --- |
| from PyQt5.QtSql import QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox, QLineEdit   class Login(QMdiSubWindow):  def \_\_init\_\_(self,main=None):  self.main = main  super().\_\_init\_\_()  loadUi('./ui/login.ui', self)  self.setWindowTitle('Login')  self.login.clicked.connect(self.cekLogin)  self.username.returnPressed.connect(self.cekLogin)  self.password.returnPressed.connect(self.cekLogin)  self.checkBox.clicked.connect(self.showHide)   def cekLogin(self):  message = QMessageBox()  query = QSqlQuery()  username = self.username.text()  password = self.password.text()  if username != '' and password != '':  query.exec\_("SELECT \* FROM petugas WHERE username='"+username+"' AND password='"+password+"'")  if query.next():  self.notif.setText('')  self.main.Petugas.setVisible(True)  self.main.Anggota.setVisible(True)  self.main.Jenis.setVisible(True)  self.main.Simpanan.setVisible(True)  self.main.Penarikan.setVisible(True)  self.main.Daftar\_Simpanan.setVisible(True)  self.main.Pinjaman.setVisible(True)  self.main.Angsuran.setVisible(True)  self.main.Login.setVisible(False)  message.setText(' \*\* Anda Berhasil Login \*\* ')  message.exec\_()  self.close()  self.main.mdi.addSubWindow(self.main.dash)  self.main.dash.showMaximized()  else:  self.notif.setText('Username atau Password Anda Salah !!')  else:  self.notif.setText('Masukkan username dan password anda !!')   def showHide(self):  if self.checkBox.isChecked():  self.password.setEchoMode(QLineEdit.Normal)  else:  self.password.setEchoMode(QLineEdit.Password) |

Penjelasan lisiting program:

Membuat class Login yang merupakan turunan dari QMDISubwindow (*karena tampilan menggunakan mdi maka, hanya membuat sub window yang merupakan turunan dari mdisubwindow*).

Dibaris 17-23 membuat method login yang digunakan untuk mengauthentikasi user yang akan mengakses halaman aplikasi, dengan kemudian menjalankan query *select* berdasarkan username dan password, jika data ditemukan didatabase maka akan memanipulasi menu pada mainwindow untuk didapat ditampilkan.

Selain itu dibaris 35-36 untuk menampilkan message box ketika login berhasil. Dibaris 45-49 membuat method untuk memanipulasi lineedit, dimana ketika chekbox dicentang maka lineedit password akan menampilkan pada mode text, dan sebaliknya menampilkan pada mode password.

1. Listing Program Dashboard

|  |
| --- |
| from PyQt5.QtCore import QPropertyAnimation, QRect from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow  class Dashboard(QMdiSubWindow):  def \_\_init\_\_(self):  super().\_\_init\_\_()  loadUi('./ui/dashboard.ui', self)  self.setWindowTitle('Dashboard')  self.Animasi()   def Animasi(self):  self.anim = QPropertyAnimation(self.label, b"geometry")  self.anim.setDuration(8000)  self.anim.setStartValue(QRect(230, -100, 945, 215))  self.anim.setEndValue(QRect(230, 180, 945, 215))  self.anim.start() |

Penjelasan listing program :

Membuat class Dashboard yang juga merupakan turunan dari Qmdisubwindow. Dibaris 12-17 membuat method animasi yang akan dijalankan ketika window tersebut dipanggil, animasi akan berefek pada label yang akan menampilkan slideup.

1. Listing Program Petugas

|  |
| --- |
| from PyQt5.QtCore import Qt from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox   class Petugas(QMdiSubWindow):  def \_\_init\_\_(self):  super().\_\_init\_\_()  loadUi('./ui/petugas.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()  self.tampilPetugas()  self.clear()   self.simpan.clicked.connect(self.simpanPetugas)  self.hapus.clicked.connect(self.hapusPetugas)  self.cari.clicked.connect(lambda : self.cariPetugas(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariPetugas(self.txtcari.text()))   self.setWindowTitle('Petugas')   def tampilPetugas(self):  self.model.setTable('petugas')  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Petugas")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Username")  self.model.setHeaderData(3, Qt.Horizontal, "Password")  self.model.setHeaderData(4, Qt.Horizontal, "No. Telp")  self.model.setHeaderData(5, Qt.Horizontal, "Alamat")  self.tableView.setModel(self.model)  self.tableView.setColumnWidth(5, 180)   def simpanPetugas(self):  nama = self.nama.text()  telp = self.telp.text()  user = self.user.text()  passw = self.passw.text()  alamat = self.alamat.toPlainText()  if nama == "" or telp == "" or user == "" or passw == "" or alamat == "" :  self.notif.setText('Masukkan data dengan benar !')  else:  self.notif.setText('')  query = QSqlQuery()  if query.exec\_("insert into petugas values(null,'"+nama +"','"+user+"','"+passw+"','"+telp+"','"+alamat+"')"):  self.message.setText(" \*\* Data berhasil ditambahkan ke database \*\*")  self.message.exec\_()  self.tampilPetugas()  self.clear()   def hapusPetugas(self):  self.model.removeRow(self.tableView.currentIndex().row())  self.model.select()  self.tampilPetugas()   def cariPetugas(self,filter):  query = QSqlQuery("SELECT \* FROM petugas WHERE nama LIKE '%" + filter + "%' OR alamat LIKE '%" + filter + "%'")  self.model.setQuery(query)  self.tableView.setModel(self.model)   def clear(self):  self.nama.setText('')  self.telp.setText('')  self.user.setText('')  self.passw.setText('')  self.alamat.setPlainText('') |

Penjelasan listing program :

Dibaris 17-20 digunakan untuk menangani event yang tejadi pada window tersebut. Dibaris 24-35 digunakan untuk menampilkan data petugas dengan menggunakan model yang memilih table petugas dan menampilkan pada table view.

Dibaris 37-52 untuk masukkan data, yang menjalankan query insert ketikan validasi telah dipenuhi dan menjalakan query dan memberikan notifikasi dengan message box.

Dibaris 54-57 untuk menghapus data pada database berdasarkan data yang yang dipilih saat ini.

Dibaris 59-62 merupakan method untuk mencari data yang di filter berdasarkan kolom nama dan alamat.

Dibaris 64-69 merupakan method untuk mengosongkan inputan.

1. Listing Program Anggota

|  |
| --- |
| from PyQt5.QtCore import Qt, QDateTime from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox   class Anggota(QMdiSubWindow):  def \_\_init\_\_(self, main=None):  super().\_\_init\_\_()  self.main = main  loadUi('./ui/anggota.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()  self.tampilAnggota()  self.clear()   self.simpan.clicked.connect(self.simpanAnggota)  self.hapus.clicked.connect(self.hapusAnggota)  self.cari.clicked.connect(lambda : self.cariAnggota(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariAnggota(self.txtcari.text()))   self.setWindowTitle('Anggota')   def tampilAnggota(self):  self.model.setTable('anggota')  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "NIK")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Tanggal Lahir")  self.model.setHeaderData(3, Qt.Horizontal, "Jenis Kelamin")  self.model.setHeaderData(4, Qt.Horizontal, "No. Telp")  self.model.setHeaderData(5, Qt.Horizontal, "Alamat")  self.model.setHeaderData(6, Qt.Horizontal, "Tanggal Gabung")  self.tableView.setModel(self.model)  self.tableView.setColumnWidth(0, 150)  self.tableView.setColumnWidth(1, 170)  self.tableView.setColumnWidth(5, 180)   def simpanAnggota(self):  nik = self.nik.text()  nama = self.nama.text()  if self.gender1.isChecked():  gender = self.gender1.text()  elif self.gender2.isChecked():  gender = self.gender2.text()  lahir = self.lahir.text().replace('/','-')  telp = self.telp.text()  alamat = self.alamat.toPlainText()  if nik == "" or nama == "" or gender == "" or lahir == "" or telp == "" or alamat == "":  self.notif.setText('Masukkan data dengan benar ! ')  elif len(nik) != 16:  self.notif.setText('NIK tidak benar !')  else:  self.notif.setText('')  query = QSqlQuery()  query.exec\_("select nik from anggota where nik='"+nik+"'")  if not query.next():  if query.exec\_("insert into anggota values('"+nik+"','"+nama +"','"+lahir+"','"+gender+"','"+telp+"','"+alamat+"',date())"):  self.tampilAnggota()  self.clear()  self.message.setText(" \t\*\* Anggota berhasil didaftar, \n Lanjutkan untuk pembayaran Simpanan Pokok \*\*")  self.message.exec\_()  self.main.mdi.addSubWindow(self.main.simpanan)  self.main.simpanan.jumlah.setFocus(True)  self.main.simpanan.nik.setText(nik)  self.main.simpanan.showMaximized()  else:  self.notif.setText('Anggota telah terdaftar !')   def hapusAnggota(self):  self.model.removeRow(self.tableView.currentIndex().row())  self.model.select()  self.tampilAnggota()   def cariAnggota(self,filter):  query = QSqlQuery("SELECT \* FROM anggota WHERE nik LIKE '%" + filter + "%' OR nama LIKE '%" + filter + "%'OR alamat LIKE '%" + filter + "%'")  self.model.setQuery(query)  self.tableView.setModel(self.model)   def clear(self):  self.nik.setText('')  self.nama.setText('')  self.gender1.setChecked(False)  self.gender2.setChecked(False)  self.lahir.setDateTime(QDateTime.currentDateTime())  self.telp.setText('')  self.alamat.setPlainText('') |

Penjelasan listing program :

Pada listing ini juga merupakan turunan dari Qmdisubwindow. Dibaris 25-39 merupakan method untuk menampilkan data anggota dengan menggunakan model yang memilih tabel anggota dari database.

