

# Kelompok 1

*by* Satu Kelompok

---

**Submission date:** 23-Jan-2021 09:59PM (UTC+0530)

**Submission ID:** 1492842352

**File name:** Laporan\_Kelompok\_1.pdf (1.81M)

**Word count:** 7002

**Character count:** 41592

# Laporan Tugas Besar

## Algoritma dan Pemrograman

### Document Validation



Dosen :  
Boby Mugi Pratama, S.Si., M.Han

Asisten :  
Indra Lukman Hakim 11191030  
Rani Yuniar Putriyanti 14191036

Disusun Oleh :  
Jackson Pangestuti 10201050  
Muhammad Nur Rahman 10201062  
M.Nawfal AlHadi 11201050  
Muhammad Huzaimi 11201062  
Ahmad Gilang Ramadan 12201002  
Rismawati 12201058  
Syarifah Chairunnisa 12201066

10 Desember 2020

## Daftar Isi

<b>15</b>	<b>Daftar Isi</b>	<b>2</b>
<b>Bab 1 PENDAHULUAN</b> <span style="float: right;"><b>4</b></span>		
1.1	Latar Belakang	4
1.2	Rumusan Masalah	4
1.3	Tujuan	5
1.4	Fitur	5
<b>Bab II LANDASAN TEORI</b> <span style="float: right;"><b>6</b></span>		
2.1	Landasan Teori Umum	6
2.1.1	Pengertian Dokumen	6
2.1.2	Pengertian Pemalsuan Data	6
2.1.3	Pengertian Sistem	6
2.2	Landasan Teori Pemrograman	7
2.2.1	Python	7
2.2.2	Variable	8
2.2.3	String	8
2.2.4	List	9
2.2.5	Read dan Write	10
2.2.6	For and While	11
2.2.7	Fungsi	12
2.2.8	Try except	13
2.2.9	PyQt	13
<b>Bab III PEMBAHASAN</b> <span style="float: right;"><b>14</b></span>		
3.1	Flowchart	14
3.1.1	Flowchart Pilih Menu	14
3.1.2	Flowchart Menambahkan User	14
3.1.3	Flowchart Menambahkan Dokumen	15
3.1.4	Flowchart Menampilkan List Dokumen	16
3.1.5	Flowchart Membandingkan Data	17
3.2	Pseudocode	17
3.3	Directory	21

3.4 Source Code	21
3.5 Additional Files	37
3.6 Run Program	38
3.6.1 Masukkan User Baru	38
3.6.2 Masukkan Dokumen Baru	39
3.6.3 List Dokumen	41
3.6.4 Cek Dokumen	43
<b>23 Bab IV PENUTUP</b>	<b>46</b>
4.1 Kesimpulan	46
4.2 Saran	46
<b>Daftar Pustaka</b>	<b>47</b>
<b>Lampiran</b>	<b>48</b>
<b>Lembar Pertanyaan</b>	<b>53</b>

## 11 Bab 1 PENDAHULUAN

### 1.1 Latar Belakang

Di era modern ini, informasi data kampus menjadi hal yang penting bagi para mahasiswa dengan adanya data tersebut, mahasiswa menjadi mudah mengakses media informasi terbaru dari kampus sehingga mengupload dokumen penting. Hal itu umum dilakukan oleh sebuah <sup>11</sup> perguruan tinggi dan juga kegiatan lain yang berhubungan dengan dokumen antara lain pengarsipan dokumen, baik dokumen dalam bentuk fisik maupun elektronik, sehingga diperlukan suatu metode yang praktis dan efisien dalam pengelolaannya. Salah satu metode yang digunakan dalam pengelolaan dokumen berupa sistem validasi dokumen. Cara seperti ini akan menimbulkan potensi pemalsuan dokumen, dimana pada dokumen tersebut telah diupload oleh mahasiswa. Jika ada mahasiswa di luar perguruan tinggi itu atau mahasiswa yang memiliki niatan jahat memperoleh dokumen-dokumen tersebut, maka mahasiswa itu dapat mengambil dokumen untuk membuat surat sejenis atau hal lain atas nama mahasiswa lain.

