

LAPORAN
TUGAS AKHIR

APLIKASI GUI PPDB

Mata Kuliah Pemrograman Berorientasi Objek



Disusun Oleh:

Willy Rahma Wijaya
Mochammad Alvin Apriyanto
Allvintantyo Fristya Gita Nanda

2213020192
2213020024
2213020207

Daftar Isi

Kata Pengantar	3
BAB I Pendahuluan	4
Bab II Pembahasan	5
A. Aplikasi GUI.....	5
B. QT Designer.....	5
C. Python	5
D. MySQL	5
Flowchart dan Diagram Class	6
A. Flowchart	6
B. Class Diagram.....	7
Hasil Pemrograman dan Pembahasan.....	8
Tampilan Utama:	8
Tampilan menu Pendaftaran:.....	9
Tampilan Menu Input Soal:.....	10
Tampilan kuis :	11
Tampilan Menu Hasil Seleksi setelah Kuis:.....	11
Tampilan Menu Game Edukatif:	12

Kata Pengantar

Aplikasi Penerimaan Pendaftaran Peserta Didik Baru SD (PPDB SD) merupakan hasil ide kami dalam mata kuliah pemrograman berorientasi objek untuk memenuhi ujian akhir semester. Aplikasi ini tidak hanya untuk menginput pendaftaran tetapi memiliki game edukatif berupa puzzle. Game edukatif yang kami tawarkan tidak hanya menyuguhkan keseruan semata, melainkan juga dirancang secara teliti untuk mendukung pemahaman konsep-konsep pelajaran dengan cara yang interaktif dan menghibur. Kami yakin bahwa melalui pendekatan ini, anak-anak akan lebih termotivasi dan terlibat aktif dalam belajar, sambil tetap mengembangkan keterampilan kognitif dan pemecahan masalah.

Dengan adanya integrasi teknologi melalui game edukatif dalam program PPDB kami, kami berharap dapat menjadi mitra dalam mempersiapkan generasi yang tangguh, kreatif, dan siap menghadapi tantangan masa depan. Kini juga dapat dilihat bahwa Pendidikan dan teknologi dapat berkolaborasi dengan baik dan sejalan dengan sistem yang diinginkan.

Salam Kami

Kelompok 7

BAB I

Pendahuluan

Dalam dunia pendidikan yang terus berkembang, integrasi teknologi telah menjadi suatu keharusan untuk meningkatkan pengalaman belajar. Penerapan teknologi dalam pembelajaran bukan hanya sekadar tren, melainkan juga merupakan respons terhadap kebutuhan generasi muda yang tumbuh dalam era digital.

Program Penerimaan Peserta Didik Baru (PPDB) untuk Sekolah Dasar (SD) yang kami tawarkan hadir sebagai langkah progresif dalam pendidikan. Dengan melihat bagaimana anak-anak semakin terlibat dengan teknologi, kami menyadari pentingnya menciptakan lingkungan belajar yang menggabungkan teknologi dengan pendidikan yang efektif.

Oleh karena itu, kami memperkenalkan game edukatif sebagai bagian tak terpisahkan dari proses PPDB SD. Game ini tidak hanya sebagai sarana hiburan semata, tetapi juga sebagai alat pembelajaran yang menarik dan interaktif bagi anak-anak. Melalui permainan ini, mereka dapat belajar sambil bermain, memperdalam pemahaman mereka terhadap materi pelajaran, serta mengembangkan keterampilan-keterampilan penting seperti pemecahan masalah, kreativitas, dan keterampilan sosial.

Penggunaan game edukatif dalam PPDB SD ini didukung oleh pemahaman akan kebutuhan pembelajaran yang beragam di antara peserta didik. Dengan menyediakan metode pembelajaran yang berbeda, kami berharap dapat memenuhi kebutuhan individu setiap anak, memotivasi mereka untuk belajar, serta memberikan pengalaman belajar yang menyenangkan dan bermakna.

Kami percaya bahwa integrasi teknologi melalui game edukatif dalam PPDB SD bukan hanya menghadirkan kesempatan baru dalam pendidikan, tetapi juga membuka pintu menuju pemahaman yang lebih mendalam dan penerimaan yang lebih luas terhadap pembelajaran yang beragam.

Bab II

Pembahasan

Pembahasan ini akan membahas *software development* apa saja yang kelompok kami pakai.

A. Aplikasi GUI

Menurut Lastiansah (2012), *user interface* adalah cara program dan pengguna berinteraksi. Istilah *user interface* terkadang digunakan sebagai pengganti *human computer interaction* (HCI) dimana semua aspek dari interaksi pengguna dan komputer. *Graphical User Interface* (GUI) adalah mekanisme interaksi utama antara perangkat dan pengguna untuk memilih berbagai jenis tindakan. Apa yang berinteraksi dengan pengguna adalah kumpulan elemen yang disebut objek seperti tombol dan *icon*. Mereka bisa dilihat, didengar, disentuh, atau dirasakan. Objek selalu terlihat oleh pengguna dan digunakan untuk melakukan tugas. Contoh interaksi yang dilakukan yang dilakukan pengguna dalam sistem *graphic user interface* (GUI) meliputi mengakses dan memodifikasi dengan menunjuk, memilih, dan memanipulasi (Wilbert O. Galitz, 2002: 4).

B. QT Designer

Qt Designer adalah antarmuka grafis yang diisi dengan widget Qt dan alat lain yang digunakan untuk membangun GUI. Menggunakan drag aplikasi Qt Designer dan drop interface, Anda dapat membuat dan menyesuaikan dialog Anda sendiri, jendela, dan Widget (Willman, Joshua M, 2020).

C. Python

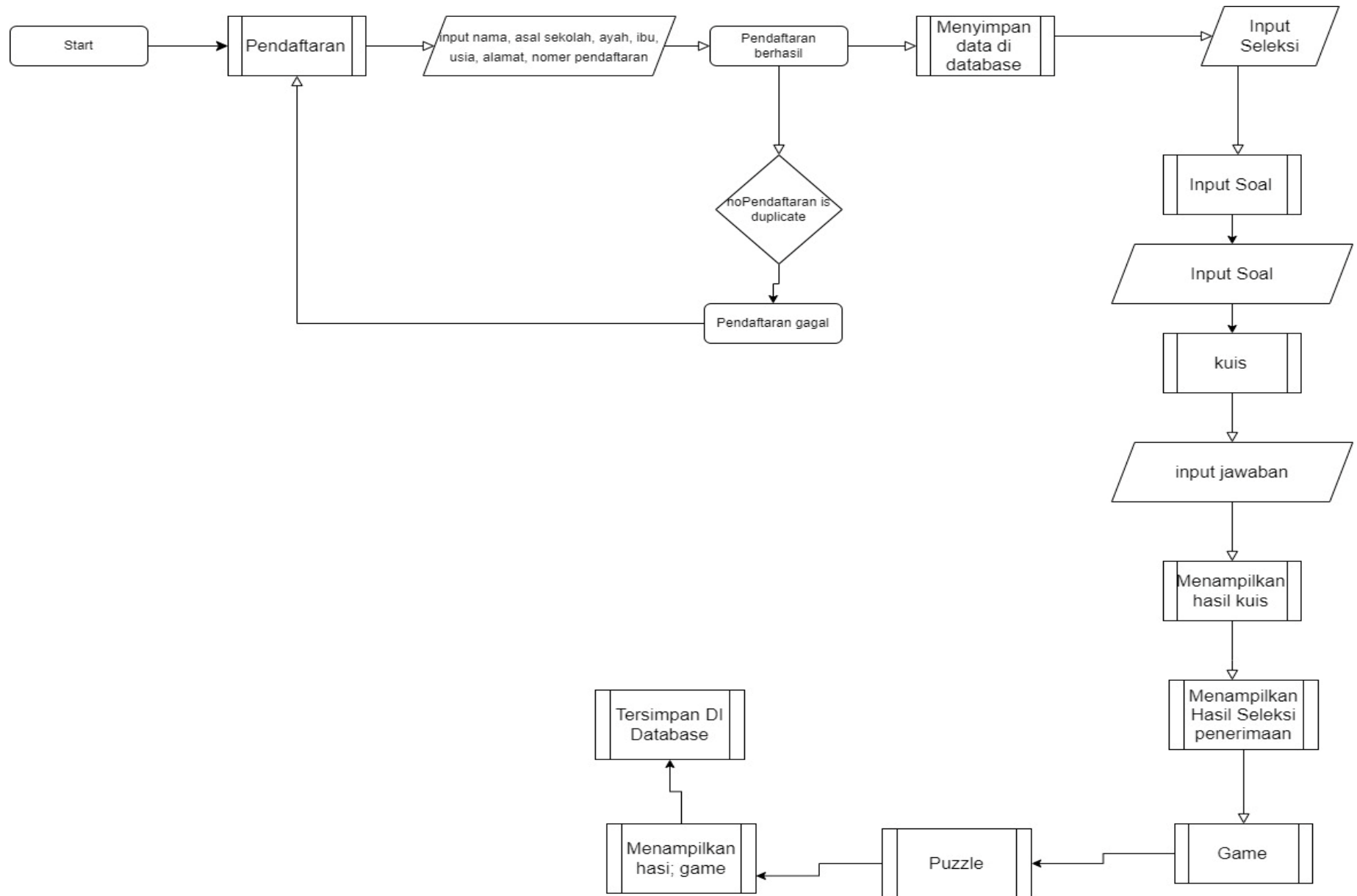
Aplikasi ini dibuat dengan python versi 3.12 dengan menggunakan modul library PyQt5. Python adalah bahasa pemrograman yang fleksibel dan sederhana yang didefinisikan dalam dokumen dokumennya (Nosrati, 2011).