Dibaris 41- 70 merupakan method untuk menambah data dimana dalam method tersebut terdapat validasi, ketika validasi terpenuhi dan nik masih belum pernah dimasukkan, maka akan menjalankan query insert untuk menambah data.

Dibaris 72-75 merupakan method untuk menghapus data berdasarkan data yang dipilih saat ini di table view.

Dibaris 77-80 merupakan method untuk melakukan pencarian dengan menjalankan query select yang di filter berdasarkan nik, nama, atau alamat. Dibaris 82-89 merupakan method untuk mengosongkan inputan.

1. Listing Program Jenis

|  |
| --- |
| from PyQt5.QtCore import Qt, QDateTime from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox   class Jenis(QMdiSubWindow):  def \_\_init\_\_(self):  super().\_\_init\_\_()  loadUi('./ui/jenis.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()  self.clear()  self.tampilkanJenis()   self.simpan.clicked.connect(self.simpanJenis)  self.hapus.clicked.connect(self.hapusJenis)   self.setWindowTitle('Jenis')   def tampilkanJenis(self):  self.model.setTable('jenis')  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Jenis")  self.model.setHeaderData(1, Qt.Horizontal, "Nama Jenis")  self.model.setHeaderData(2, Qt.Horizontal, "Keterangan")  self.tableView.setModel(self.model)  self.tableView.setColumnWidth(1, 150)  self.tableView.setColumnWidth(2, 430)   def simpanJenis(self):  jenis = self.jenis.text()  desc = self.desc.toPlainText()  if jenis == "" or desc == "":  self.notif.setText('Masukkan data dengan banar !')  else:  self.notif.setText('')  query = QSqlQuery()  if query.exec\_("insert into jenis values(null,'"+jenis +"','"+desc+"')"):  self.message.setText(" \*\* Data berhasil ditambahkan ke database \*\*")  self.message.exec\_()  self.tampilkanJenis()  self.clear()   def hapusJenis(self):  self.model.removeRow(self.tableView.currentIndex().row())  self.model.select()  self.tampilkanJenis()   def clear(self):  self.jenis.setText('')  self.desc.setPlainText('') |

Penjelasan listing program :

Pada listing tersebut membuat class yang merupakan turunan dari Qmdisubwindow. Dibaris 22-31 merupakan method yang digunakan untuk menampilkan data menggunakan model dengan memilih table jenis dari data dan ditempatkan pada table view.

Dibaris 33- 35 merupakan method untuk menambah data dengan menjalankann query insert yang berisi jenis dan deskripsi. Ketika berhasil ditambahkan maka akan menampilkan notifikasi berupa message box.

Dibaris 47-50 merupakan method untuk menghapus data dari database berdasarkan baris pada table view saat ini.

1. Listing Program Simpanan

|  |
| --- |
| from PyQt5.QtCore import Qt, QDateTime from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox  import dialogAnggota  class Transaksi(QMdiSubWindow):  def \_\_init\_\_(self,main=None):  super().\_\_init\_\_()  self.main = main  loadUi('./ui/simpanan.ui', self)  query = QSqlQuery("select \* from jenis where nama\_jenis like '%simpanan%'")  while query.next():  self.jenis.addItem(query.record().value(1))   self.message = QMessageBox()  self.model = QSqlTableModel()   self.clear()  self.tampilTransaksi()   self.simpan.clicked.connect(self.saveSimpanan)  self.cari\_nik.clicked.connect(self.getAnggota)  self.hapus.clicked.connect(self.hapusSimpanan)  self.cari.clicked.connect(lambda : self.cariPetugas(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariSimpanan(self.txtcari.text()))   self.setWindowTitle('Penarikan')   def tampilTransaksi(self):  query = QSqlQuery("SELECT a.nik, a.nama, a.gender, j.id\_jenis, j.nama\_jenis, s.\* FROM anggota a, jenis j, transaksi s WHERE a.nik=s.nik AND j.id\_jenis=s.id\_jenis AND j.id\_jenis!=4")  self.model.setQuery(query)  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Anggota")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Jenis Kelamin")  self.model.setHeaderData(4, Qt.Horizontal, "Jenis Simpanan")  self.model.setHeaderData(8, Qt.Horizontal, "Jumlah")  self.model.setHeaderData(9, Qt.Horizontal, "Tanggal Simpan")  self.tableView.setModel(self.model)   self.tableView.setColumnHidden(3,True)  self.tableView.setColumnHidden(5,True)  self.tableView.setColumnHidden(6,True)  self.tableView.setColumnHidden(7,True)  self.tableView.setColumnWidth(0, 150)  self.tableView.setColumnWidth(1, 170)  self.tableView.setColumnWidth(4, 170)    def saveSimpanan(self):  nik = self.nik.text()  jenis = self.jenis.currentIndex()  jumlah = self.jumlah.text()  tanggal = self.tanggal.text().replace('/','-')  if nik == "" or jumlah == "" or tanggal == "":  self.notif.setText('Masukkan data dengan benar !')  elif len(nik) != 16:  self.notif.setText('Nik tidak benar !')  else:  self.notif.setText('')  query = QSqlQuery()  if query.exec\_("insert into transaksi values(null,'"+nik+"','"+str(jenis+1)+"','"+jumlah+"','"+tanggal+"')"):  query.exec\_("select nik from simpanan where nik='"+nik+"'")  if query.next():  query.exec\_("update simpanan set jumlah=(select jumlah from simpanan where nik='"+nik+"') + "+jumlah+" where nik='"+nik+"'")  else:  query.exec\_("insert into simpanan values(null,'"+nik+"','"+jumlah+"')")  self.message.setText(" \*\* Data berhasil ditambahkan ke database \*\*")  self.message.exec\_()  self.tampilTransaksi()  self.clear()  self.main.dfsimpan.tampilSimpanan()   def getAnggota(self):  anggota = dialogAnggota.DialogAnggota()  anggota.exec\_()  self.nik.setText(anggota.datanik)   def clear(self):  self.nik.setText('')  self.jenis.setCurrentIndex(0)  self.jumlah.setText('')  self.tanggal.setDateTime(QDateTime.currentDateTime())   def hapusSimpanan(self):  query = QSqlQuery()  index = self.tableView.currentIndex().row()  datamodel = self.tableView.model()  kode\_trans = datamodel.data(datamodel.index(index, 5), 0)  query.exec\_("delete from transaksi where id\_trans='" + str(kode\_trans) + "'")  self.tampilTransaksi()   def cariSimpanan(self,filter):  query = QSqlQuery("SELECT a.nik, a.nama, a.gender, j.id\_jenis, j.nama\_jenis, s.\* FROM anggota a, jenis j, transaksi s WHERE a.nik=s.nik AND j.id\_jenis=s.id\_jenis AND j.id\_jenis!=4 AND (a.nik LIKE '%" + filter + "%' OR a.nama LIKE '%" + filter + "%' OR j.nama\_jenis LIKE '%" + filter + "%' OR s.tgl\_trans LIKE '%" + filter + "%')")  self.model.setQuery(query)  self.tableView.setModel(self.model) |

Penjelasan listing program :

Pada listing tersebut membuat class Transaksi yang merupakan turunan dari Qmdisubwindow. 13-15 merupakan sintak untuk menambahkan data pada pada combobox yang mengambil dari table simpanan. Dibaris 31-42 merupakan method untuk menampilkan data, dimana dalam query tersebut menggubungkan dari 3 table, yaitu table jenis, anggota, dan transaksi berdasarkan kesamaan key masing-masing table.

Dibaris 52-74 merupakan method dimana dalam query pertama akan menambahkan pada table transaksi, di query kedua dilakukan pengecekan terlebih dahulu ketika ketika pada table simpanan belum ada data dengan nik yang dimasukkan maka akan menambahkan data pada table simpanan, sebaliknya akan mengubah jumlah simpanan pada table simpanan.

Dibaris 76-79 merupakan method untuk mengambil nik dari sebuah dialog yang kemudian dimasukkan pada lineedit.

Dibaris 31-35 merupakan method untuk mengosongkan inputan dengan memberikan nilai kosongan.

Dibaris 87-93 merupakan method untuk menghapus data dengan kode transaksi dari model berdasarkan baris yang saat ini dari table view dan menghapus mengunggunakan query delete berdasarkan id\_transaksi.

Dibaris 85-98 merupakan method untuk melakukan pencarian dengan memfilter berdasarkan nik, nama, jenis simpanan, atau tanggal transaksi.