18

Untuk itu, menjaga keamanan dokumen dan data yang merupakan hal yang sangat penting untuk kerahasiaan informasi terutama yang berisi informasi sensitif yang hanya boleh diketahui datanya oleh pihak tertentu saja. Dimana mahasiswa yang telah mengupload dokumennya di website yang telah dimiliki oleh pihak perguruan tinggi harus merahasiakan datanya agar tidak mudah diketahui oleh orang lain sehingga, kami membuat program pengamanan informasi atau data-data yang penting dapat dilakukan dengan menyandikan data dengan kode-kode yang rumit agar tidak mudah untuk diketahui sehingga tidak ada pihak yang dapat mengakses ataupun menyalahgunakannya.

Oleh karena itu, program atau sistem bernama *validasi document* ini dibuat untuk digunakan meminimalisir potensi pemalsuan terhadap dokumen digital mahasiswa dalam website perguruan tinggi ataupun dokumen perusahaan. Agar penggunaanya merasa aman setelah mengupload dokumennya.

### 21 1.2 Rumusan Masalah

Dari latar belakang yang sudah dipaparkan diatas, penulis merumuskan beberapa masalah:

- a. Mudahnya seseorang untuk mengakses dokumen-dokumen mahasiswa maupun karyawan dapat menimbulkan potensi akan tindakan pemalsuan sehingga dibutuhkan sebuah sistem atau program untuk dapat mencegah tindakan tersebut.
- b. Keamanan data dokumen hal yang sangat penting berisi informasi yang sensitif, maka dari itu dibuat sistem pengamanan data informasi.

### **1.3 Tujuan**

Dari permasalahan yang sudah dipaparkan di atas, penulis membuat program ini dengan tujuan :

- a. Untuk meminimalisir potensi pemalsuan terhadap dokumen digital mahasiswa dalam perguruan tinggi atau dokumen perusahaan yang menggunakan nama perusahaan.
- b. Untuk menjaga dokumen agar identitas pemilik dokumen itu tidak disalahguna sehingga pengguna merasa aman.

### **1.4 Fitur**

Fitur yang terdapat pada program *validasi document* adalah sebagai berikut :

- a. Dapat menyimpan dokumen mahasiswa atau karyawan.
- b. Dapat melakukan pengecekan dokumen dengan output berupa hasil persentase perbandingan dengan dokumen yang ada di database.
- c. Dapat melihat list dokumen yang tersimpan sesuai nim/id karyawan.

26

## Bab II LANDASAN TEORI

### 2.1 Landasan Teori Umum

#### 2.1.1 Pengertian Dokumen

Dokumen merupakan unit terstruktur dari informasi terekam, diterbitkan ataupun tidak diterbitkan, dalam bentuk kopi makas atau bentuk elektronik, dan dikelola sebagai unit diskrit dalam sistem informasi Sulistyo, 2003, hal. 378. Wiggins (2000), mendefinisikan dokumen sebagai rekaman yang saling berkaitan (coherent), berisi informasi yang berguna bagi pemakainya.

#### 2.1.2 Pengertian Pemalsuan Data

Pemalsuan data diartikan sebagai suatu perbuatan yang mempunyai tujuan untuk meniru, menciptakan suatu benda yang sifatnya tidak asli lagi atau membuat suatu benda kehilangan keabsahannya. Sama halnya dengan membuat surat palsu, pemalsuan dapat dapat terjadi terhadap sebagian atau seluruh isi data. Menurut Soenarto Soerodibroto (1994 : 154), mengemukakan bahwa, barang siapa di bawah suatu tulisan membubuhkan tanda tangan orang lain sekalipun atas perintah dan persetujuan orang tersebut telah memalsukan tulisan itu. Perbedaan prinsip antara perbuatan membuat surat palsu dan memalsu surat, adalah bahwa membuat surat/membuat palsu surat, sebelum perbuatan dilakukan, belum ada surat,kemudian dibuat suatu surat yang isinya sebagian atau seluruhnya adalah bertentangan dengan kebenaran atau palsu. Seluruh tulisan dalam surat itu dihasilkan oleh perbuatan membuat surat palsu. Surat yang demikian disebut dengan surat palsu atau surat tidak asli.

