

Perancangan Sistem *Login* Pada Aplikasi Berbasis GUI Menggunakan QtDesigner Python

Dias Ayu Budi Utami

Universitas AMIKOM Purwokerto

dias@amikompurwokerto.ac.id

Abstract

Advances in technology are increasingly making applications have increasingly sophisticated quality. Various programming languages also take part in building various applications with various types of platforms. One type of desktop application. Currently, desktop platform applications can be built using a variety of programming languages. One of them is the python programming language. The python programming language has several advantages, namely: the python programming language is simple and flexible so that it is easy to develop (reusable), desktop applications generated using python are dynamic and event-based which makes the application responsive. Making desktop applications using python is easier with the QtDesigner, namely classes in the python programming language that are event handlers that are suitable for desktop applications. One example of a GUI application that will be created using QtDesigner based on the python programming language is the login system in the application.

Keywords: *Login; GUI; Python; Event Handler; QtDesigner*

Abstrak

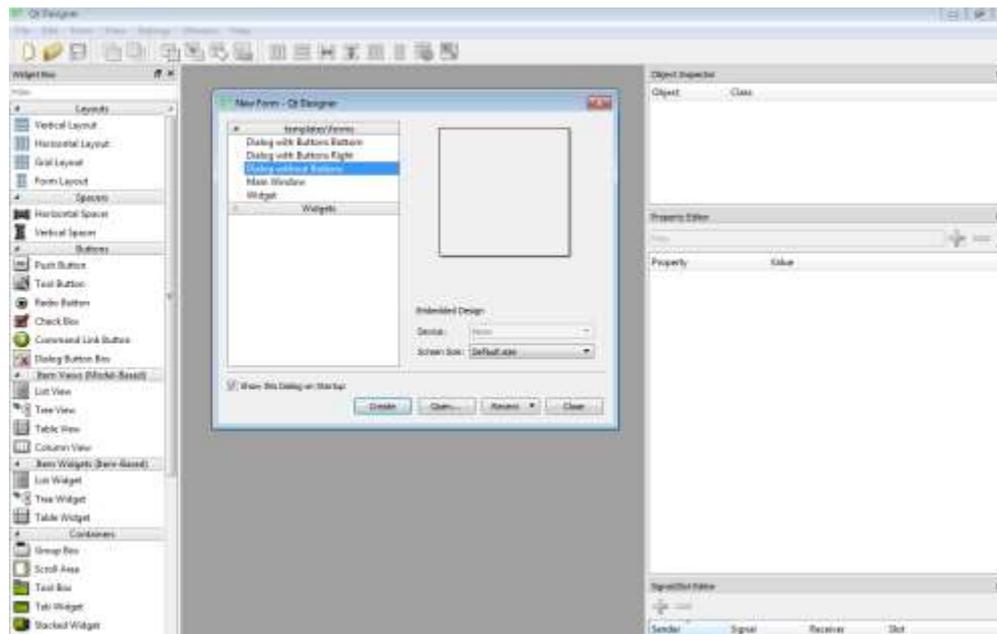
Kemajuan Teknologi semakin membuat aplikasi mempunyai kualitas yang semakin canggih. Berbagai bahasa pemrograman turut mengambil peran dalam membangun berbagai aplikasi dengan jenis platform yang beragam. Salah satunya jenis aplikasi GUI. Saat ini, aplikasi berplatform GUI (Graphical User Interface) dapat dibangun menggunakan berbagai macam bahasa pemrograman. Salah satunya yaitu bahasa pemrograman python. Bahasa pemrograman python memiliki beberapa keunggulan yaitu : bahasa pemrograman python bersifat sederhana dan fleksibel sehingga mudah dilakukan pengembangan (reusable), aplikasi GUI yang dihasilkan dengan menggunakan QtDesigner pada python bersifat dinamis dan berbasis event yang membuat aplikasi menjadi responsif. Pembuatan aplikasi GUI menggunakan python semakin mudah dengan adanya QtDesigner yaitu kelas-kelas dalam bahasa pemrograman python yang bersifat event handler yang cocok untuk aplikasi GUI. Salah satu contoh aplikasi GUI yang akan dibuat menggunakan QtDesigner berbasis bahasa pemrograman python adalah sistem *login* pada aplikasi.