1. Listing Program Penarikan

|  |
| --- |
| from PyQt5.QtCore import Qt, QDateTime from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox  import dialogAnggota  class Penarikan(QMdiSubWindow):  def \_\_init\_\_(self,main=None):  super().\_\_init\_\_()  self.main=main  loadUi('./ui/penarikan.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()   self.clear()  self.tampilTransaksi()   self.simpan.clicked.connect(self.simpanPenarikan)  self.cari\_nik.clicked.connect(self.getAnggota)  self.hapus.clicked.connect(self.hapusPenarikan)  self.cari.clicked.connect(lambda : self.cariPenarikan(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariPenarikan(self.txtcari.text()))   self.setWindowTitle('Penarikan')   def tampilTransaksi(self):  query = QSqlQuery("SELECT a.nik, a.nama, a.gender, t.\* FROM anggota a, transaksi t WHERE a.nik=t.nik AND t.id\_jenis=4")  self.model.setQuery(query)  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Anggota")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Jenis Kelamin")  self.model.setHeaderData(6, Qt.Horizontal, "Jumlah Penarikan")  self.model.setHeaderData(7, Qt.Horizontal, "Tanggal Penarikan")  self.tableView.setModel(self.model)   self.tableView.setColumnHidden(3,True)  self.tableView.setColumnHidden(4,True)  self.tableView.setColumnHidden(5,True)  self.tableView.setColumnWidth(0, 150)  self.tableView.setColumnWidth(1, 170)   def simpanPenarikan(self):  nik = self.nik.text()  jumlah = self.jumlah.text()  tanggal = self.tanggal.text().replace('/','-')  if nik == "" or jumlah == "" or tanggal == "":  self.notif.setText('Masukkan data dengan benar !')  elif len(nik) != 16:  self.notif.setText('Nik tidak benar !')  else:  self.notif.setText('')  query = QSqlQuery()  query.exec\_("select nik from simpanan where nik='" + nik + "'")  if query.next():  query.exec\_("insert into transaksi values(null,'" + nik + "','4','" + jumlah + "','" + tanggal + "')")  query.exec\_("select jumlah from simpanan where nik='" + nik + "'")  if query.next():  hasil = query.record().value(0)  hasil -= int(jumlah)  if hasil < 0:  self.message.setText(" \*\* Penarikan melebihi jumlah simpanan \*\*")  self.message.exec\_()  else:  query.exec\_("update simpanan set jumlah='" + str(hasil) + "' where nik='" + nik + "'")  self.message.setText(" \*\* Data berhasil ditambahkan ke database \*\*")  self.message.exec\_()  self.tampilTransaksi()  self.clear()  self.main.dfsimpan.tampilSimpanan()  else:  self.message.setText(" \*\* Anggota belum pernah menyimpan \*\*")  self.message.exec\_()   def getAnggota(self):  anggota = dialogAnggota.DialogAnggota()  anggota.exec\_()  self.nik.setText(anggota.datanik)   def clear(self):  self.nik.setText('')  self.jumlah.setText('')  self.tanggal.setDateTime(QDateTime.currentDateTime())   def hapusPenarikan(self):  query = QSqlQuery()  index = self.tableView.currentIndex().row()  datamodel = self.tableView.model()  kode\_trans = datamodel.data(datamodel.index(index, 3), 0)  query.exec\_("delete from transaksi where id\_trans='" + str(kode\_trans) + "'")  self.tampilTransaksi()   def cariPenarikan(self,filter):  query = QSqlQuery("SELECT a.nik, a.nama, a.gender, t.\* FROM anggota a, transaksi t WHERE a.nik=t.nik AND t.id\_jenis=4 AND (a.nik LIKE '%" + filter + "%' OR a.nama LIKE '%" + filter + "%' OR t.jumlah LIKE '%" + filter + "%'OR t.tgl\_trans LIKE '%" + filter + "%')")  self.model.setQuery(query)  self.tableView.setModel(self.model) |

Penjelasan listing program :

Dibaris 28-44 merupakan method untuk menampilkan data dengan menggunakan query yang memilih dari table jenis, anggota, dan transaksi dimana jenis adalah penarikan.

Dibaris 46-76 merupakan method untuk menambahkan data pada table transaksi, kemudian mengambil jumlah simpanan yang lama dikurangi dengan jumlah penarikan, dan menjalankan query update untuk mengubah jumlah simpanan.

Dibaris 78-99 sama dengan class yang ada pada class transaksi.

1. Listing Program Daftar Simpanan

|  |
| --- |
| from PyQt5.QtCore import Qt from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox   class DaftarSimpanan(QMdiSubWindow):  def \_\_init\_\_(self,main=None):  super().\_\_init\_\_()  self.main = main  loadUi('./ui/daftar simpan.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()  self.tampilSimpanan()  self.setWindowTitle('Penarikan')  self.hapus.clicked.connect(self.hapusSimpanan)  self.cari.clicked.connect(lambda : self.cariSimpanan(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariSimpanan(self.txtcari.text()))   def tampilSimpanan(self):  query = QSqlQuery("SELECT a.nik, a.nama, a.gender, a.alamat, s.\* FROM anggota a, simpanan s WHERE a.nik=s.nik")  self.model.setQuery(query)  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Anggota")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Jenis Kelamin")  self.model.setHeaderData(3, Qt.Horizontal, "Alamat")  self.model.setHeaderData(6, Qt.Horizontal, "Jumlah Simpanan")  self.tableView.setModel(self.model)   self.tableView.setColumnHidden(4,True)  self.tableView.setColumnHidden(5,True)  self.tableView.setColumnWidth(0, 170)  self.tableView.setColumnWidth(1, 200)  self.tableView.setColumnWidth(3, 260)   def hapusSimpanan(self):  query = QSqlQuery()  index = self.tableView.currentIndex().row()  datamodel = self.tableView.model()  nik = datamodel.data(datamodel.index(index, 0), 0)  query.exec\_("delete from transaksi where nik='" + str(nik) + "'")  query.exec\_("delete from simpanan where nik='" + str(nik) + "'")  self.tampilSimpanan()  self.main.simpanan.tampilTransaksi()  self.main.penarikan.tampilTransaksi()   def cariSimpanan(self,filter):  query = QSqlQuery("SELECT a.nik, a.nama, a.gender, a.alamat, s.\* FROM anggota a, simpanan s WHERE a.nik=s.nik AND (a.nik LIKE '%" + filter + "%' OR a.nama LIKE '%" + filter + "%' OR a.alamat LIKE '%" + filter + "%' OR s.jumlah LIKE '%" + filter + "%')")  self.model.setQuery(query)  self.tableView.setModel(self.model) |

Penjelasan listing program :

Pada listing tersebut membuat class DaftarSimpanan yang merupakan turunan dari Qmdisubwindow. Dibaris 21-37 merupakan method untuk menampilkan data mengunggunakan query select, dengan memilih dari table anggota simpanan. Baris 31-37 untuk manipulasi table dengan mengatur panjang dari kolom dan menghidden kolom 4 dan 5.

Dibaris 39-48 merupakan method untuk menghapus data dengan menggunakan query dari table transaksi dan simpana berdasarkan NIK yang dipilih.

1. Listing Program Pinjaman

|  |
| --- |
| from PyQt5.QtCore import Qt, QDate from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox  import dialogAnggota  class Pinjaman(QMdiSubWindow):  def \_\_init\_\_(self,main=None):  super().\_\_init\_\_()  self.main = main  loadUi('./ui/pinjaman.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()   self.tampilPinjaman()  self.clear()  self.autocode()   self.simpan.clicked.connect(self.savePinjaman)  self.bunga.valueChanged.connect(self.setTotal)  self.jumlah.textChanged.connect(self.setTotal)  self.lamapinjam.valueChanged.connect(self.setTempo)  self.cari\_nik.clicked.connect(self.getAnggota)   self.hapus.clicked.connect(self.hapusPinjaman)  self.cari.clicked.connect(lambda : self.cariPinjaman(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariPinjaman(self.txtcari.text()))   self.setWindowTitle('Pinjaman')   def autocode(self):  model = QSqlTableModel()  query = QSqlQuery("select *max*(id\_pinjam) as codepinjam from pinjaman")  model.setQuery(query)  code = model.record(0).value('codepinjam')  if code == '':  autocode = 1  else:  autocode = int(code[3:]) + 1  self.kd\_pinjam.setText('PJM{:04d}'.format(autocode))   def tampilPinjaman(self):  query = QSqlQuery("SELECT a.nik, a.nama, p.\* FROM anggota a, pinjaman p where a.nik=p.nik")  self.model.setQuery(query)  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Anggota")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Kode Pinjam")  self.model.setHeaderData(4, Qt.Horizontal, "Jumlah")  self.model.setHeaderData(5, Qt.Horizontal, "Bunga")  self.model.setHeaderData(6, Qt.Horizontal, "Tanggal Pinjam")  self.model.setHeaderData(7, Qt.Horizontal, "Lama Pinjam")  self.model.setHeaderData(8, Qt.Horizontal, "Jatuh Tempo")  self.model.setHeaderData(9, Qt.Horizontal, "Jumlah Bayar")  self.model.setHeaderData(10, Qt.Horizontal, "Sisa Bayar")  self.model.setHeaderData(11, Qt.Horizontal, "Status")  self.tableView.setModel(self.model)   self.tableView.setColumnHidden(3, True)  self.tableView.setColumnWidth(0, 120)  self.tableView.setColumnWidth(1, 170)  self.tableView.setColumnWidth(5, 50)   def savePinjaman(self):  nik = self.nik.text()  kode = self.kd\_pinjam.text()  jumlah = self.jumlah.text()  bunga = self.bunga.value()  lama\_pinjam = self.lamapinjam.value()  jml\_bayar = self.total.text()  tgl\_pinjam = self.tgl\_pinjam.text().replace('/','-')  jth\_tempo = self.jatuh\_tempo.text().replace('/','-')  if nik == "" or kode == "" or jumlah == "" or bunga == "" or lama\_pinjam == "" or jml\_bayar == "":  self.notif.setText('Harap Lengkapi isian anda !')  else:  self.notif.setText('')  query = QSqlQuery()  if query.exec\_("insert into pinjaman values('" + kode + "','" + nik + "','" + jumlah + "','" + str(  bunga) + "','" + tgl\_pinjam + "','" + str(  lama\_pinjam) + "','" + jth\_tempo + "','" + jml\_bayar + "','" + jml\_bayar + "','Belum Lunas')"):  self.message.setText(" \*\* Data berhasil ditambahkan ke database \*\*")  self.message.exec\_()  self.clear()  self.autocode()  self.tampilPinjaman()   def setTotal(self):  jumlah = self.jumlah.text()  bunga = self.bunga.value()  lama = self.lamapinjam.value()  if jumlah != "" and lama != 0:  bunga = self.main.hitungBunga(int(jumlah), bunga, lama)  self.total.setText(str(bunga))   def setTempo(self):  tanggal = self.tgl\_pinjam.date()  bulan = self.lamapinjam.value()  self.jatuh\_tempo.setDate(tanggal.addMonths(bulan))  self.setTotal()   def getAnggota(self):  anggota = dialogAnggota.DialogAnggota()  anggota.exec\_()  self.nik.setText(anggota.datanik)   def hapusPinjaman(self):  query = QSqlQuery()  index = self.tableView.currentIndex().row()  datamodel = self.tableView.model()  kode\_pinjam = datamodel.data(datamodel.index(index, 2), 0)  query.exec\_("delete from pinjaman where id\_pinjam='"+kode\_pinjam+"'")  query.exec\_("delete from angsuran where id\_pinjam='"+kode\_pinjam+"'")  self.tampilPinjaman()  self.main.angsuran.tampilAngsuran()  self.autocode()   def clear(self):  self.nik.setText('')  self.jumlah.setText('')  self.bunga.setValue(0)  self.lamapinjam.setValue(0)  self.total.setText('')  self.tgl\_pinjam.setDate(QDate.currentDate())  self.jatuh\_tempo.setDate(QDate.currentDate())   def cariPinjaman(self,filter):  query = QSqlQuery("SELECT a.nik, a.nama, p.\* FROM anggota a, pinjaman p where a.nik=p.nik AND (a.nik LIKE '%" + filter + "%' OR a.nama LIKE '%" + filter + "%' OR p.id\_pinjam LIKE '%" + filter + "%' OR p.tgl\_pinjam LIKE '%" + filter + "%'OR p.jatuh\_tempo LIKE '%" + filter + "%' OR p.status LIKE '%" + filter + "%')")  self.model.setQuery(query)  self.tableView.setModel(self.model) |