D. MySQL

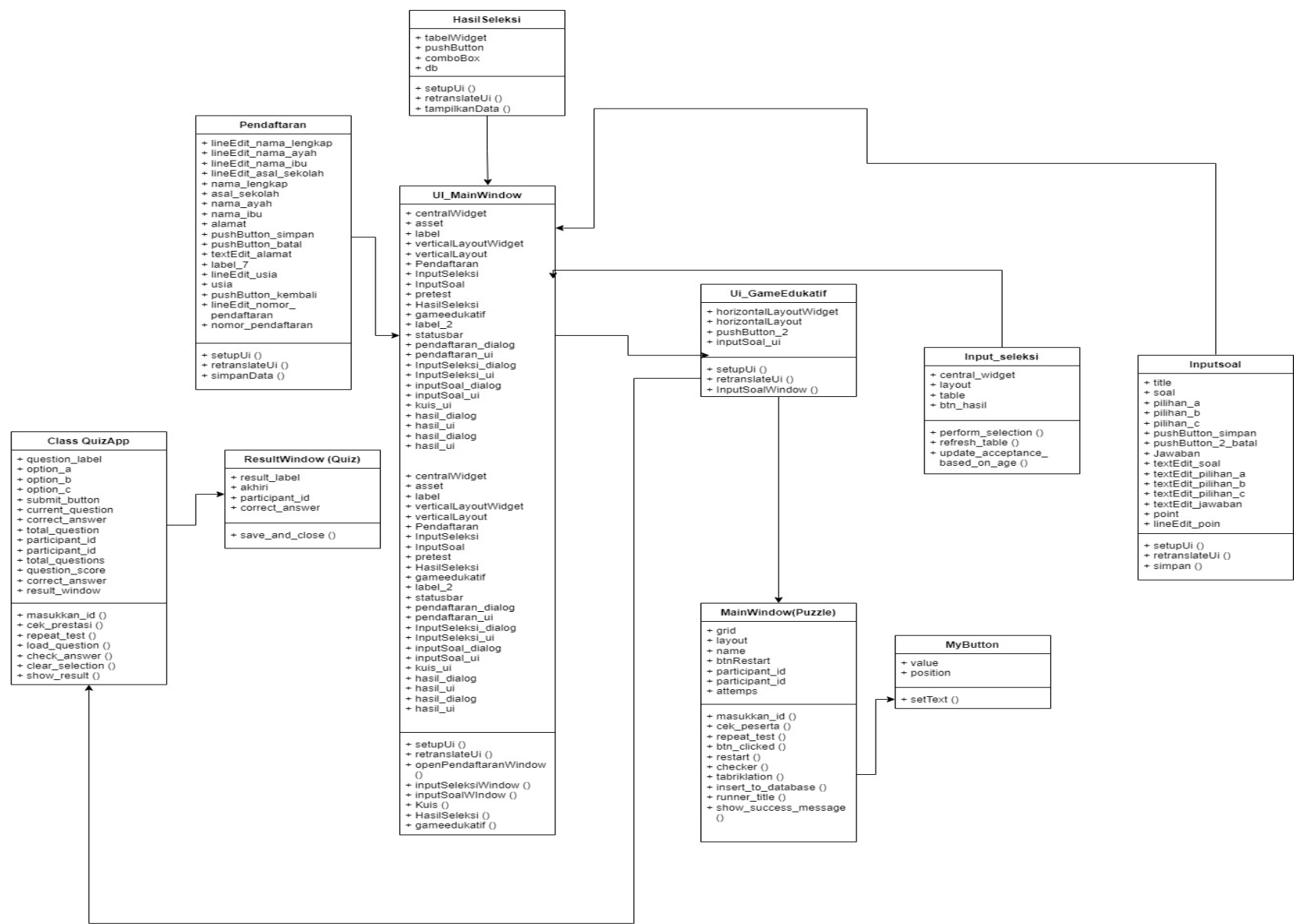
MySQL adalah salah satu jenis database server yang sangat terkenal. Kepopulerannya disebabkan Mysql menggunakan SQL sebagai bahasa dasar untuk mengakses databasenya. Mysql termasuk jenis RDBMS (Relational Database Management System) (Butler, T & Yank, K., 2016).

Flowchart dan Diagram Class

A. Flowchart



B. Class Diagram



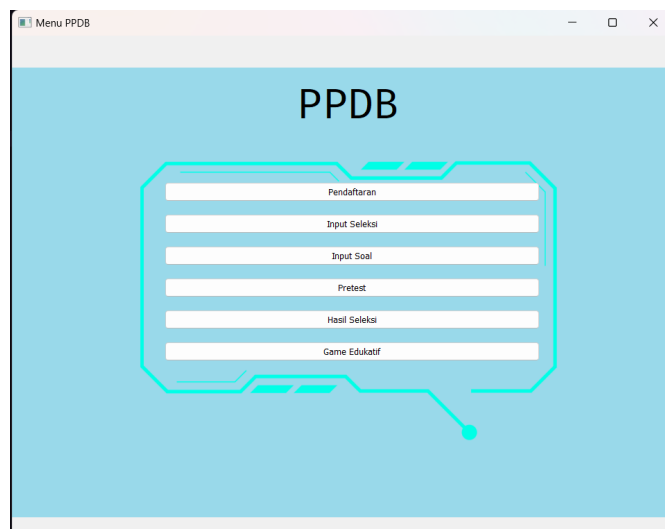
Hasil Pemrograman dan Pembahasan

📁 puzzle	02/01/2024 01:53	File folder	
📄 bg	25/12/2023 12:34	PNG File	5 KB
📄 gameedu	02/01/2024 15:12	Python Source File	2 KB
📄 hasilseleksi	19/12/2023 20:09	Python Source File	4 KB
📄 inputsoal	17/12/2023 22:30	Python Source File	6 KB
📄 kuis	26/12/2023 16:47	Python Source File	7 KB
📄 main	25/12/2023 12:00	PNG File	14 KB
📄 menu	25/12/2023 18:42	Python Source File	7 KB
📄 pendaftaran2	19/12/2023 22:21	Python Source File	8 KB
📄 percobaanis	19/12/2023 16:40	Python Source File	3 KB

Gambar 1

Dilihat dari gambar bahwa kami memisahkan menjadi beberapa file agar tidak terlalu banyak code dalam satu file, pemanggilan dari file tersebut dengan cara memanggil class yang ada dalam file tersebut.