Kata Kunci: *Login; GUI; Python; Event Handler; QtDesigner*

1. PENDAHULUAN

Semakin banyak bahasa pemrograman untuk membuat aplikasi GUI. Salah satunya yaitu bahasa pemrograman python. Keunggulan menggunakan bahasa pemrograman python yaitu bersifat sederhana dan fleksibel sehingga mudah dilakukan pengembangan (*reusable*), aplikasi GUI yang dihasilkan dengan menggunakan QtDesigner pada python bersifat dinamis dan berbasis event yang membuat aplikasi menjadi responsif. Pembuatan aplikasi GUI menggunakan python semakin mudah dengan adanya QtDesigner yaitu kelas-kelas dalam bahasa pemrograman python yang bersifat event handler yang cocok untuk aplikasi GUI.

QtDesigner merupakan perangkat perancangan visual yang dapat dipakai untuk menciptakan GUI dengan cepat dengan teknik *drag* dan *drop* tanpa perlu menulis satu baris kode. QtDesigner memiliki beberapa widget dengan level yang berbeda yaitu QDialog, QWidget dan QMainWindow. Berikut tampilan halaman QtDesigner dapat dilihat seperti gambar 1.



Gambar 1. Tampilan QTDesigner

Setiap aplikasi GUI memiliki *widget* level-atas dan sejumlah *widget* sebagai *widget* anaknya. *Widget* level-atas dapat berupa `QDialog`, `QWidget`, atau `QMainWindow`, tergantung dari template yang Anda perlukan. Jika Anda ingin menciptakan sebuah aplikasi berdasar template dialog, maka *widget* level-atas yang akan diwarisi adalah `QDialog`. Sama halnya, untuk menciptakan sebuah aplikasi berdasar *template* Main Window, *widget* level-atas akan berupa `QMainWindow`, dan untuk menciptakan aplikasi berdasar template *widget*, Anda perlu mewarisi kelas `QWidget`. Seperti yang telah disebutkan, *widget-widjet* lain yang dipakai untuk GUI dinamakan dengan *widget* anak. Dengan menggunakan QTDesigner maka akan mempermudah pembuatan aplikasi *login* yang berbasis GUI dengan menggunakan bahasa pemrograman python.

2. KERANGKA TEORI

2.1. 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).

2.2. QTDesigner

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).

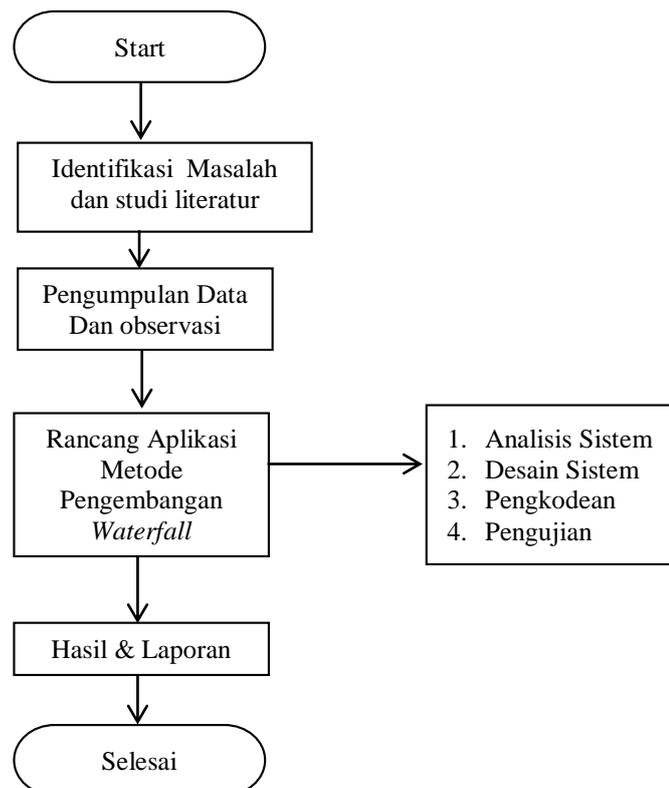
2.3. Python

Python adalah bahasa pemrograman yang fleksibel dan sederhana yang didefinisikan dalam dokumen-dokumennya sebagai berikut (Nosrati, 2011). Python adalah sebuah bahasa pemrograman dinamis yang sering digunakan dalam pengembangan aplikasi pada berbagai domain. Hal ini memungkinkan suatu program ditulis dalam beberapa pendekatan sekaligus. Misalnya, antar muka grafis dibuat dalam bentuk orientasi objek, sedangkan pemrosesan dalam bentuk fungsional atau prosedural. Bahasa pemrograman Python memiliki beberapa fitur yang dapat digunakan oleh pengembang perangkat lunak. Berikut adalah beberapa fitur yang ada pada bahasa pemrograman Python (Lutz, 2010):