Penjelasan listing program:

Pada listing tersebut berisi class Pinjaman yang merupakan turunan dari Qmdisubwindow. Dibaris 33-42 merupakan method untuk membuat kode otomatis dengan menjalakan query max id\_pinjam untuk mendapatkan id terakhir dari table pinjaman dengan menambah dengan 1.

Dibaris 44-60 merupakan method untuk menampilkan data dengan menggunakan query yang memanggil dari tabel anggota dan pinjaman yang kemudian dimasukkan pada table view.

Dibaris 67-86 merupakan method untuk menambahkan data pinjaman dengan menjalankan query insert pada table pinjaman dan menampilkan notifikasi dalam menampilkan notifikasi message box ketika berhasil ditambahkan.

Dibaris 88-94 merupakan method yang akan memberikan menangani event ketika memasukkan jumlah bunga yang secara otomatis akan menghitung total pembayaran.

Dibaris 96-100 merupakan method yang akan menentukan jatuh tempo secara otomatis berdasarkan lama peminjaman yang di masukkan.

Dibaris 107-113 merupakan method yang akan menghapus data menggunakan query delete pada tabel simpanan dan angsuran.

Dibaris 118-125 merupakan method yang nantinya digunakan untuk mengosongkan inputan.

Dibaris 127-130 merupakan method yang digunakan untuk melakukan pencarian dengan menjalankan query select yang difilter berdasarkan nik, nama, id\_pinjam, tanggal pinjam, jatuh tempo, atau status pinjaman.

1. Listing Program Angsuran

|  |
| --- |
| from PyQt5.QtCore import Qt, QDateTime from PyQt5.QtSql import QSqlTableModel, QSqlQuery from PyQt5.uic import loadUi from PyQt5.QtWidgets import QMdiSubWindow, QMessageBox  import dialogAnggota  class Angsuran(QMdiSubWindow):  def \_\_init\_\_(self,main=None):  super().\_\_init\_\_()  self.main = main  loadUi('./ui/angsuran.ui', self)   self.message = QMessageBox()  self.model = QSqlTableModel()   self.clear()  self.tampilAngsuran()   self.simpan.clicked.connect(self.simpanAngsuran)  self.cari\_nik.clicked.connect(self.getAnggota)  self.hapus.clicked.connect(self.hapusAngsuran)  # self.cari.clicked.connect(lambda : self.cariPetugas(self.txtcari.text()))  self.txtcari.returnPressed.connect(lambda : self.cariAngsuran(self.txtcari.text()))   self.setWindowTitle('Angsuran')   def tampilAngsuran(self):  query = QSqlQuery("SELECT a.nik, a.nama, p.id\_pinjam, p.jumlah\_bayar, ang.\* FROM anggota a, pinjaman p, angsuran ang where a.nik=ang.nik and p.id\_pinjam=ang.id\_pinjam")  self.model.setQuery(query)  self.model.setEditStrategy(QSqlTableModel.OnFieldChange)  self.model.select()  self.model.setHeaderData(0, Qt.Horizontal, "Kode Anggota")  self.model.setHeaderData(1, Qt.Horizontal, "Nama")  self.model.setHeaderData(2, Qt.Horizontal, "Kode Pinjam")  self.model.setHeaderData(3, Qt.Horizontal, "Total Bayar")  self.model.setHeaderData(7, Qt.Horizontal, "Jumlah Angsuran")  self.model.setHeaderData(8, Qt.Horizontal, "Angsuran Ke")  self.model.setHeaderData(9, Qt.Horizontal, "Tanggal Angsuran")  self.tableView.setModel(self.model)   self.tableView.setColumnHidden(4,True)  self.tableView.setColumnHidden(5,True)  self.tableView.setColumnHidden(6,True)  self.tableView.setColumnWidth(0, 150)  self.tableView.setColumnWidth(1, 170)   def simpanAngsuran(self):  nik = self.nik.text()  kd\_pinjam = self.kd\_pinjam.text()  jumlah = self.jumlah.text()  angsur = self.angsuran.text()  tanggal = self.tanggal.text().replace('/','-')  query = QSqlQuery()  query.exec\_("select nik, id\_pinjam from pinjaman where nik='"+nik+"' and id\_pinjam='"+kd\_pinjam+"'")  if query.next():  query.exec\_("insert into angsuran values(null,'" + nik + "','" + kd\_pinjam + "','" + jumlah + "','" + angsur + "', '" + tanggal + "')")  query.exec\_("update pinjaman set sisa=(select sisa from pinjaman where nik='"+nik+"' and id\_pinjam='"+kd\_pinjam+"') - "+jumlah+" where nik='"+nik+"' and id\_pinjam='"+kd\_pinjam+"'")  self.message.setText("\*\* Data berhasil ditambahkan ke database \*\*")  self.message.exec\_()  self.clear()  self.tampilAngsuran()  self.main.pinjaman.tampilPinjaman()  else:  self.message.setText("\*\* Anggota tidak ditemukan \*\*")  self.message.exec\_()   def getAnggota(self):  anggota = dialogAnggota.DialogAnggota()  anggota.exec\_()  self.nik.setText(anggota.datanik)   if anggota.datanik != '':  query = QSqlQuery()  query.exec\_("select id\_pinjam from pinjaman where nik='"+anggota.datanik+"' and status like '%belum lunas%'")  if query.next():  kd\_pinjam = query.value(0)  self.kd\_pinjam.setText(kd\_pinjam)  query.exec\_("SELECT pinjaman.id\_pinjam, pinjaman.jumlah, pinjaman.bunga, pinjaman.lama\_pinjam, angsuran.id\_pinjam, *max*(angsuran.angsur\_ke) as angsur FROM pinjaman LEFT JOIN angsuran ON pinjaman.id\_pinjam = angsuran.id\_pinjam where pinjaman.id\_pinjam='"+kd\_pinjam+"' or angsuran.id\_pinjam='"+kd\_pinjam+"'")  if query.next():  bulan\_ke = query.value(5)  if bulan\_ke == '':  bulan\_ke = 1  else:  bulan\_ke += 1  print(bulan\_ke)  self.jumlah.setText(str(self.main.hitungBunga(query.value(1), query.value(2), query.value(3), bulan\_ke)))  self.angsuran.setText(str(bulan\_ke))  else:  self.message.setText("\*\* Pinjaman anggota tersebut tidak ditemukan ! \*\*")  self.message.exec\_()  self.clear()   def hapusAngsuran(self):  query = QSqlQuery()  index = self.tableView.currentIndex().row()  datamodel = self.tableView.model()  kode\_angsur = datamodel.data(datamodel.index(index, 4), 0)  query.exec\_("delete from angsuran where id\_angsuran='"+str(kode\_angsur)+"'")  self.tampilAngsuran()   def clear(self):  self.nik.setText('')  self.kd\_pinjam.setText('')  self.jumlah.setText('')  self.angsuran.setText('')  self.tanggal.setDateTime(QDateTime.currentDateTime())   def cariAngsuran(self,filter):  query = QSqlQuery("SELECT a.nik, a.nama, p.id\_pinjam, p.jumlah\_bayar, ang.\* FROM anggota a, pinjaman p, angsuran ang where a.nik=ang.nik and p.id\_pinjam=ang.id\_pinjam AND (a.nik LIKE '%" + filter + "%' OR a.nama LIKE '%" + filter + "%' OR p.id\_pinjam LIKE '%" + filter + "%' OR ang.tgl\_angsur LIKE '%" + filter + "%')")  self.model.setQuery(query)  self.tableView.setModel(self.model) |

Penjelasan listing program:

Pada listing program tesebut membuat class Angsuran yang merupakan turunan dari Qmdisubwindow. Dibaris 20-24 merupakan proses penangan event yang diberikan.

Dibaris 28-40 merupakan method yang digunakan untuk namapilkan data dengan menggunakan query select dari tabel anggota, pinjaman, dan angsuran. Yang kemudian query dimasukkna pada model dan ditampilkan pada table view.