Tampilan Utama:

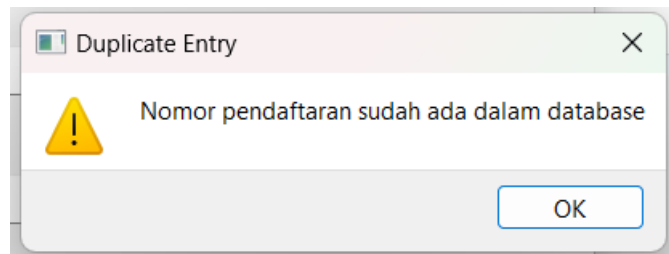


Dalam gambar disediakan beberapa menu dari pendaftaran hingga game edukatif yang isi nantinya berupa game puzzle.

Tampilan menu Pendaftaran:

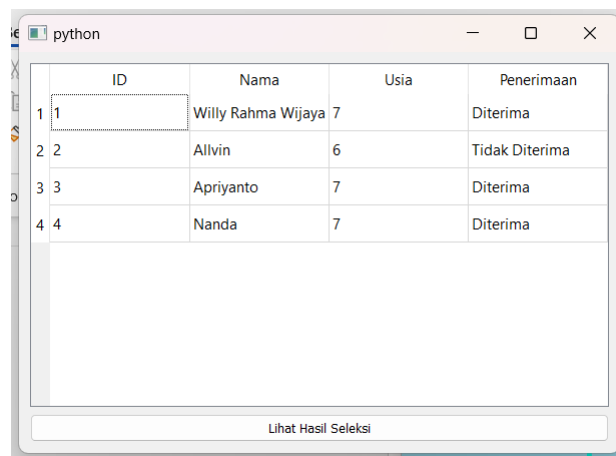


Tampilan pendaftaran seperti gambar diatas digunakan untuk mengisi pendaftaran siswa, jika nomor pendaftaran sudah tertera akan muncul message box seperti berikut



Tetapi jika nomor pendaftaran belum ada di database maka pendaftaran akan berhasil.

Tampilan menu input seleksi:

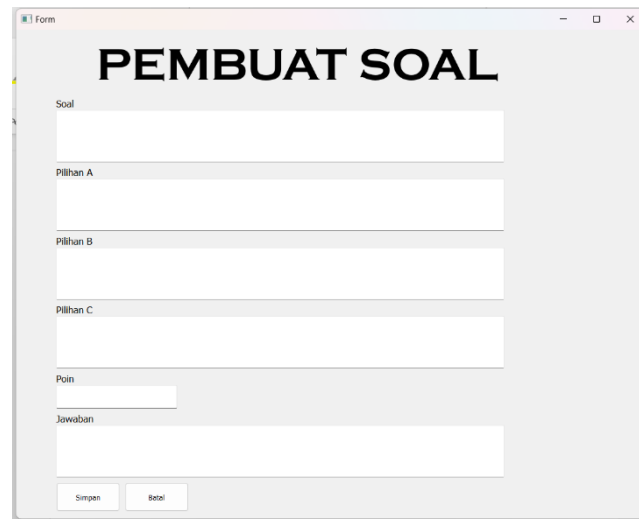


	ID	Nama	Usia	Penerimaan
1	1	Willy Rahma Wijaya	7	Diterima
2	2	Allvin	6	Tidak Diterima
3	3	Apriyanto	7	Diterima
4	4	Nanda	7	Diterima

Lihat Hasil Seleksi

Kami memilih otomatis dengan kriteria bahwa peserta dengan usia kurang dari 7 tahun akan otomatis tidak diterima.

Tampilan Menu Input Soal:

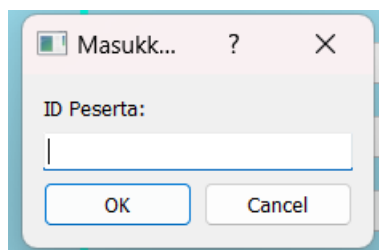


The screenshot shows a window titled 'Form' with the main heading 'PEMBUAT SOAL'. It contains several input fields: 'Soal' (Question), 'Pilihan A' (Option A), 'Pilihan B' (Option B), 'Pilihan C' (Option C), 'Poin' (Points), and 'Jawaban' (Answer). At the bottom, there are two buttons: 'Simpan' (Save) and 'Batal' (Cancel).

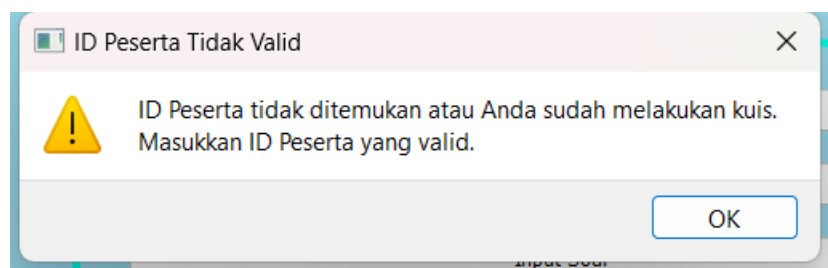
Di aplikasi ppdb juga ada kuis dengan input soal mandiri yang nantinya soal itu akan tersimpan di database dan akan terkoneksi dengan menu pretest.

Tampilan Menu Pretest:

Sebelum melakukan pretest peserta memasukkan id sesuai yang terdaftar di input hasil seleksi, jika peserta tidak terdaftar/sudah melakukan kuis akan muncul message box peringatan.