1. Multi Paradigm Design
2. Open Source
3. Simplicity
4. Library Support
5. Portability
6. Extendable
7. Scalability

3. METODOLOGI

Konsep penelitian mempunyai beberapa tahap dan menggunakan metode pengembangan sistem seperti yang tertera pada gambar 2.



Gambar 2. Metode Pengembangan Sistem

a. Identifikasi Masalah dan Studi Literatur

Dalam penelitian hal pertama yang dilakukan adalah mengidentifikasi masalah. Identifikasi masalah adalah proses penelitian yang penting karena dapat menentukan kualitas penelitian yang dilakukan dan untuk merumuskan permasalahan yang akan menjadi latar belakang dalam objek penelitian yang dilakukan. Masalah yang diidentifikasi adalah bagaimana membuat aplikasi *login* menggunakan QTDesigner.

b. Pengumpulan Data dan Observasi

Setelah menyelesaikan identifikasi masalah lalu melanjutkan ke tahap berikutnya yaitu pengumpulan data. Pengumpulan data ini dilakukan dengan cara studi pustaka yaitu mengumpulkan dan mempelajari dari buku – buku / jurnal yang ada agar dapat menemukan masalah yang sedang diteliti dan mengobservasi langsung melalui pengamatan model-model aplikasi *login* yang sering digunakan.

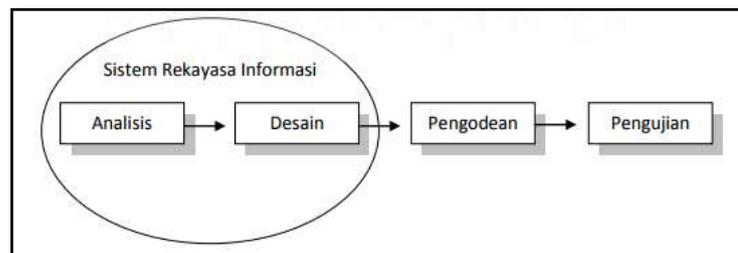
c. Perancangan Aplikasi dengan Metode Pengembangan *Waterfall*

Tahapan ini dimulai dari analisis sistem, desain sistem, pengkodean dan pengujian.

d. Hasil & Laporan

Setelah melakukan tahap – tahap diatas maka tahap yang terakhir dalam penelitian mendapatkan sebuah hasil dari menyelesaikan sebuah masalah yang diteliti yaitu pembuatan aplikasi *login* berbasis GUI menggunakan QTDesigner.

Metode yang digunakan dalam penelitian ini adalah metode *waterfall*. Metode *waterfall* termasuk metode penelitian yang klasik biasa disebut juga dengan *life cycle classic*. Pada pengembangan penulis menggunakan metode Air terjun (*waterfall*). Menurut Rosa dan M. Shalahuddin (2018) Model SDLC air terjun (*waterfall*) sering juga disebut model sekuensial linier (*sequential linier*) atau alur hidup klasik (*classic life cycle*). Proses tersebut digambarkan dalam gambar 3.



Gambar 3. Metode *Waterfall*
(Sumber: Rosa dan M. Shalahuddin (2018))

Langkah–langkah penelitian dalam metode *waterfall*:

1. Analisis kebutuhan perangkat lunak

Proses analisis kebutuhan mengidentifikasi kebutuhan apa saja yang dibutuhkan oleh perangkat lunak. Kebutuhan tersebut meliputi kebutuhan data, kebutuhan *software*, kebutuhan *hardware* dan kebutuhan *brainware*. Sedangkan *software* yang dibutuhkan untuk membuat aplikasi *login* berbasis GUI adalah Anaconda, XAMPP dan notepad++. Kebutuhan *brainware* yaitu *user*.