Dibaris 46-6 merupakan method untuk menambah data angsuran, di baris 55 query pertama untuk memasukkan data dengan query insert pada tabel angsuran. Dibaris 56 query kedua untuk mengubah sisa pinjaman yang telah diangsur. Dimana sisa pinjaman yang lama dikurangi dengan jumlah angsuran.

Dibaris 68-92 merupakan method untuk memberikan nilai secara otomatis pada inputan dengan menggunakan query select dari tabel pinjaman yang di join dengan angsuran. Baris 86 akan memberikan nilai jumlah angsuran otomatis yang dihitung dengan method hitungBunga dari mainwindow.

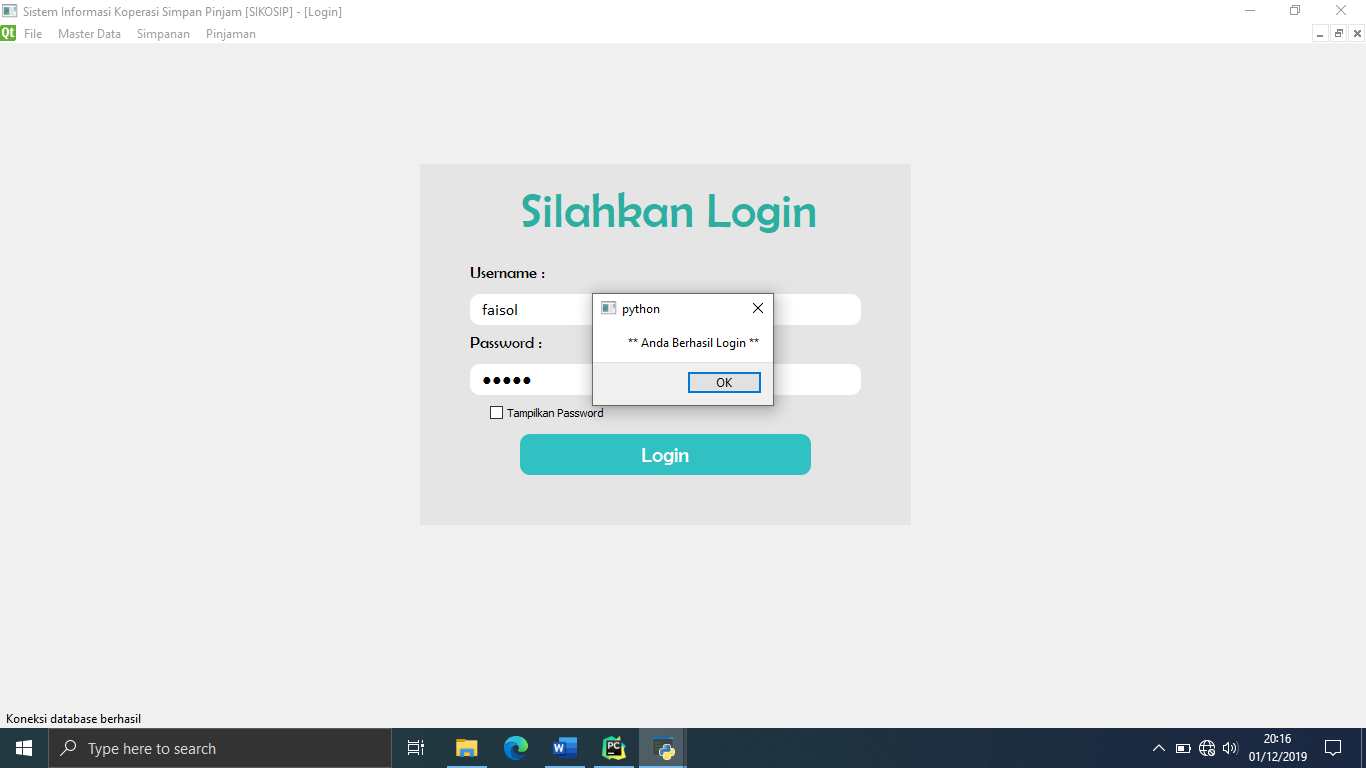
Dibaris 93-99 merupakan method untuk menghapus data angsuran dengan menggunakan query delete berdasarkan id ansuran.

Dibaris 101-106 merupakan method yang akan menangani untuk mengosongkan inputan.

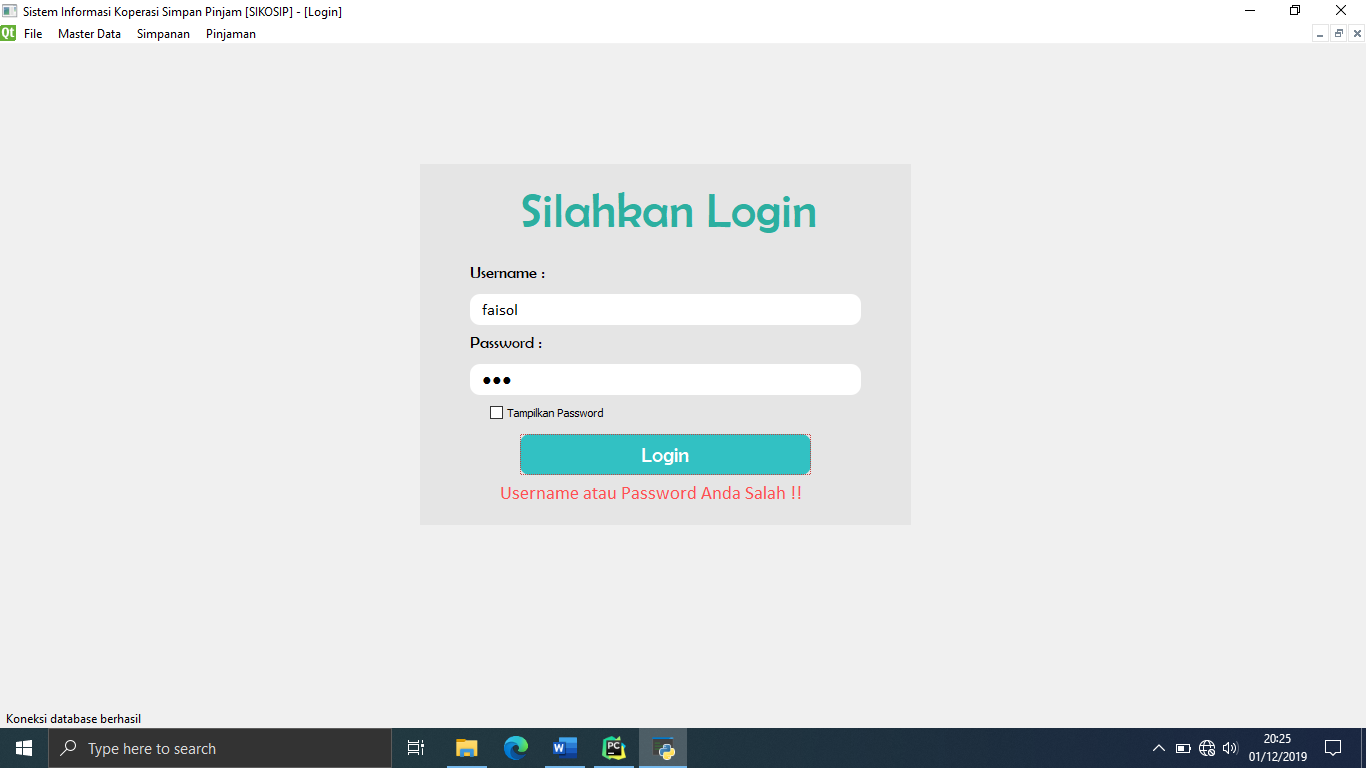
Dibaris 108-111 merupakan method untuk melakukan pencarian dengan memfilter berdasarkan nik, nama, id pinjam, dan tanggal angsuran.