#### 2.1.3 Pengertian Sistem

Menurut Jesa Ariawan dan Sri Wahyuni (2015), sistem adalah kumpulan dari sub-sub sistem baik sistem abstrak maupun fisik yang saling terintegrasi dan berkolaborasi untuk mencapai tujuan tertentu. Sistem adalah setiap sesuatu yang terdiri dari objek-objek, atau unsur-unsur, atau komponen - komponen yang bertata kaitan dan bertata hubungan satu sama lain, sedemikian rupa sehingga unsur-unsur tersebut merupakan satu kesatuan pemrosesan atau pengolahan yang tertentu.

## 2.2 Landasan Teori Pemrograman

7

### 2.2.1 Python

Python adalah bahasa pemrograman model skrip (scripting language) yang berorientasi objek. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi. Python merupakan bahasa pemrograman yang freeware atau perangkat bebas dalam arti sebenarnya, tidak ada batasan dalam penyalinannya atau mendistribusikannya. Lengkap dengan source codenya, debugger dan profiler, antarmuka yang terkandung di dalamnya untuk pelayanan antarmuka, fungsi sistem, GUI (antarmuka pengguna grafis), dan basis datanya. (Triasanti, 2010). Contoh program sederhana python:

Baris	pyhton.py
1	3 # Meminta inputan dari user
2	
3	
4	bil1 = input('Masukkan bilangan pertama: ')
5	
6	bil2 = input('Masukkan bilangan kedua: ')
7	
8	# Menjumlahkan bilangan
9	
10	jumlah = float(bil1) + float(bil2)
11	
12	
13	
14	
15	# Menampilkan jumlah
16	
17	print('Jumlah {0} + {1} adalah {2}'.format(bil1, bil2, jumlah))
18	
19	
20	
21	
22	#Maka output yang keluar:
23	
24	Masukkan bilangan pertama: 1.5
25	
26	
27	Masukkan bilangan kedua: 4.7
28	
29	Jumlah 1.5 + 4.7 = 6.2
30	
31	

## 2.2.2 Variable

2

Variabel python adalah lokasi memori yang dicadangkan untuk menyimpan nilai-nilai. Ini berarti bahwa ketika Anda membuat sebuah variabel Anda memesan beberapa ruang di memori. Variabel menyimpan data yang dilakukan selama program dieksekusi, yang nantinya isi dari variabel tersebut dapat diubah oleh operasi - operasi tertentu pada program yang menggunakan variabel. Variabel dapat menyimpan berbagai macam tipe data. Di dalam pemrograman Python, variabel mempunyai sifat yang dinamis, artinya variabel Python tidak perlu dideklarasikan tipe data tertentu dan variabel Python dapat diubah saat program dijalankan. Contohnya:

Baris	Variabel.py
1	1
2	variabel_ku = "ini isi variabel"
3	variabel2 = 20
4	
5	#Kemudian untuk melihat isi variabel, kita dapat
6	menggunakan fungsi print.
7	
8	print variabel_ku
9	
10	print variabel2
11	
12	#Maka outputnya adalah
13	
14	ini isi variable
15	
16	20
17	
18	

## 2.2.3 String

String pada Python adalah tipe data yang memuat satu karakter atau lebih karakter (*sequences of character*) yang diapit oleh tanda petik tunggal ('') atau tanda petik (""). Dalam bahasa pemrograman Python, deklarasi suatu string tidak dibedakan penggunaan tanda petik atau tanda petik tunggal. Contohnya:

Baris	string.py
1	<b>3</b>
2	var1 = 'Hello Python'
3	
4	var2 = 'Programming with Python'
5	
6	#untuk mengaksesnya
7	<b>3</b>
8	var1 = 'Hello Python!'
9	
10	var2 = "I love Python"
11	
12	print("var1[0]", var1[0])
13	
14	print("var2[2:6]:" , var2[2:6])
15	
16	#Maka outputnya adalah:
17	
18	var1[0]: H
19	
20	var2[2:6]: love
21	