2. Desain

Proses desain merupakan hasil dari proses analisis yang mulai diterapkan untuk pembuatan perangkat lunak sesuai dengan kebutuhan yang sudah diketahui. Desain meliputi desain sistem dan desain interface aplikasi *login* yang akan dibuat.

3. Pengkodean

Pengkodean yaitu kode program mulai dibuat berdasarkan desain interface yang sudah dibuat. Ide – ide yang ada didesain mulai dikonversi ke dalam program.

4. Pengujian

pengujian dilakukan dengan pengujian beta untuk mengetahui apakah aplikasi *login* yang sudah dibuat layak dan dapat diterima oleh user atau tidak.

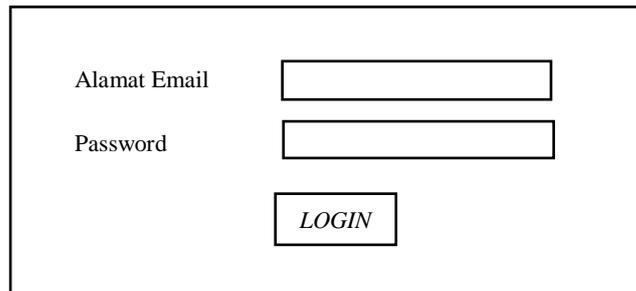
4. HASIL DAN PEMBAHASAN

1. Analisis kebutuhan aplikasi *login*

- Data yang dibutuhkan dalam pembuatan aplikasi *login* adalah data *user* yaitu alamat *email* dan *password*.
- Kebutuhan *hardware* yaitu perangkat keras yang digunakan dalam pembuatan aplikasi *login*.
- Kebutuhan *software* untuk membuat aplikasi *login* yaitu Anaconda, XAMPP dan notepad++

2. Desain

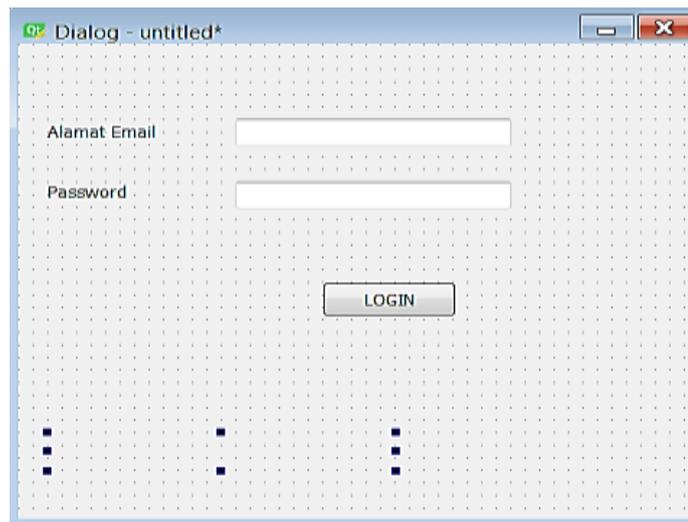
Sebelum membuat aplikasi *login*, dibuat desain perancangan untuk memudahkan proses desain. Perancangan interface dapat dilihat pada gambar 4.



The image shows a login form design within a rectangular border. It contains three input fields: a text box for 'Alamat Email', a text box for 'Password', and a button labeled 'LOGIN'.

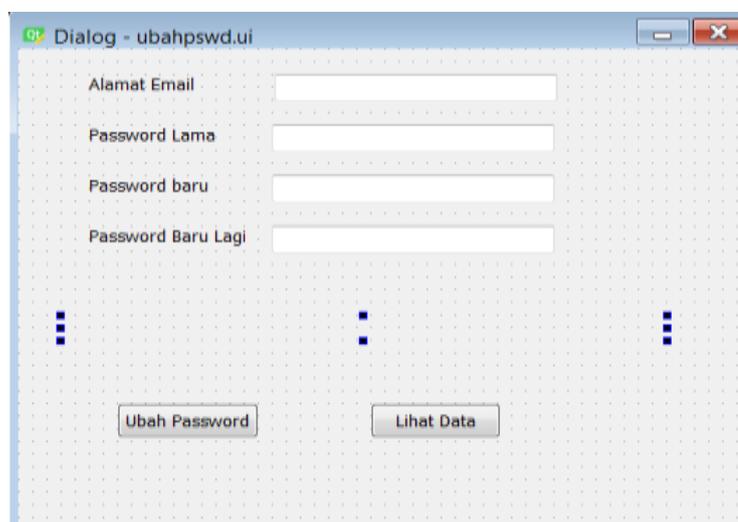
Gambar 4. Perancangan Form *Login*

Hasil dari perancangan implementasikan menggunakan QtDesigner dan widget akan tampak seperti pada gambar 5.