* 1. **Capture Hasil Running**

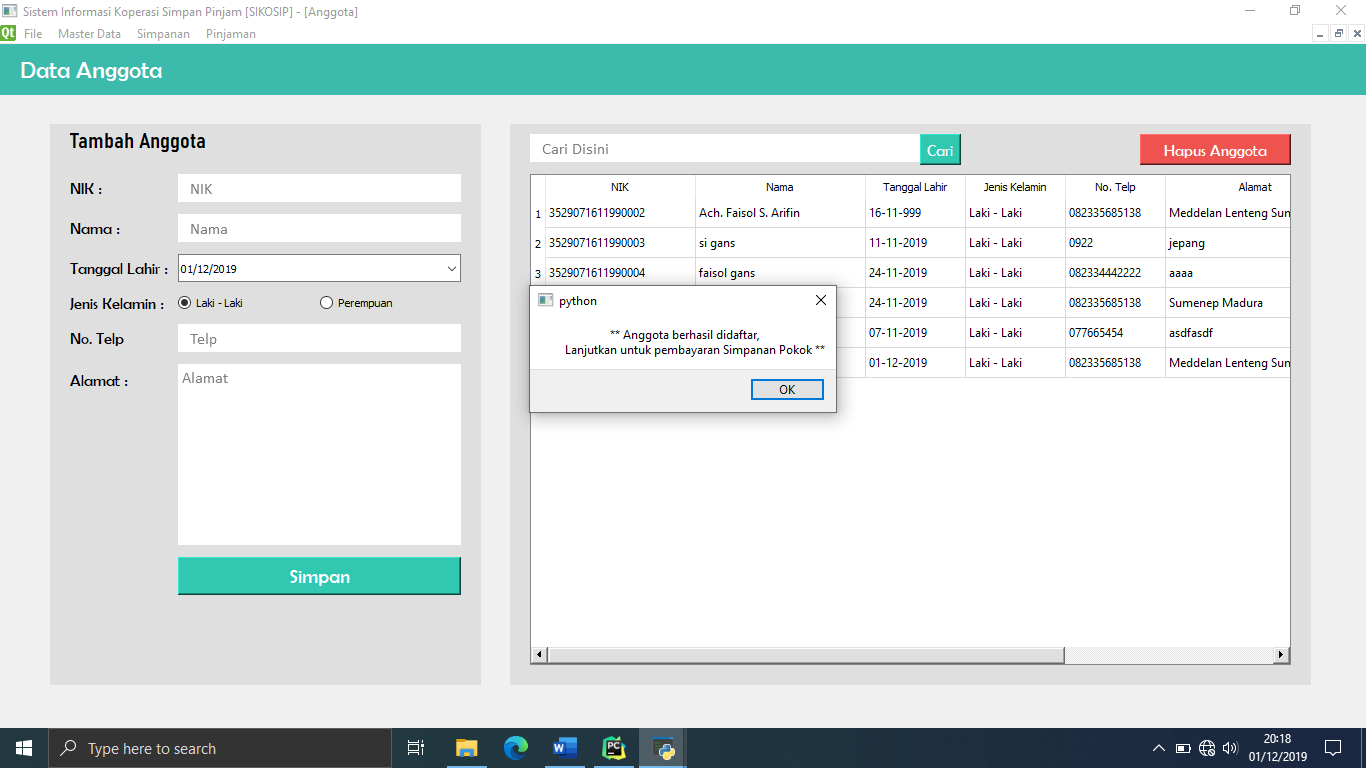
1. Ketika authentikasi berhasil



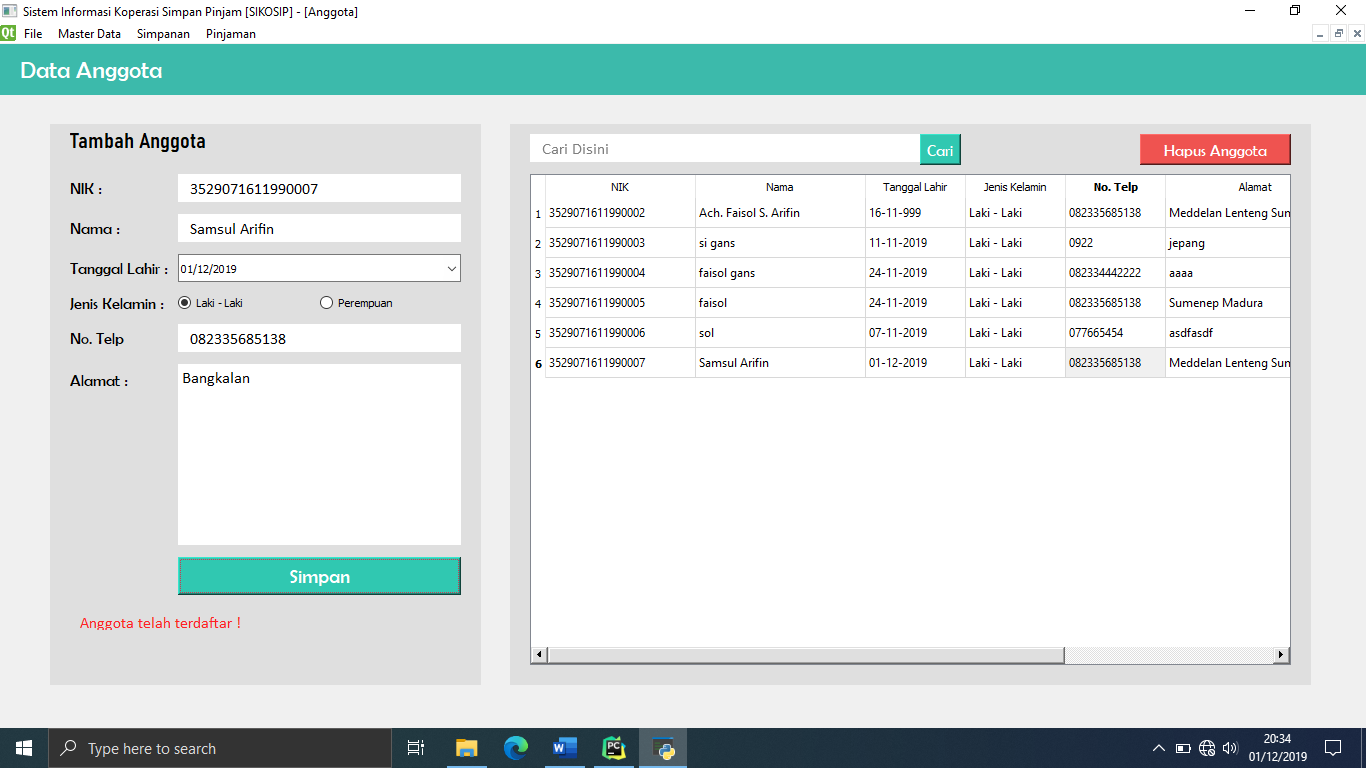
1. Ketika authentikasi gagal



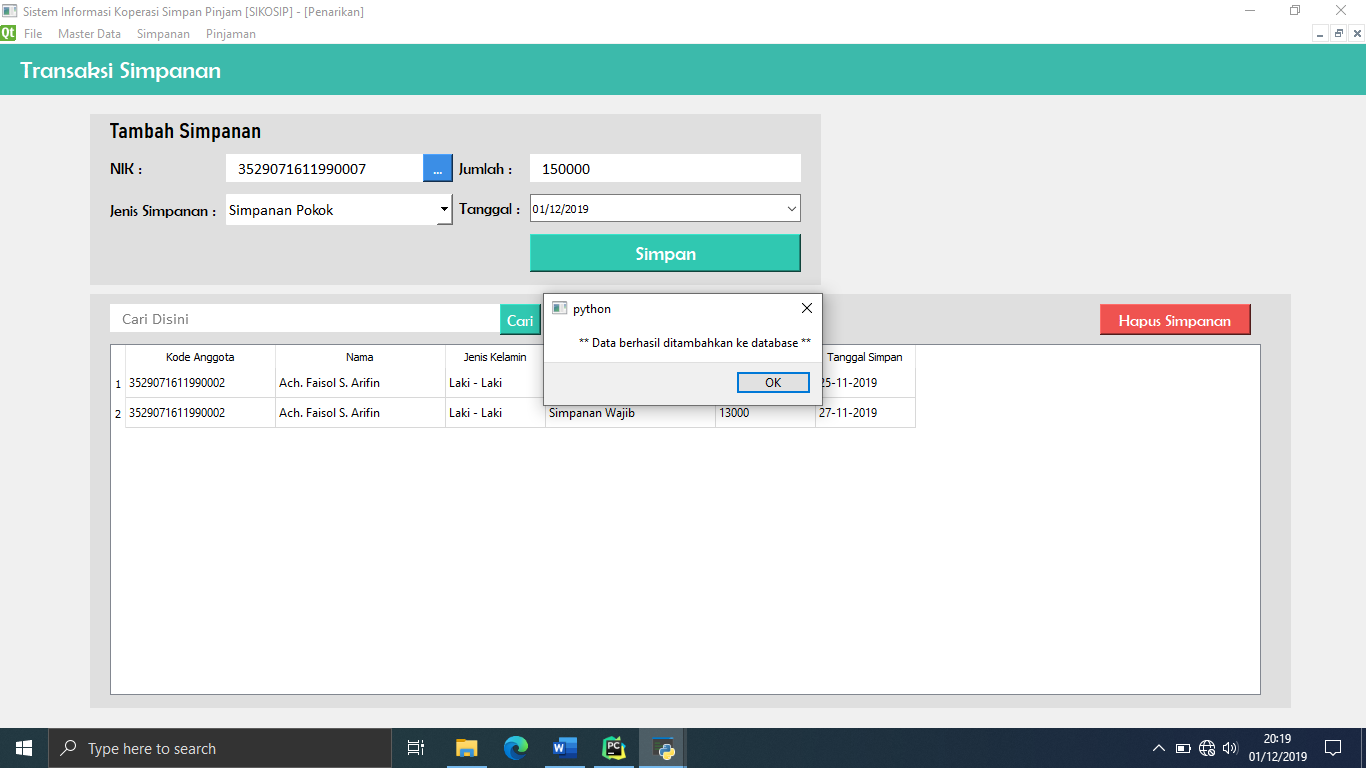