#### 2.2.4 List

**2**

List adalah struktur data yang paling dasar adalah urutan atau lists. Setiap elemen-elemen berurutan akan diberi nomor posisi atau indeksnya. Indeks pertama dalam list adalah nol, indeks kedua adalah satu dan seterusnya..List adalah tipe data yang paling serbaguna yang tersedia dalam bahasa Python, yang dapat ditulis sebagai daftar nilai yang dipisahkan koma (item) antara tanda kurung siku. Hal penting tentang daftar adalah item dalam list tidak boleh sama jenisnya. List merupakan struktur data yang menyimpan kumpulan item. Item dapat berupa bilangan integer, float, string dan boolean. Item di dalam list terurut dan nilainya dapat berubah atau lebih dikenal dengan istilah *mutable*. Contoh list :

Baris	list.py
1	#membuat list kosong
2	minuman = []
3	
4	
5	
6	#membuat list dengan isi 1 item
7	hobi = ["berenang"]
8	
9	
10	
11	#membuat list dengan isi lebih dari 1 item
12	
13	sayur = ["kubis", "brokoli", "bayam"]
14	

## 2.2.5 Read dan Write

10

Read digunakan untuk menerima input, dengan posisi kursor – penunjuk aktif, setelah diberi input pada baris berikutnya. Read tidak membaca end of line marker. Pernyataan ini tidak akan membuang kelebihan data. Kelebihan data akan diberikan pada read atau readln berikutnya. Read tidak ganti baris, masih dalam baris yang sama.

Perintah ini berfungsi untuk menerima atau memasukkan input ke program, dan masih satu baris (di baris yang sama). kode Perintah akan terus dibaca hingga ENTER ditekan (berganti baris). Perintah ini juga biasa disebut dengan *input* atau *masukan* dalam pemrograman.

11

Write digunakan untuk menuliskan output, dengan posisi kursor – penunjuk aktif, terletak pada akhir output yang ditampilkan pada baris yang sama. Write tidak menambahkan CR setelah menampilkan data semua argument. Write menampilkan hasil tanpa ganti baris dan tampilan berikutnya akan disambung dalam baris yang sama. Contoh read and write :

Baris	Read and Write.py
1	#membuka file
2	file_bio = open("biodata".txt, "r+")
3	
4	
5	#membaca file
6	teks = file_bio.read()
7	
8	
9	
10	#menulis file
11	file_bio.write(teks)
12	
13	
14	
15	#menutup file
16	file_bio.close()
17	

## 2.2.6 For and While

5

For merupakan syntax untuk membuat suatu perulangan dengan jumlah yang telah diketahui. Perulangan for akan mengulang blok kode program disertai nilai pada rentang tertentu secara berurutan. Rentang nilai tersebut didapatkan melalui fungsi range yang menghasilkan nilai bilangan bulat antara nilai awal sampai dengan batas akhir. Walaupun fungsi range sering digunakan dalam perulangan for, fungsi tersebut bukanlah satu-satunya cara untuk melakukan perulangan. For juga dapat digunakan untuk mengunjungi setiap elemen dalam list. Contoh penggunaan perulangan For didalam list :

Baris	Perulangan FOR
	2
1	angka = ["1", "2", "3", "4"]
2	for a in angka
3	print (a)

Maka output yang keluar adalah

Baris	Perulangan FOR
1	1
2	2
3	3
4	4

13

While adalah perulangan uncountable atau perulangan yang jumlah proses pengulangannya tidak ditentukan. Ia akan menjalankan baris kode di dalam blok kodennya secara terus menerus selama masih memenuhi ekspresi yang sudah ditentukan sebelumnya, yang berarti ia akan terus mengulang selama kondisi bernilai True. Contoh penggunaan perulangan While :