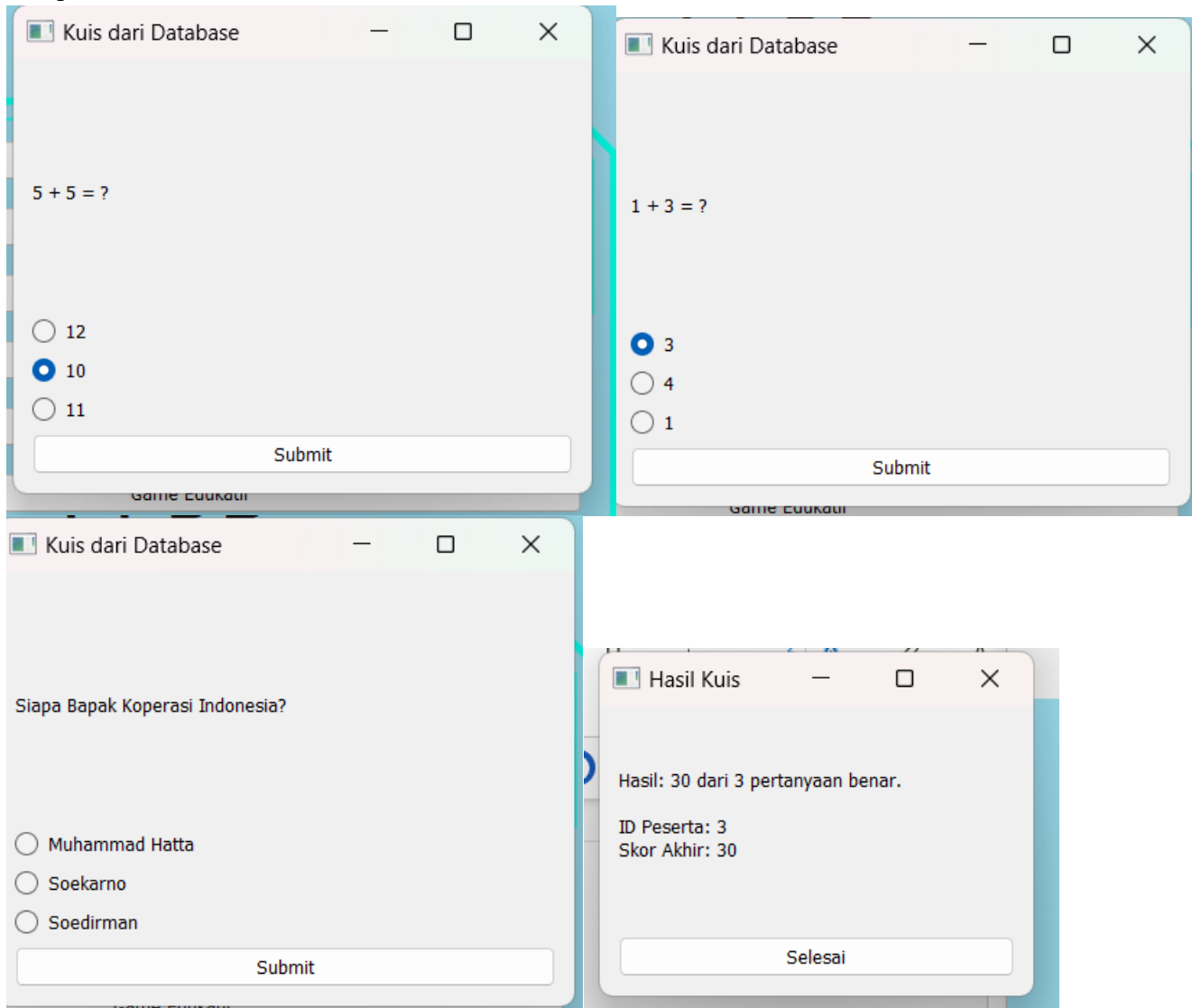


The screenshot shows a small dialog box titled 'Masukk...' with a question mark icon and a close button. It contains a label 'ID Peserta:' followed by an empty text input field. At the bottom, there are two buttons: 'OK' and 'Cancel'.



The screenshot shows an error message box titled 'ID Peserta Tidak Valid' with a close button. It features a yellow warning triangle icon and the text: 'ID Peserta tidak ditemukan atau Anda sudah melakukan kuis. Masukkan ID Peserta yang valid.' (Participant ID not found or you have already taken the quiz. Enter a valid Participant ID). There is an 'OK' button at the bottom right.

Tampilan kuis :



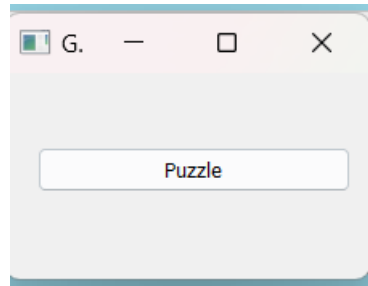
Tampilan Menu Hasil Seleksi setelah Kuis:

The screenshot shows a 'Dialog' window with a table displaying student selection results. The table has four columns: 'Nama Siswa', 'nama_ayah', 'Usia', and 'Skor'. There are two rows of data. Below the table is a 'Tampilkan' button.

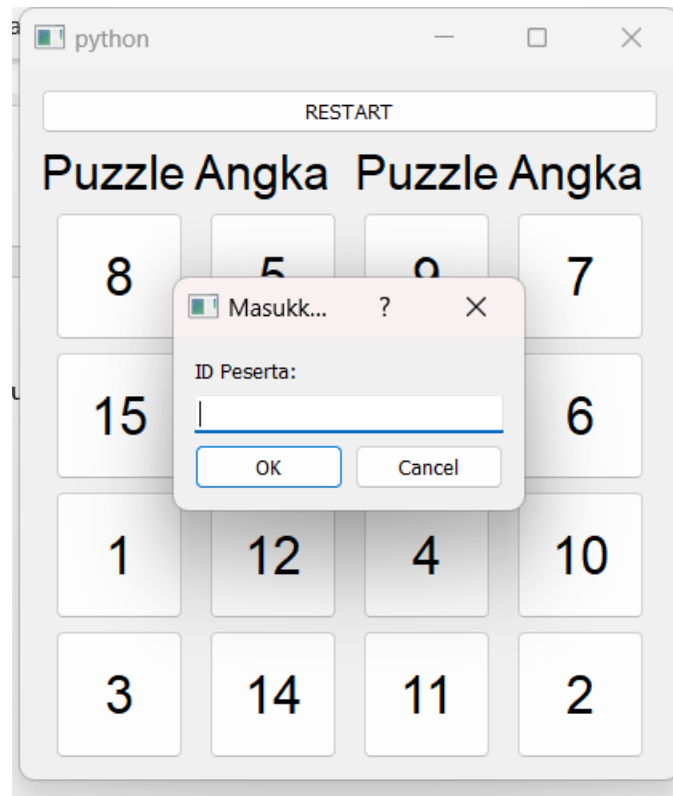
	Nama Siswa	nama_ayah	Usia	Skor
1	Willy Rahma Wijaya	Wii	7	20
2	Nanda	Alvin	7	30

Ketika siswa tersebut diterima tetapi belum memulai kuis maka tidak tampil.

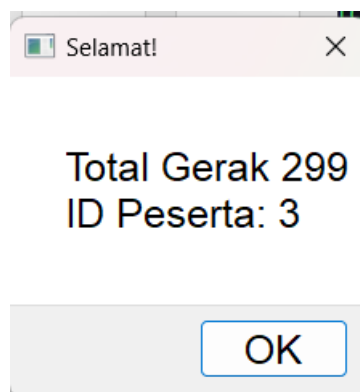
Tampilan Menu Game Edukatif:



Sebelum memulai peserta memasukkan id puzzle, ini bertujuan nantinya hasil puzzle akan disimpan didatabase



Ketika peserta berhasil akan muncul message box seperti ini



Penjelasan Setiap Class

Class Puzzle:

Puzzle memiliki 2 file yang pertama untuk awal yang kedua untuk button

Class pertama (file Pertama):

```
You, 17 hours ago | I author (You)
class MainWindow(QWidget):
    nametxt = " Puzzle Angka Puzzle Angka "
    numbers = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
    winner = numbers[1:] + numbers[:1]
    attempts = 0

    def __init__(self):
        random.shuffle(self.numbers)
        super().__init__()
        self.grid = QGridLayout()
        self.layout = QVBoxLayout()
        self.name = QLabel(self.nametxt)
        self.name.setFont(QFont('Arial', 18))
        self.btnRestart = QPushButton("RESTART")

        self.participant_id = None # Add this line to initialize participant_id

        self.buttons = [MyButton(i) for i in self.numbers]
        self.positions = [(i, j) for i in range(4) for j in range(4)]

        for btn, pos in zip(self.buttons, self.positions):
            btn.position = list(pos)
            self.grid.addWidget(btn, *pos)
            btn.clicked.connect(self.btn_clicked)

        self.btnRestart.clicked.connect(self.restart)
        threading.Thread(target=self.runner_title).start()

        self.layout.addWidget(self.btnRestart)
        self.layout.addWidget(self.name)
        self.layout.addLayout(self.grid)

        self.setLayout(self.layout)
        self.show()
        self.masukkan_id()

    def masukkan_id(self):
        while True:
            participant, ok_pressed = QInputDialog.getText(
                self, 'Masukkan ID Peserta', 'ID Peserta:')
            if ok_pressed:
                self.participant_id = participant
                if self.cek_peserta() and not self.repeat_test():
                    break
            else:
                QMessageBox.warning(
                    self, 'ID Peserta Tidak Valid', 'ID Peserta tidak ditemukan atau Anda sudah melakukan kuis. Masukkan ID Peserta yang valid.')
        else:
            sys.close()
```