1. Berhasil menambahkan data anggota



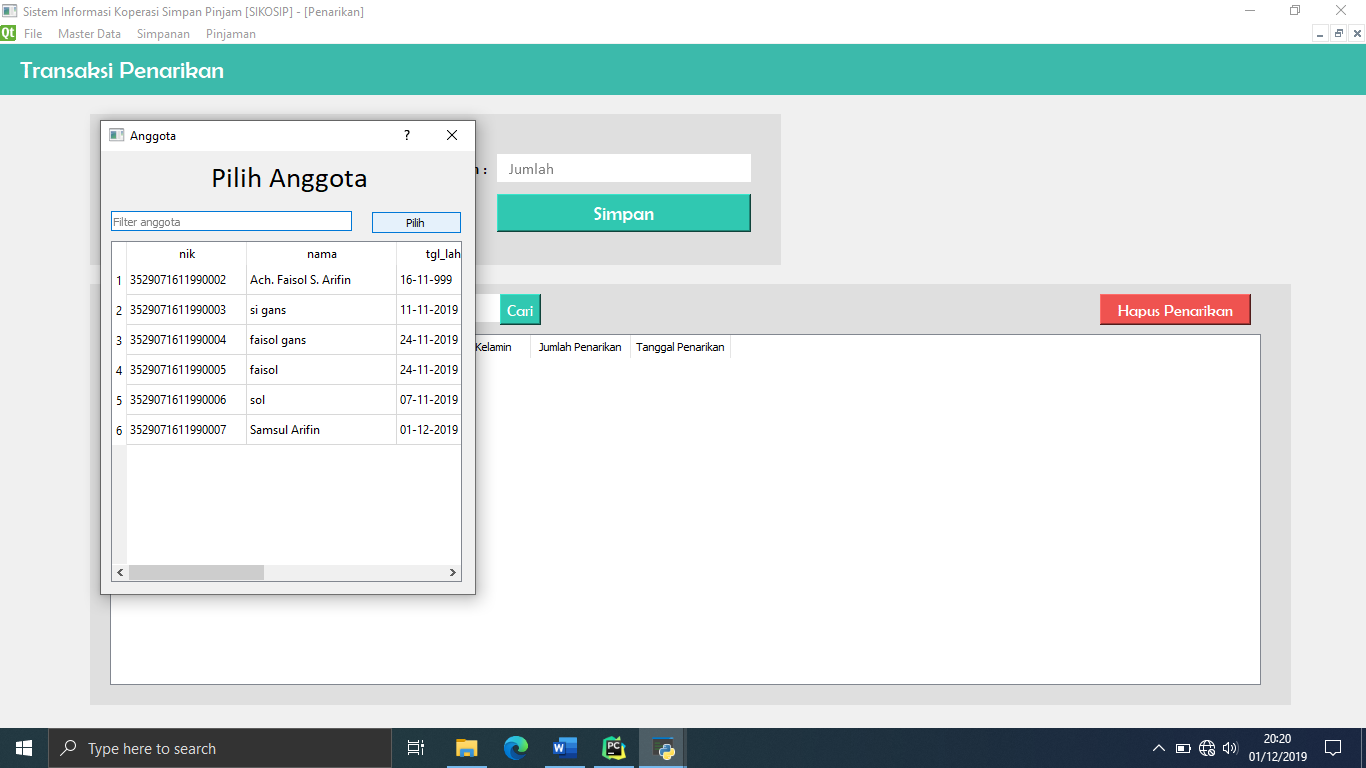
1. Ketika anggota telah terdaftar



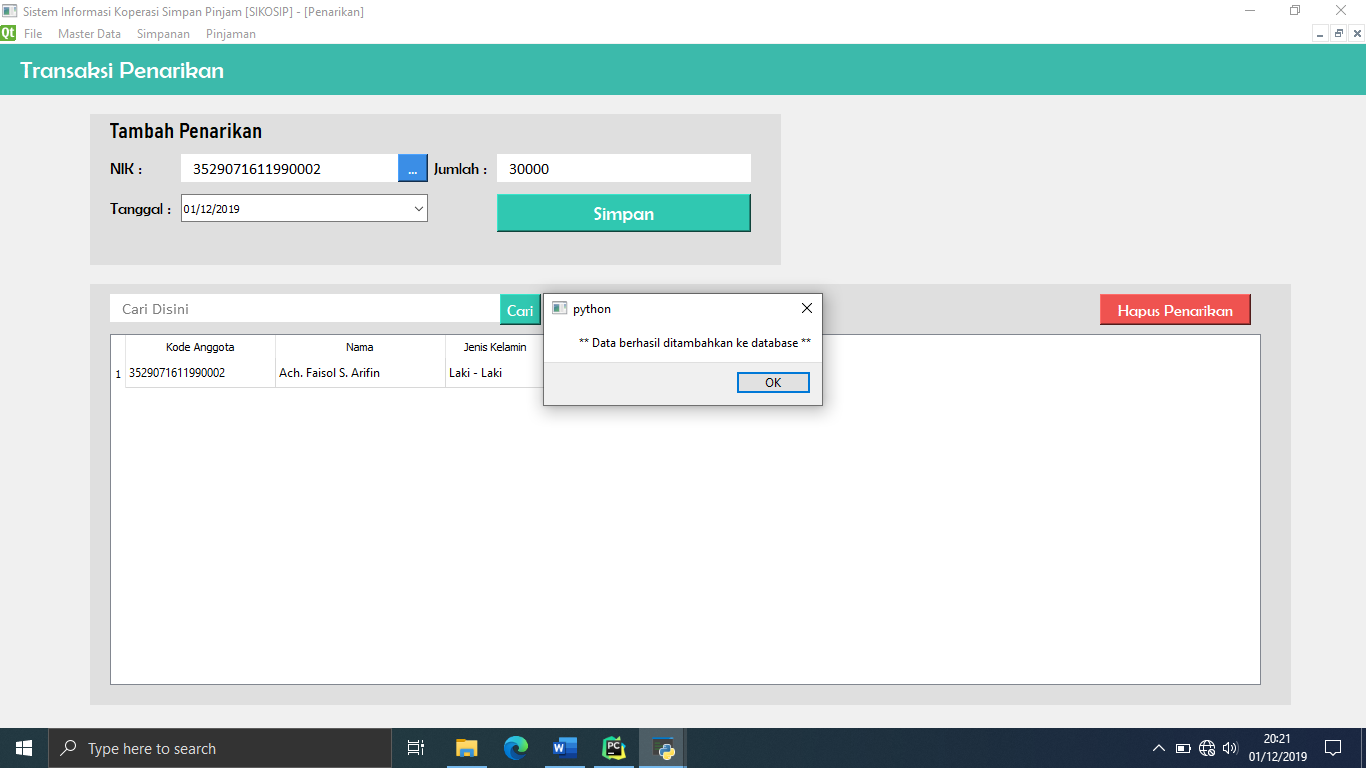
1. Berhasil menambahkan transaksi simpanan