Baris	Perulangan WHILE
1	a = 1
2	While a <= 3
3	print('ITK')
4	a += 1

Maka output yang keluar adalah

Baris	Perulangan WHILE
1	ITK
2	ITK
3	ITK

1

## 2.2.7 Fungsi

Fungsi adalah blok kode terorganisir dan dapat digunakan kembali yang digunakan untuk melakukan sebuah tindakan/action. Fungsi memberikan modularitas yang lebih baik aplikasi Anda dan tingkat penggunaan kode yang tinggi. Mendefinisikan Fungsi Python anda dapat menentukan fungsi untuk menyediakan fungsionalitas yang dibutuhkan. Berikut adalah aturan sederhana untuk mendefinisikan fungsi dengan Python. Fungsi blok dimulai dengan def kata kunci diikuti oleh nama fungsi dan tanda kurung (). Setiap parameter masukan atau

argumen harus ditempatkan di dalam tanda kurung ini. Anda juga dapat menentukan parameter di dalam tanda kurung ini. Pernyataan pertama dari sebuah fungsi dapat berupa pernyataan opsional - string dokumentasi fungsi atau docstring. Blok kode dalam setiap fungsi dimulai dengan titik dua (:) dan indentasi. Pernyataan kembali [ekspresi] keluar dari sebuah fungsi, secara opsional menyampaikan kembali ekspresi ke pemanggil. Pernyataan pengembalian tanpa argumen sama dengan return None.

Contoh fungsi :

Baris	fungsi.py
28	<code>def printme( str ):</code>
2	<code>    "This prints a passed string into this</code>
3	<code>    function"</code>
4	<code>        print (str)</code>
	<code>return</code>

#### 2.2.8 Try except

<sup>3</sup> Python menyediakan metode penanganan eksepsi dengan menggunakan pernyataan try dan except. Di dalam blok try kita meletakkan baris program yang kemungkinan akan terjadi error. Bila terjadi error, maka penanganannya diserahkan kepada blok except.menangani error secara umum, dikelompokkan apakah dia adalah TypeError, ValueError, SyntaxError, dan lain sebagainya. Sebuah pernyataan try, bisa memiliki sejumlah pernyataan except untuk menangani jenis – jenis eksepsi secara lebih spesifik.