Class puzzle memiliki nama MainWindow, isi dari class diatas adalah nametxt untuk judul, numbers yang berisi array untuk button winners untuk [1:] untuk mengambil seluruh angka ditambah dengan [:1] untuk menambah angka 0 yang lalu dijumlahkan dengan operasi penjumlahan, attempts digunakan untuk menghitung Langkah setiap gerak puzzle. Isi def init dimulai dari random.shuffle untuk mengacak sebuah angka yang ada di dalam puzzle. Grid untuk kotak/posisi untuk angka, untuk layout memakai vertikal untuk button restart, self button untuk mengatur posisi, MyButton sendiri merupakan class dari file button. Self participant digunakan untuk merekam sebuah id yang nanti keluar di akhir. Untuk def masukkan_id untuk memasukkan id peserta.

```

def cek_peserta(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    participant_id = self.participant_id

    cursor.execute(
        "SELECT COUNT(*) FROM status_kuis WHERE id_pendaftaran = %s", (participant_id,))
    count = cursor.fetchone()[0]

    conn.close()

    return count > 0

def repeat_test(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    participant_id = self.participant_id

    cursor.execute(
        "SELECT COUNT(*) FROM hasil_kuis WHERE id_pendaftaran = %s", (participant_id,))
    count = cursor.fetchone()[0]

    conn.close()

    return count > 0

```

Def cek peserta untuk mengecek apakah peserta sudah terdaftar/diterima, def repeat_test untuk mengecek peserta sudah melakukan puzzle atau belum.

```

def btn_clicked(self):
    self.attempts += 1
    # self.tabriklation()
    btn = self.sender()
    for i in self.buttons:
        if i.value == 0 and (i.position[0] == btn.position[0] + 1 or i.position[0] == btn.position[0] - 1) and i.position[1] == btn.position[1]:
            btn.value, i.value = i.value, btn.value
            btn.setText(btn.value == 0)
            i.setText(i.value == 0)
        elif i.value == 0 and (i.position[1] == btn.position[1] + 1 or i.position[1] == btn.position[1] - 1) and i.position[0] == btn.position[0]:
            btn.value, i.value = i.value, btn.value
            btn.setText(btn.value == 0)
            i.setText(i.value == 0)

    self.checker()

def restart(self):
    self.attempts = 0
    self.buttons.clear()
    random.shuffle(self.numbers)
    self.buttons = [MyButton(i) for i in self.numbers]
    for btn, pos in zip(self.buttons, self.positions):
        btn.position = list(pos)
        self.grid.addWidget(btn, *pos)
        btn.clicked.connect(self.btn_clicked)

def checker(self):
    count = 0
    if self.buttons[15].value == 0:
        for i in range(16):
            if self.buttons[i].value == self.winner[i]:
                count += 1
    if count >= 15:
        self.tabriklation()

```

Def btn_clicked, self attempts untuk menginkremen Langkah, self sender untuk menerima sinyal perpindahan gerak, for i in self.button setiap ada yang memenuhi kriteria sekitar ada yang kosong dengan button akan terjadi pertukaran jika tidak ada maka tidak dapat berpindah. Def restart digunakan untuk mereset button puzzle, self button clear untuk menghapus tampilan angka sebelumnya yang sebelum direset, random.shuffle untuk mengacak posisi puzzle, self button untuk membuat posisi angka, lalu di for untuk melakukan pengacakan. Def checker, count untuk menginisiasi sebuah score, if self.button[15] value == 0 untuk mengecek apakah Langkah terakhir ada button kosong

- ilustrasi (13,14,15, button kosong) jika true maka akan melakukan perulangan , if count >= 15 maka akan menjalankan fungsi tabriklation.

```
def tabriklation(self):
    dlg = QMessageBox()
    dlg.setWindowTitle("Selamat!")
    text = f"\nTotal Gerak {self.attempts}\nID Peserta: {
        self.participant_id}\n" # Include participant_id in the message
    dlg.setText(text)
    dlg.setFont(QFont('Arial', 15))
    dlg.setStandardButtons(QMessageBox.Ok | QMessageBox.Cancel)
    ret = dlg.exec_()

    if ret == QMessageBox.Ok:
        self.insert_to_database()
        self.masukkan_id()
    else:
        sys.exit()

def insert_to_database(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    query = "INSERT INTO hasil_puzzle (id_peserta, skor) VALUES (%s, %s)"
    values = (self.participant_id, self.attempts)

    cursor.execute(query, values)
    conn.commit()
    self.show_success_message()
    conn.close()

def runner_title(self):
    txt = self.nametxt
    ln = len(txt)
    while True:
        txt = txt[1:] + txt[:1]
        self.name.setText(txt)
        time.sleep(0.15)

def show_success_message(self):
    dlg = QMessageBox()
    dlg.setWindowTitle("Data Tersimpan")
    dlg.setText("Data telah berhasil disimpan ke dalam database.")
    dlg.setFont(QFont('Arial', 12))
    dlg.setStandardButtons(QMessageBox.Ok)
    dlg.exec_()
```

Def tabriklation merupakan akhir dari puzzle jika dapat menyelesaikan puzzle dengan akan muncul message box, Ketika diklik oke akan tersimpan di database , def insert_to_database untuk menyimpan/insert ke database, def runner title untuk untuk menjalankan running text, def show_succes_massage untuk menampilkan pesan bahwa sudah tersimpan di database.

Class MyButton (file ke 2):

```
class MyButton(QPushButton):
    def __init__(self, value, position=None):
        super().__init__()

        self.setFixedSize(QSize(80, 80))
        self.setFont(QFont('Arial', 20))
        self.value = value
        self.position = position
        self.setText(False)

    def setText(self, click: bool):
        if self.value == 0:
            super().setText(f'')
        else:
            super().setText(f'{self.value}')
```

Class my button untuk mengatur size ukuran button, font, value,posisi, text.

Class menu :

```
class Ui_MainWindow(object):
    def setupUi(self, MainWindow):
        MainWindow.setObjectName("MainWindow")
        MainWindow.resize(955, 721)
        self.centralwidget = QtWidgets.QWidget(MainWindow)
        self.centralwidget.setObjectName("centralwidget")
        self.asset2 = QtWidgets.QLabel(self.centralwidget)
        self.asset2.setGeometry(QtCore.QRect(100, 70, 771, 601))
        self.asset2.setStyleSheet("image: url(D:/PBO/qt/main.png);")
        self.asset2.setText("")
        self.asset2.setObjectName("asset2")
        self.label = QtWidgets.QLabel(self.centralwidget)
        self.label.setGeometry(QtCore.QRect(-410, -450, 1871, 1641))
        self.label.setStyleSheet(
            "image: url(D:/PBO/qt/bg.png);")
        self.label.setText("")
        self.label.setObjectName("label")
        self.verticalLayoutWidget = QtWidgets.QWidget(self.centralwidget)
        self.verticalLayoutWidget.setGeometry(QtCore.QRect(220, 200, 541, 281))
        self.verticalLayoutWidget.setObjectName("verticalLayoutWidget")
        self.verticalLayout = QtWidgets.QVBoxLayout(self.verticalLayoutWidget)
        self.verticalLayout.setContentsMargins(0, 0, 0, 0)
        self.verticalLayout.setObjectName("verticalLayout")
        self.Pendaftaran = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.Pendaftaran.setObjectName("Pendaftaran")
        self.verticalLayout.addWidget(self.Pendaftaran)
        self.InputSeleksi = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.InputSeleksi.setObjectName("InputSeleksi")
        self.verticalLayout.addWidget(self.InputSeleksi)
        self.InputSoal = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.InputSoal.setObjectName("InputSoal")
        self.verticalLayout.addWidget(self.InputSoal)
        self.Pretest = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.Pretest.setObjectName("Pretest")
        self.verticalLayout.addWidget(self.Pretest)
        self.hasilseleksi = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.hasilseleksi.setObjectName("hasilseleksi")
        self.verticalLayout.addWidget(self.hasilseleksi)
        self.gameedukatif = QtWidgets.QPushButton(self.verticalLayoutWidget)
        self.gameedukatif.setObjectName("gameedukatif")
        self.verticalLayout.addWidget(self.gameedukatif)
```

Untuk class menu ini untuk bagian menampilkan diawal sebagai user interface pertama.