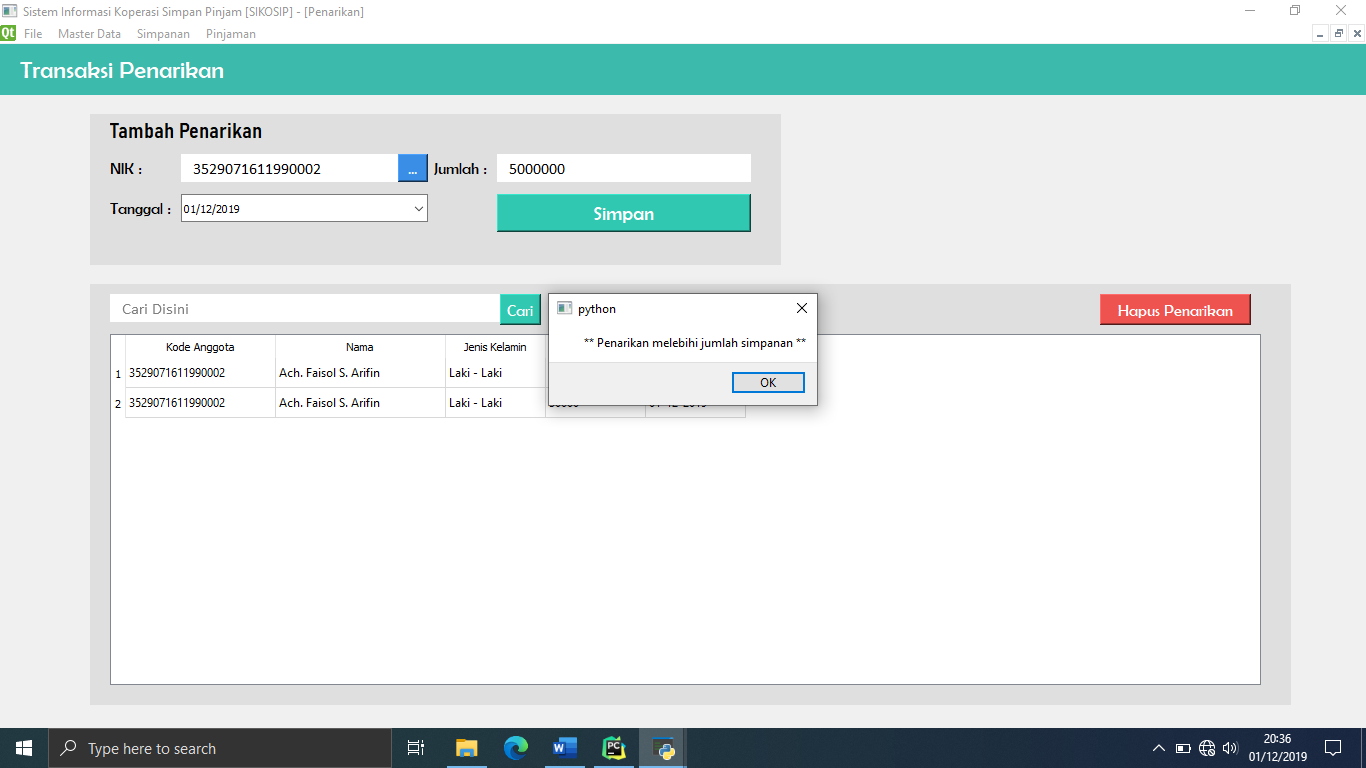
1. Dialog window untuk mencari nik



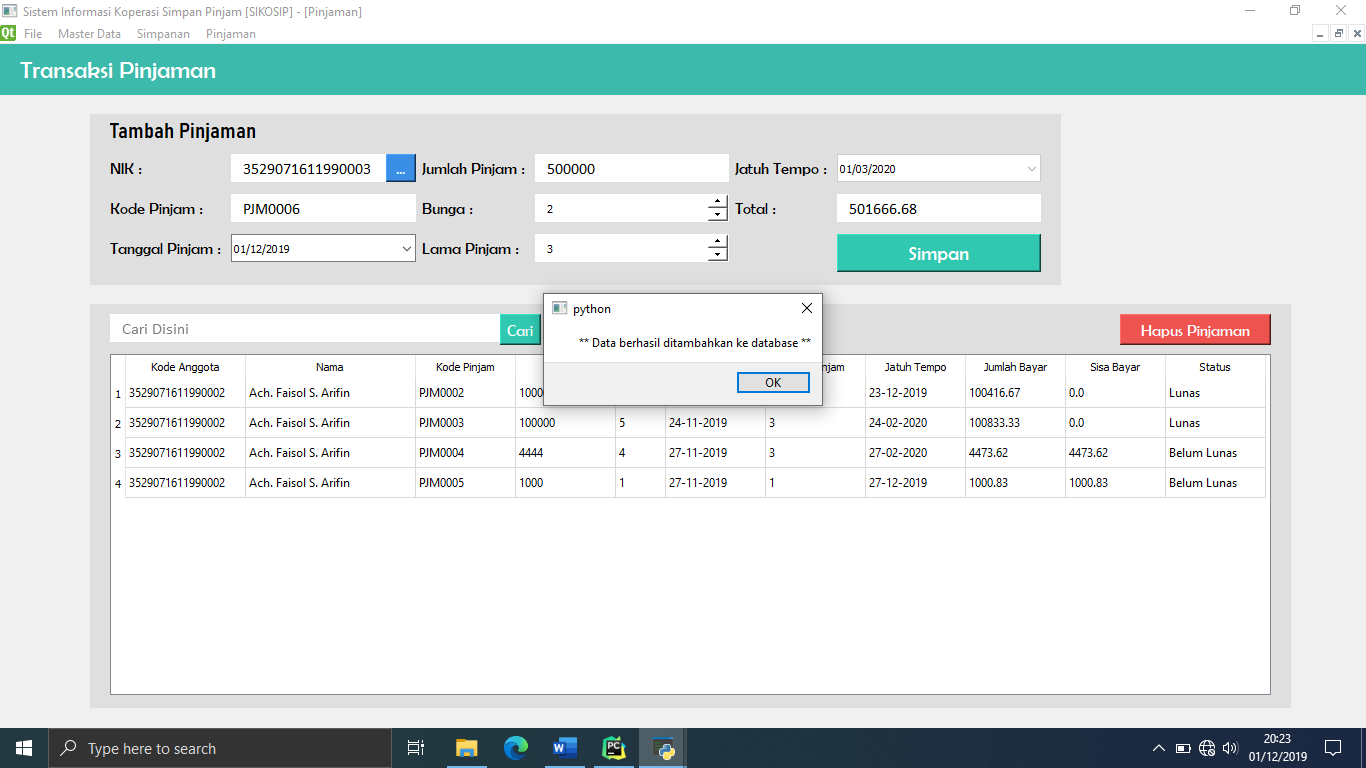
1. Berhasil menambahkan transaksi penarikan



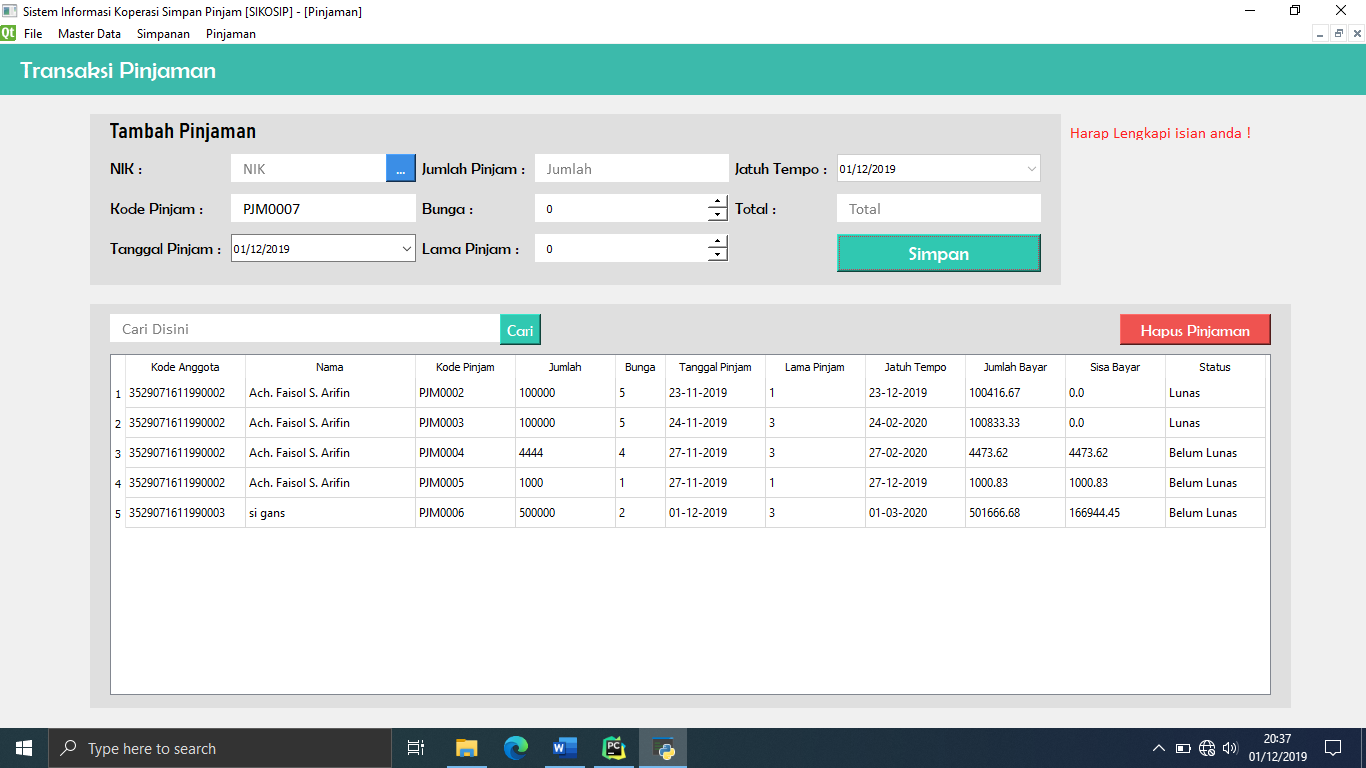
1. Ketika penarikan melebih jumlah simpanan



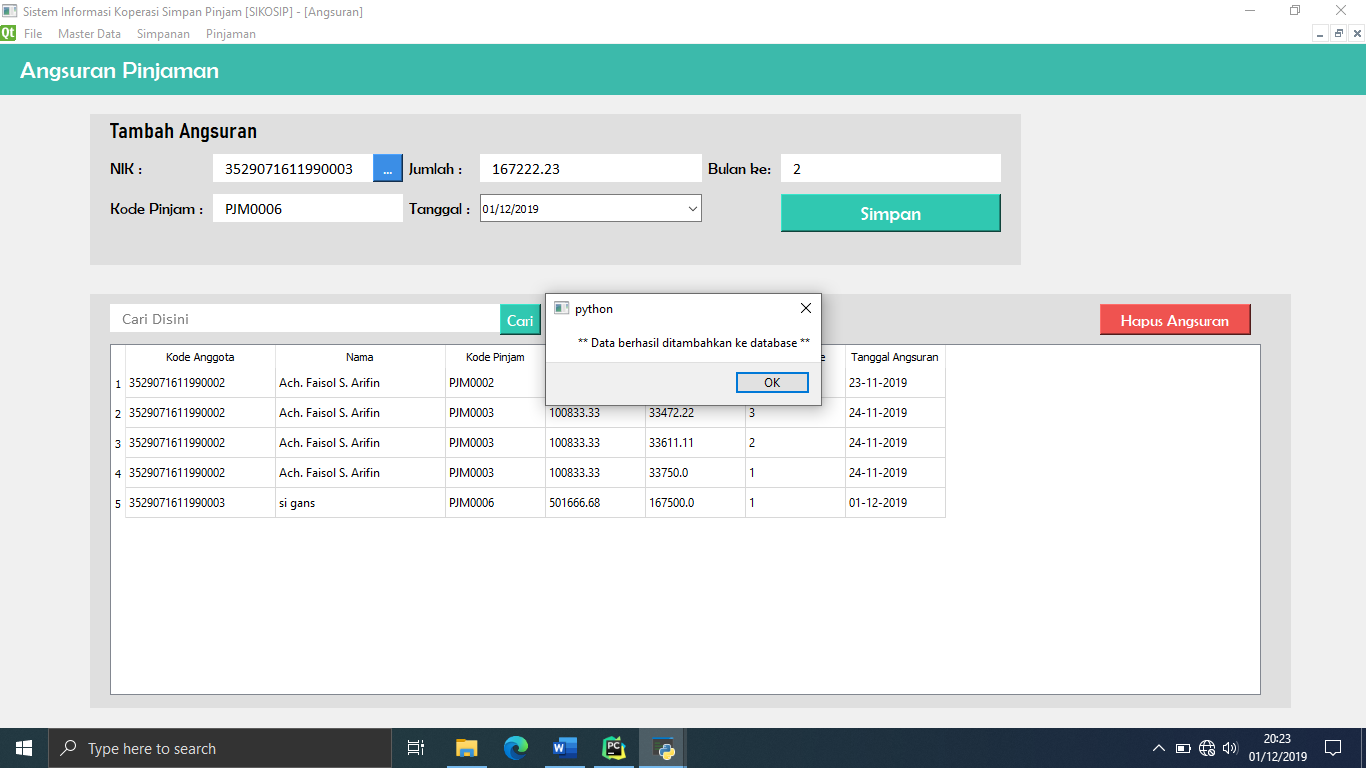
1. Berhasil menambahkan data pinjaman



1. Validari di setiap form



1. Berhasil menambahkan data angsuran



1. Tidak menemukan pinjaman anggota