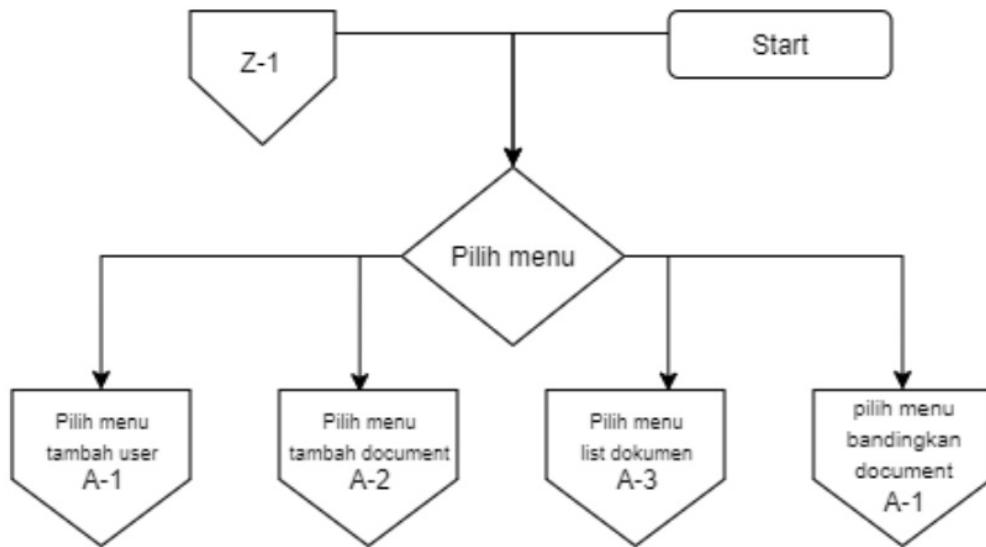
#### 2.2.9 PyQt

PyQt adalah lintas platform GUI perangkat Qt yang mengikat pada Python, diimplementasikan sebagai plug-in. PyQt adalah perangkat lunak gratis yang dikembangkan oleh perusahaan Inggris Riverbank Computing. Ini tersedia di bawah persyaratan yang sama untuk Qt versi yang lebih tua dari 4,5; ini berarti berbagai termasuk lisensi GNU General Public License (GPL) dan lisensi komersial, tetapi tidak GNU Lesser General Public License (LGPL).PyQt mendukung Microsoft Windows serta berbagai varian dari UNIX, termasuk Linux dan MacOS.