Langsung saja ke button untuk pemanggilan:

```
def openPendaftaranWindow():
    self.pendaftaran_dialog = QtWidgets.QMainWindow()
    self.pendaftaran_ui = Pendaftaran()
    self.pendaftaran_ui.setupUi(self.pendaftaran_dialog)
    self.pendaftaran_dialog.show()

self.Pendaftaran.clicked.connect(openPendaftaranWindow)

def inputSeleksiWindow():
    self.inputSeleksi_dialog = QtWidgets.QMainWindow()
    self.inputSeleksi_ui = InputSeleksi()
    self.inputSeleksi_ui.show()

self.InputSeleksi.clicked.connect(inputSeleksiWindow)

def InputSoalWindow():
    self.inputSoal_dialog = QtWidgets.QMainWindow()
    self.inputSoal_ui = InputSoal()
    self.inputSoal_ui.setupUi(self.inputSoal_dialog)
    self.inputSoal_dialog.show()

self.InputSoal.clicked.connect(InputSoalWindow)

def Kuis():
    self.kuis_ui = QuizApp()
    self.kuis_ui.show()

self.Pretest.clicked.connect(Kuis)

def Hasilseleksi():
    self.hasil_dialog = QtWidgets.QMainWindow()
    self.hasil_ui = HasilSeleksi()
    self.hasil_ui.setupUi(self.hasil_dialog)
    self.hasil_dialog.show()

self.hasilseleksi.clicked.connect(Hasilseleksi)

def gameedukatif():
    self.hasil_dialog = QtWidgets.QMainWindow()
    self.hasil_ui = Ui_GameEdukatif()
    self.hasil_ui.setupUi(self.hasil_dialog)
    self.hasil_dialog.show()

self.gameedukatif.clicked.connect(gameedukatif)
```

Setiap button yang diklik akan muncul sesuai tampilannya sesuai class yang dipanggil.


```

1  class InputSeleksi(QMainWindow):
2      def __init__(self):
3          super().__init__()
4
5          self.setObjectName('Penerimaan Siswa')
6          self.setGeometry(100, 100, 600, 400)
7
8          self.central_widget = QWidget()
9          self.setCentralWidget(self.central_widget)
10
11         self.layout = QVBoxLayout()
12         self.central_widget.setLayout(self.layout)
13
14         self.table = QTableWidgetItem()
15         self.table.setColumnCount(4)
16         self.table.setHorizontalHeaderLabels(
17             ['ID', 'Nama', 'Usia', 'Penerimaan'])
18         self.table.horizontalHeader().setSectionResizeMode(QHeaderView.Stretch)
19         self.layout.addWidget(self.table)
20
21         self.btn_hasil = QPushButton('Lihat Hasil Seleksi')
22         self.btn_hasil.clicked.connect(self.perform_selection)
23         self.layout.addWidget(self.btn_hasil)
24
25         self.refresh_table()
26
27     def perform_selection(self):
28         self.update_acceptance_based_on_age()
29         self.refresh_table()
30         QMessageBox.information(
31             self, 'Info', 'Hasil seleksi telah diperbarui!')
32
33     def refresh_table(self):
34         self.table.setRowCount(0)
35
36         mydb = mysql.connector.connect(
37             host="localhost",
38             user="root",
39             password="",
40             database="ppdb"
41         )
42
43         mycursor = mydb.cursor()
44
45         mycursor.execute(
46             "SELECT id_pendaftaran, nama_siswa, usia, status_penerimaan FROM pendaftaran")
47         data = mycursor.fetchall()
48
49         for row_number, row_data in enumerate(data):
50             self.table.insertRow(row_number)
51             for column_number, column_data in enumerate(row_data):
52                 item = QTableWidgetItem(str(column_data))
53                 self.table.setItem(row_number, column_number, item)
54
55         mycursor.close()
56
57     def update_acceptance_based_on_age(self):
58         mydb = mysql.connector.connect(
59             host="localhost",
60             user="root",
61             password="",
62             database="ppdb"
63         )
64
65         mycursor = mydb.cursor()
66
67         mycursor.execute(
68             "SELECT id_pendaftaran, usia FROM pendaftaran")
69         data = mycursor.fetchall()
70
71         for row_data in data:
72             id_pendaftaran = row_data[0]
73             usia = int(row_data[1])
74
75             if usia < 7:
76                 status_penerimaan = 'Tidak Diterima'
77             else:
78                 status_penerimaan = 'Diterima'
79
80             mycursor.execute("UPDATE pendaftaran SET status_penerimaan = %s WHERE id_pendaftaran = %s",
81                             (status_penerimaan, id_pendaftaran))
82
83         mydb.commit()
84         mycursor.close()
85
86
87     if __name__ == '__main__':
88         app = QApplication(sys.argv)
89         ui = InputSeleksi()
90         ui.show()
91         sys.exit(app.exec_())
92

```

(Penjelasan Pada InputSeleksi)

- Input seleksi merupakan subclass dari QMainWindow yang merupakan perwakilan dari jendela utama aplikasi
- Selanjutnya fungsi untuk menetapkan judul ukuran dan tata letak input seleksi pada jendela utama
- Selanjutnya QTableWidgetItem digunakan untuk membuat kolom ID, Nama, Usia, dan Penerimaan
- Selanjutnya QPushButton digunakan untuk menambahkan tombol push dengan judul Lihat Hasil Seleksi dan dihubungkan sinyal kliknya dengan metode perform_selection
- Selanjutnya fungsi refres_table digunakan untuk menghapus baris yang ada dalam tabel, dan digunakan untuk mengconnect codingan ke database dan digunakan untuk mengambil data dari tabel pendaftaran.

Selanjutnya Fungsi update_acceptance_based_on_age. Fungsi yang terhubung dengan database dan mengambil ID dan usia para pelamar, didalamnya juga digunakan untuk menentukan status penerimaan untuk para pelamar berdasarkan usianya, jika kurang dari 7 tahun tidak diterima, dan jika sebaliknya maka akan diterima.

```
class HasilSeleksi(object):
    def __init__(self, Dialog):
        Dialog.setObjectName("Dialog")
        Dialog.resize(1121, 842)
        self.pushButton = QPushButton(Dialog)
        self.pushButton.setObjectName("pushButton")
        self.pushButton.setGeometry(QRect(440, 241, 51, 31))
        self.comboBox = QComboBox(Dialog)
        self.comboBox.setObjectName("comboBox")
        self.comboBox.setGeometry(QRect(312, 241, 31, 31))
        self.comboBox.addItem("")
        self.comboBox.addItem("")
        self.pushButton.clicked.connect(self.tampilkanData)
        self.comboBox.currentIndexChanged.connect(self.tampilkanData)

    def retranslateUi(self, Dialog):
        Dialog.setWindowTitle(Dialog.tr("Dialog"))
        self.pushButton.setText(Dialog.tr("Lihat Hasil Seleksi"))
        self.comboBox.setText(Dialog.tr("Diterima"))
        self.comboBox.setItemText(1, Dialog.tr("Tidak Diterima"))

    def tampilkanData(self):
        try:
            self.db = mysql.connector.connect(
                host="localhost",
                user="root",
                password="",
                database="pmb"
            )
            print("koneksi ke database berhasil")
        except mysql.connector.Error as e:
            print(f"Gagal koneksi ke database: {e}")

        status = self.comboBox.currentText()
        cursor = self.db.cursor()

        if status == "Diterima":
            query = "SELECT nama_siswa, nama_ayah, usia, skor FROM pendaftaran INNER JOIN hasil_kuis ON pendaftaran.id_pendaftaran = hasil_kuis.id_pendaftaran WHERE status_penerimaan='Diterima'"
        else:
            query = "SELECT nama_siswa, nama_ayah, usia, skor FROM pendaftaran INNER JOIN hasil_kuis ON pendaftaran.id_pendaftaran = hasil_kuis.id_pendaftaran WHERE status_penerimaan='Tidak Diterima'"

        cursor.execute(query)

        # Menghapus semua baris yang ada di tabel
        self.tableWidget.setRowCount(0)

        # Mengambil data dari database ke QTableWidgetItem
        for row_index, row_data in enumerate(cursor.fetchall()):
            self.tableWidget.insertRow(row_index)
            for col_index, col_data in enumerate(row_data):
                item = QTableWidgetItem(str(col_data))
                self.tableWidget.setItem(row_index, col_index, item)

        cursor.close()

if __name__ == '__main__':
    import sys
    app = QApplication(sys.argv)
    Dialog = QDialog()
    ui = HasilSeleksi(Dialog)
    ui.setupUi(Dialog)
    Dialog.show()
    sys.exit(app.exec_())
```