A screenshot of a Windows-style dialog box titled "Dialog - untitled*". The dialog has a light blue border and a dotted background. It contains two text input fields: "Alamat Email" and "Password". Below the "Password" field is a button labeled "LOGIN". At the bottom of the dialog, there are three vertical ellipsis icons (three dots) on the left, center, and right sides.

Gambar 5. Implementasi *Form Login*

Berikut hasil implmentasi dari form ubah password yang berfungsi untuk merubah password seperti pada gambar 6.

A screenshot of a Windows-style dialog box titled "Dialog - ubahpswd.ui". The dialog has a light blue border and a dotted background. It contains four text input fields: "Alamat Email", "Password Lama", "Password baru", and "Password Baru Lagi". Below the "Password Baru Lagi" field are two buttons: "Ubah Password" and "Lihat Data". At the bottom of the dialog, there are three vertical ellipsis icons (three dots) on the left, center, and right sides.

Gambar 6. Implementasi *Form Login*

3. Pengkodean

Hasil dari pengkodean untuk *form* tampilan data dapat dilihat pada gambar 7.

```

1 import sqlite3, sys
2 from PyQt5.QtWidgets import QDialog, QApplication, QTableWidgetItem
3 from sqlite3 import Error
4 from tampiluser import *
5 from panggil_tambah import form_tambah
6 from panggil_hapus import form_hapus
7 class form_tampil (QDialog) :
8     def __init__ (self) :
9         super (). __init__ ()
10        self.ui = Ui_Dialog ()
11        self.ui.setupUi (self)
12        self.ui.btntampil.clicked.connect(self.lihat)
13        self.ui.btntambah.clicked.connect(self.tambahdata)
14        self.ui.bthhapus.clicked.connect(self.hapusdata)
15        self.show()
16    def tambahdata (self) :
17        self.ui = form_tambah ()
18        self.show()
19    def hapusdata (self) :
20        self.ui = form_hapus ()
21        self.show()
22    def lihat (self) :
23        statemenSQL="SELECT * FROM login"
24        try :
25            koneksi = sqlite3.connect ("perpustakaan.db")
26            cur = koneksi.cursor ()
27            cur.execute (statemenSQL)
28            baris = cur.fetchall ()
29            noBaris=0
30            for tuple in baris :
31                self.ui.lbpesan.setText("")
32                noKolom=0
33                for kolom in tuple :
34                    satuKolom = QTableWidgetItem (kolom)
35                    self.ui.tbuser.setItem (noBaris, noKolom, satuKolom)
36                    noKolom+=1
37                noBaris+=1
38        except Error as e :
39            self.ui.tbuser.clear()
40            self.ui.lbpesan.setText("Error mengakses baris")
41        finally :
42            koneksi.close ()
43    if __name__ == "__main__" :
44        app = QApplication (sys.argv)
45        w = form_tampil ()
46        w.show ()
47        sys.exit (app.exec ())

```

Gambar 7. Koding untuk Form Tampilkan Data

Berikut merupakan tampilan form untuk menampilkan data serta menambah dan menghapus data yang dapat dilihat pada gambar 8. sesuai dengan koding diatas.

The screenshot shows a Qt dialog window titled "Dialog - tampiluser.ui". Inside the window, there is a section titled "Data User" with a "Tampilkan" button. Below this is a table with 3 rows and 2 columns. The first row has headers "1" and "2". The second and third rows have empty cells. Below the table are "Tambah Data" and "Hapus Data" buttons.