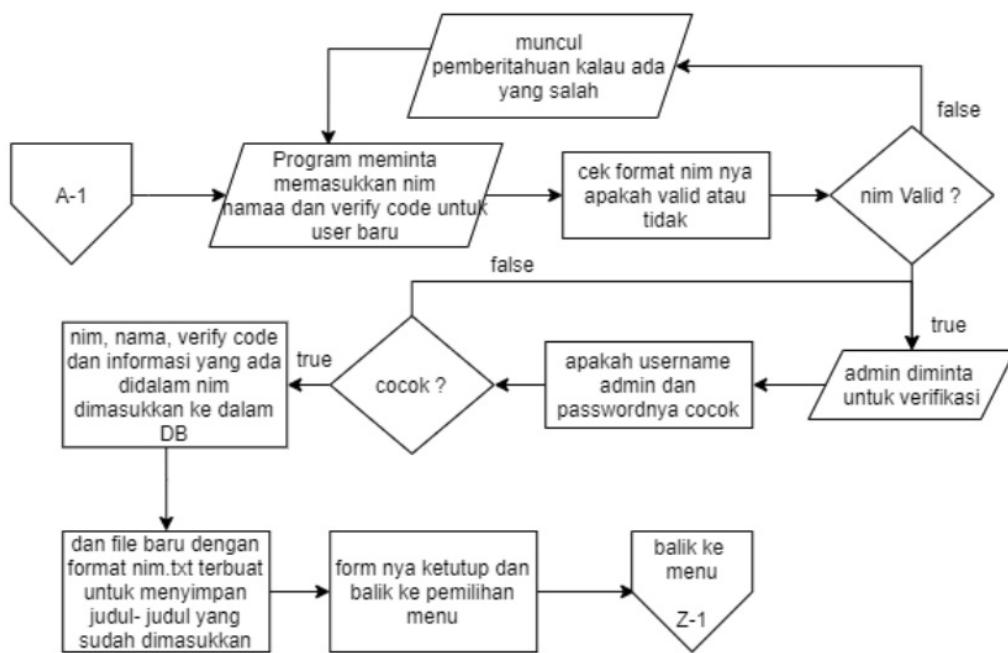
## Bab III PEMBAHASAN

### 3.1 Flowchart

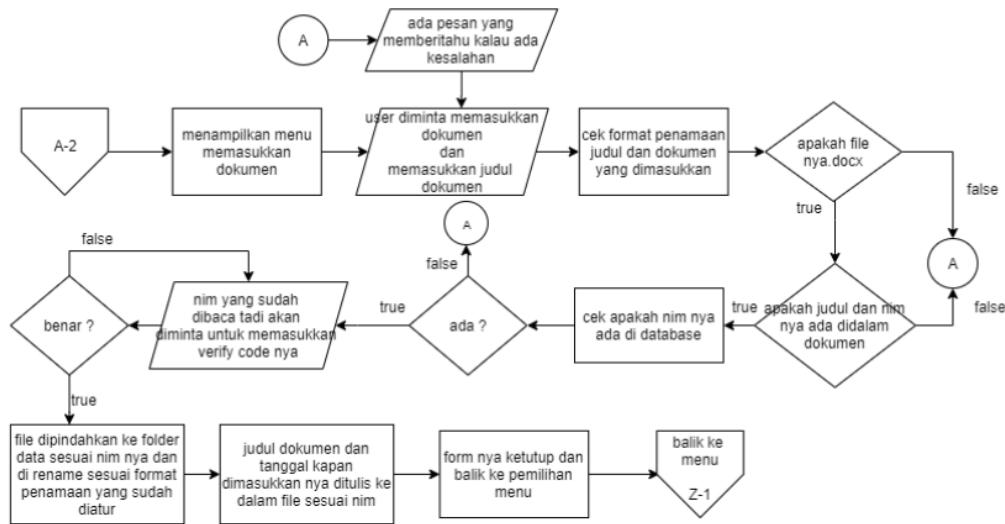
#### 3.1.1 Flowchart Pilih Menu



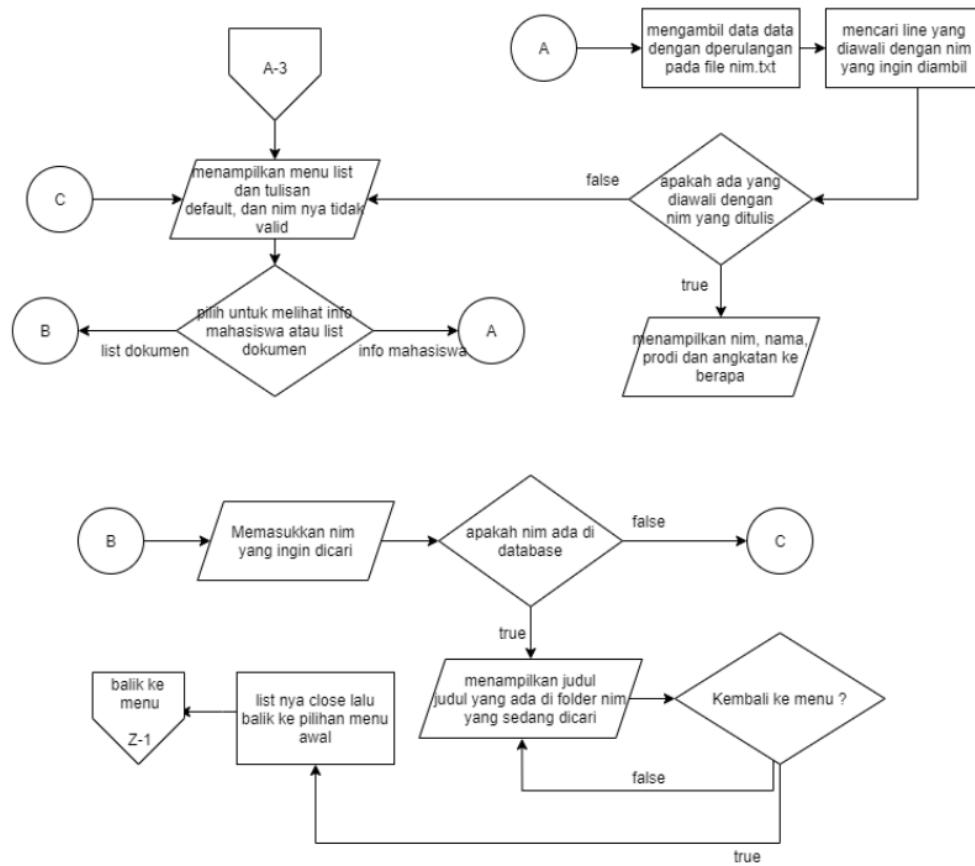
#### 3.1.2 Flowchart Menambahkan User



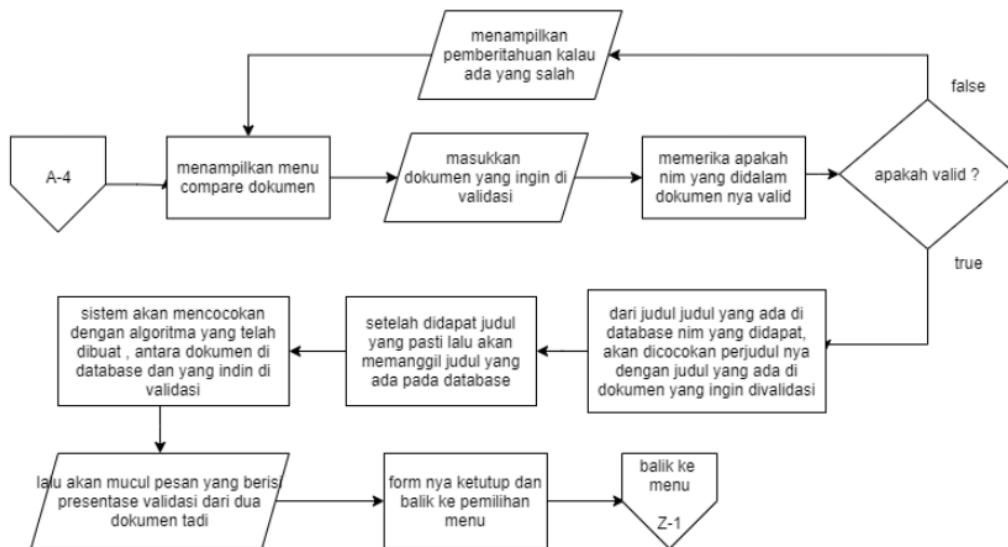
### 3.1.3 Flowchart Menambahkan Dokumen



### 3.1.4 Flowchart Menampilkan List Dokumen



### 3.1.5 Flowchart Membandingkan Data



### 3.2 Pseudocode

	pseudocode
	<pre> firstMenu : menu read(firstMenu)  while true:     if (firstMenu is newUser) then:         newUserMenu()      else if (firstMenu is newDocx) then:         newDocxMenu()      else (if firstMenu is listDoc) then:         listDocMenu()      else if (firstMenu is compareDoc) then:         compareDocxMenu()      else if (firstMenu is quit) then:         break  def newUserMenu():     nim : int     vefCode : string     name : string   </pre>

```

        write( newUserFunction(read(nim), read(name),
read(vefCode)) )

def newUserFunction(nim, name, vefCode):
    prodi : string
    angkatan : int

    with open('info', 'r') as info:
        for i in info:
            if len(nim) is 8:
                if nim[0:2] in i:
                    // example of every line writed like
this : 11, Informatika
                    prodi = read(i.split())[0]
                    break
            else if len(nim) is 7:
                if nim[0] in i:
                    prodi = read(i.split())[0]
                    break