(Penjelasan Pada Hasil Seleksi)

- Class HasilSeleksi yang mewakili antarmuka pengguna dan fungsi untuk menampilkan hasil seleksi, didalamnya terdapat fungsi setupUi untuk menginisialisasi elemen GUI seperti tabel QTableWidgetItem, tombol QPushButton, dan menu dropdown (QComboBox). Didalam fungsi ini juga digunakan untuk menetapkan label kolom pada tabel, mengatur tata letak elemen-elemen GUI
- Selanjutnya ada Fungsi retranslateUi yang digunakan untuk menetapkan teks pada elemen GUI seperti halnya Judul jendelanya, teks tombol, dan opsi dropdown
- Selanjutnya ada Fungsi tampilkanData yang digunakan untuk menghubungkan code dengan database, jika berhasil metode ini mendapatkan status penerimaan yang dipilih dari

dropdown, Berdasarkan pada status buat query untuk mengambil data dari tabel pendaftaran, mengambil data dari database menggunakan cursor dan memasukkan nya ke dalam tableWidget

Kuis dan ClassKuis

```
7  class QuizApp(QWidget):
8      def __init__(self):
9          super().__init__()
10
11         self.setWindowTitle('Kuis dari Database')
12         self.setGeometry(100, 100, 400, 300)
13
14         self.question_label = QLabel()
15         self.option_a = QRadioButton()
16         self.option_b = QRadioButton()
17         self.option_c = QRadioButton()
18         self.submit_button = QPushButton('Submit')
19
20         layout = QVBoxLayout()
21         layout.addWidget(self.question_label)
22         layout.addWidget(self.option_a)
23         layout.addWidget(self.option_b)
24         layout.addWidget(self.option_c)
25         layout.addWidget(self.submit_button)
26
27         self.setLayout(layout)
28
29         self.submit_button.clicked.connect(self.check_answer)
30
31         self.current_question = 0
32         self.correct_answers = 0
33         self.total_questions = 0
34         self.participant_id = None
35
36         self.masukkan_id()
37         self.load_question()
```

```

def masukan_id(self):
    while True:
        participant, ok_pressed = QInputDialog.getText(
            self, 'Masukkan ID Peserta', 'ID Peserta:')
        if ok_pressed:
            self.participant_id = participant
            if self.cek_peserta() and not self.repeat_test():
                break
            else:
                QMessageBox.warning(
                    self, 'ID Peserta Tidak Valid', 'ID Peserta tidak ditemukan atau Anda sudah melakukan kuis. Masukkan ID Peserta yang valid.')
        else:
            sys.close()

def cek_peserta(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    participant_id = self.participant_id

    cursor.execute(
        "SELECT COUNT(*) FROM status_kuis WHERE id_pendaftaran = %s", (participant_id,))
    count = cursor.fetchone()[0]

    conn.close()

    return count > 0

def repeat_test(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    participant_id = self.participant_id

    cursor.execute(
        "SELECT COUNT(*) FROM hasil_kuis WHERE id_pendaftaran = %s", (participant_id,))
    count = cursor.fetchone()[0]

    conn.close()

    return count > 0

def load_question(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM pertanyaan")
    questions = cursor.fetchall()
    self.total_questions = len(questions)

    if self.current_question < self.total_questions:
        question, question_score = questions[self.current_question][1], questions[self.current_question][5]
        self.question_label.setText(question)
        self.option_a.setText(questions[self.current_question][2])
        self.option_b.setText(questions[self.current_question][3])
        self.option_c.setText(questions[self.current_question][4])
        self.question_score = question_score
        self.correct_answer = questions[self.current_question][6]

```

```

        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    participant_id = self.participant_id

    cursor.execute(
        "SELECT COUNT(*) FROM hasil_kuis WHERE id_pendaftaran = %s", (participant_id,))
    count = cursor.fetchone()[0]

    conn.close()

    return count > 0

def load_question(self):
    conn = mysql.connector.connect(
        host='localhost',
        user='root',
        password='',
        database='ppdb'
    )
    cursor = conn.cursor()

    cursor.execute("SELECT * FROM pertanyaan")
    questions = cursor.fetchall()
    self.total_questions = len(questions)

    if self.current_question < self.total_questions:
        question, question_score = questions[self.current_question][1], questions[self.current_question][5]
        self.question_label.setText(question)
        self.option_a.setText(questions[self.current_question][2])
        self.option_b.setText(questions[self.current_question][3])
        self.option_c.setText(questions[self.current_question][4])
        self.question_score = question_score
        self.correct_answer = questions[self.current_question][6]

```

```

        else:
            self.show_result()

        conn.close()

    def check_answer(self):
        answer = ''
        if self.option_a.isChecked():
            answer = self.option_a.text()
        elif self.option_b.isChecked():
            answer = self.option_b.text()
        elif self.option_c.isChecked():
            answer = self.option_c.text()

        correct_answer = self.correct_answer
        question_score = self.question_score

        if answer == correct_answer:
            self.correct_answers += question_score

        self.current_question += 1

        if self.current_question == self.total_questions:
            self.show_result()
        else:
            self.clear_selection()
            self.load_question()

    def clear_selection(self):
        self.option_a.setChecked(False)
        self.option_b.setChecked(False)
        self.option_c.setChecked(False)

    def show_result(self):
        result_text = f'Hasil: {self.correct_answers} dari {
            self.total_questions} pertanyaan benar.\n\n'
        result_text += f'ID Peserta: {self.participant_id}\n'

```

```

def show_result(self):
    result_text = f'Hasil: {self.correct_answers} dari {
        self.total_questions} pertanyaan benar.\n\n'
    result_text += f'ID Peserta: {self.participant_id}\n'
    result_text += f'Skor Akhir: {self.correct_answers}\n'

    self.result_window = ResultWindow(
        result_text, self.participant_id, self.correct_answers)
    self.result_window.show()
    self.close()