	1	2
1		
2		
3		

Gambar 7. Form Lihat Data

4. Pengujian

Perhitungan kuesioner menggunakan Skala *Likert*. Menurut Sugiyono (2015), skala *likert* adalah skala yang digunakan untuk mengukur persepsi, pendapat atau sikap seorang atau kelompok mengenai sebuah peristiwa atau fenomena sosial, berdasarkan definisi operasional yang telah ditetapkan oleh peneliti.

Tabel 1. Tabel Bobot Nilai

Skala Penilaian	Bobot Nilai
Sangat Setuju	5
Setuju	4
Netral	3
Tidak Setuju	2
Sangat Tidak Setuju	1

Tabel 2. Tabel Bobot Hasil

Jawaban	Keterangan
0% - 19,99%	Sangat Tidak Setuju
20% - 39,99%	Tidak Setuju
40% - 59,99%	Netral
60 – 79,99%	Setuju
80 – 100%	Sangat Setuju

Untuk menghasilkan interpretasi, menggunakan rumus:

$$\text{Rumus Index \%} = \frac{\text{Total Skor}}{Y} \times 100\%$$

Keterangan :

Total Skor = Jumlah responden x bobot nilai skala *likert*

Y = Jumlah total responden x skor tertinggi

Tabel Presentasi Nilai Jumlah responden ada 5 orang dengan jumlah pertanyaan 7 pertanyaan. Dari hasil perhitungan kuesioner didapatkan presentase sebagai berikut:

Tabel 3. Perolehan Nilai Total Kuesioner

No	Aspek – aspek	Penilaian				
		SS	S	N	TS	STS
1.	Aplikasi <i>login</i> ini dapat memberikan informasi sesuai dengan kebutuhan.	3	2	0	0	0
2.	Aplikasi <i>login</i> ini dapat membatasi pengguna dan memberikan hak akses yang terbatas.	2	3	0	0	0
3.	Aplikasi <i>login</i> ini memiliki keamanan yang baik.	3	1	1	0	0
4.	Tampilan dalam aplikasi <i>login</i> ini sudah menarik.	2	3	0	0	0
5.	Fungsi tombol – tombol navigasi dalam aplikasi <i>login</i> sudah berfungsi dengan baik.	1	4	0	0	0
6.	Saya mudah dalam menggunakan aplikasi <i>login</i> ini.	3	2	0	0	0

7.	Saya dapat memperoleh informasi data user dengan cepat dan tepat.	1	4	0	0	0
----	---	---	---	---	---	---

TOTAL	15	19	1	0	0
--------------	-----------	-----------	----------	----------	----------

Pehitungan Akhir :

$$\begin{aligned}
 \text{Total Skor} &= \text{Total (jumlah responden x bobot)} \\
 &= (15 \times 5) + (19 \times 4) + (1 \times 3) + (0 \times 2) + (0 \times 1) \\
 &= 75 + 76 + 3 + 0 + 0 \\
 &= 154
 \end{aligned}$$

$$\begin{aligned}
 \text{Skor Tertinggi} &= \text{Bobot nilai tertinggi x Jumlah Responden x Jumlah Pertanyaan} \\
 &= 5 \times 5 \times 7 \\
 &= 155
 \end{aligned}$$

$$\begin{aligned}
 \text{Rumus Index \%} &= \text{Total Skor / Skor Tertinggi (Y) x 100} \\
 &= 154 / 155 \times 100 \\
 &= 99,3\%
 \end{aligned}$$

5. KESIMPULAN

Aplikasi *login* dapat dibuat menggunakan QTDesigner Python dan aplikasi layak untuk digunakan. Hal ini dibuktikan dengan hasil UAT yang menunjukkan hasil index 99,3%.

DAFTAR PUSTAKA

- A. S., Rosa dan Shalahuddin, M., 2018. *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek*. Bandung: Informatika.
- B. Herry Suharto., Wijono, Soesilo., 2006. *Membangun Aplikasi Menggunakan Qt Designer dengan Database PostgreSQL/MySQL*. Yogyakarta: Penerbit Andi.
- Lutz, M., 2010. *Programming Python. Fourth Edition ed.* Sebastopo: O'Reilly Media, Inc.
- Lastiansah, Sena., 2012. *Pengertian User Interface*. Jakarta: Elex Media Komputindo.
- 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.
- Wilbert O. Galitz., 2007. *The Essential Guide to User Interface Design*. Canada: Wiley Publishing, Inc.