```

Class ini berfungsi untuk menampilkan sebuah kuis dimana didalamnya terdapat self.submit_button.clicked.connect untuk menghubungkan dengan class lain. Ada variable self.current_question, self corrct_answers, self.total_questions dimana untuk membuat sebuah soal,pilihan, poin, dan sebuah jawaban. Untuk awalan adanya self.masukkan_id untuk memasukkan id peserta lalu self.load_question untuk memuat id yang sudah dimasukkan . Selanjutnya ada

cek_peserta untuk mengetahui status peserta, ada repeat_test untuk mengulang sebuah kuis, load_question menyimpan sebuah question yang telah dijawab, kemudian check_answer dimana jawaban yang telah dijawab dicheck kebenarannya.

```
1 import sys
2 import mysql.connector
3 from PyQt5.QtWidgets import QApplication, QWidget, QVBoxLayout, QLabel, QRadioButton, QPushButton, QMessageBox, QInputDialog
4 from PyQt5 import QtWidgets
5
6
7 class ResultWindow(QWidget):
8     def __init__(self, result_text, participant_id, correct_answers):
9         super().__init__()
10
11         self.setWindowTitle('Hasil Kuis')
12         self.setGeometry(200, 200, 300, 200)
13
14         self.result_label = QLabel(result_text)
15         self.akhiri = QPushButton('Selesai')
16
17         self.participant_id = participant_id
18         self.correct_answers = correct_answers
19
20         self.akhiri.clicked.connect(self.save_and_close)
21
22         layout = QVBoxLayout()
23         layout.addWidget(self.result_label)
24         layout.addWidget(self.akhiri)
25         self.setLayout(layout)
26
27     def save_and_close(self):
28         conn = mysql.connector.connect(
29             host='localhost',
30             user='root',
31             password='',
32             database='ppdb'
33         )
34         cursor = conn.cursor()
35
36         insert_query = "INSERT INTO hasil_kuis (id_pendaftaran, skor) VALUES (%s, %s)"
37         cursor.execute(
```

Class ResultWindow sebuah kelas untuk mengakhiri hasilkuis dan menyimpan data yang sudah dibuat

Class Game Edukatif:

```
class Ui_GameEdukatif(object):
    def setupUi(self, GameEdukatif):
        GameEdukatif.setObjectName("GameEdukatif")
        GameEdukatif.resize(227, 128)
        self.horizontalLayoutWidget = QtWidgets.QWidget(GameEdukatif)
        self.horizontalLayoutWidget.setGeometry(QtCore.QRect(20, 20, 195, 80))
        self.horizontalLayoutWidget.setObjectName("horizontalLayoutWidget")
        self.horizontalLayout = QtWidgets.QHBoxLayout(
            self.horizontalLayoutWidget)
        self.horizontalLayout.setContentsMargins(0, 0, 0, 0)
        self.horizontalLayout.setObjectName("horizontalLayout")
        self.pushButton_2 = QtWidgets.QPushButton(self.horizontalLayoutWidget)
        self.pushButton_2.setObjectName("pushButton_2")
        self.horizontalLayout.addWidget(self.pushButton_2)

        self.retranslateUi(GameEdukatif)
        QtCore.QMetaObject.connectSlotsByName(GameEdukatif)

    def retranslateUi(self, GameEdukatif):
        _translate = QtCore.QCoreApplication.translate
        GameEdukatif.setWindowTitle(
            _translate("GameEdukatif", "Game Edukatif"))
        self.pushButton_2.setText(_translate("GameEdukatif", "Puzzle"))

    def InputSoalWindow():
        self.inputSoal_ui = MainWindow()

    self.pushButton_2.clicked.connect(InputSoalWindow)
```

Untuk menampilkan button dan mengarah ke game puzzle

```

class InputSoal(object):
    def setupUi(self, Form):
        Form.setObjectName("Form")
        Form.resize(1101, 853)
        self.title = QtWidgets.QLabel(Form)
        self.title.setGeometry(QtCore.QRect(140, 30, 721, 61))
        font = QtGui.QFont()
        font.setFamily("Copperplate Gothic Bold")
        font.setPointSize(48)
        self.title.setFont(font)
        self.title.setObjectName("title")
        self.soal = QtWidgets.QLabel(Form)
        self.soal.setGeometry(QtCore.QRect(70, 120, 55, 16))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.soal.setFont(font)
        self.soal.setObjectName("soal")
        self.pilihan_a = QtWidgets.QLabel(Form)
        self.pilihan_a.setGeometry(QtCore.QRect(70, 240, 71, 16))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.pilihan_a.setFont(font)
        self.pilihan_a.setObjectName("pilihan_a")
        self.pilihan_b = QtWidgets.QLabel(Form)
        self.pilihan_b.setGeometry(QtCore.QRect(70, 360, 71, 16))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.pilihan_b.setFont(font)
        self.pilihan_b.setObjectName("pilihan_b")
        self.pilihan_c = QtWidgets.QLabel(Form)
        self.pilihan_c.setGeometry(QtCore.QRect(70, 480, 71, 16))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.pilihan_c.setFont(font)
        self.pilihan_c.setObjectName("pilihan_c")
        self.pushButton_simpan = QtWidgets.QPushButton(Form)
        self.pushButton_simpan.setGeometry(QtCore.QRect(70, 790, 111, 51))
        self.pushButton_simpan.setObjectName("pushButton_simpan")
        self.pushButton_2_batal = QtWidgets.QPushButton(Form)
        self.pushButton_2_batal.setGeometry(QtCore.QRect(190, 790, 111, 51))
        self.pushButton_2_batal.setObjectName("pushButton_2_batal")
        self.Jawaban = QtWidgets.QLabel(Form)
        self.Jawaban.setGeometry(QtCore.QRect(70, 670, 71, 16))
        font = QtGui.QFont()
        font.setPointSize(10)
        self.Jawaban.setFont(font)
        self.Jawaban.setObjectName("Jawaban")
        self.textEdit_soal = QtWidgets.QTextEdit(Form)
        self.textEdit_soal.setGeometry(QtCore.QRect(70, 140, 781, 91))
        self.textEdit_soal.setObjectName("textEdit_soal")
        self.textEdit_pilihan_a = QtWidgets.QTextEdit(Form)
        self.textEdit_pilihan_a.setGeometry(QtCore.QRect(70, 260, 781, 91))
        self.textEdit_pilihan_a.setObjectName("textEdit_pilihan_a")
        self.textEdit_pilihan_b = QtWidgets.QTextEdit(Form)

```

Class Input Soal untuk memasukkan soal yang akan dipersiapkan untuk tes kuis peserta

```

def simpan(self):
    question = self.textEdit_soal.toPlainText()
    option_a = self.textEdit_pilihan_a.toPlainText()
    option_b = self.textEdit_pilihan_b.toPlainText()
    option_c = self.textEdit_pilihan_c.toPlainText()
    point = self.lineEdit_poin.text()
    correct_answer = self.textEdit_jawaban.toPlainText()

    try:
        conn = mysql.connector.connect(
            host='localhost',
            user='root',
            password='',
            database='ppdb'
        )
        cursor = conn.cursor()

        query = "INSERT INTO pertanyaan (pertanyaan, pilihan_a, pilihan_b, pilihan_c, poin, jawaban) VALUES (%s, %s, %s, %s, %s, %s)"
        data = (question, option_a, option_b,
                option_c, point, correct_answer)

        cursor.execute(query, data)
        conn.commit()
        QMessageBox.information(
            None, 'Sukses', 'Pertanyaan berhasil dimasukkan ke dalam database')

        conn.close()
    except mysql.connector.Error as error:
        QMessageBox.critical(
            None, 'Error', f'Gagal memasukkan pertanyaan ke dalam database: {error}')

```

Def simpan untuk ketika sudah soal dan jawaban diinputkan ke proram dan klik simpan maka akan tersimpan di database

Daftar Pustaka

Butler, T & Yank, K. PHP & MySQL: Novice to Ninja, 6th Edition. SiitePoing. 2016.

Nosrati, M., 2011. *Python: An appropriate language for real world programming*. World Applied Programming Vol. 1 No.2 Hal. 110-117.

Willman, Joshua M., 2020. Beginning PyQt (A Hands-on Approach to GUI Programming). doi:10.1007/978-1-4842-5857-6.