    if len(nim) is 8:
        angkatan = read(nim)[2:4]
    else if len(nim) is 7:
        angkatan = read(nim)[1:3]

    return
adminVerificationForm(nim,name,vefcode,prodi,angkatan)

def adminVerificationForm(nim,name,vefcode,prodi,angkatan):
    username : string
    password : string

    if username is 'admin' && password is 'admin':
        with open('infoUser.txt', 'a') as user:
            user.write(nim,name,vefcode,prodi,angkatan)
            // will write the line like this : 112010XX,Nama
Orang,12345,Informatika,20
            return 'succes'
    else:
        return 'failed'

def newDocxMenu():
    pathFromCom : string
    title : string
    nim : int
    dateUpload : string

    status = newDocxFun(read(pathFromCom), read(title),
read(nim), read(dateUpload))
    if status[0] is 'succes':
        file move to folder Data/File/read(nim) /status[1]
    else if status[0] is 'failed':
        write.console("Failed")

def newDocxFun(path, title, nim, dateUpload):

```

```

// format date %y%m%d 210130
for i in open(path) :
    read i
    if title in i:
        fixTitle = title_nim_dateUpload.txt // writted
like this : judul dokumen_112010XX_210130.txt
        with open(User/nim+'.txt', w) as title:
            title.write(fixTitle, dateUpload) //
writted like this : judul dokumen_112010XX_210130.txt, 210130
        return ['succes', 'fixTitle']
    return ['failed', 'Error']

def listDocMenu():
    nim : string

    with open(User/read(nim)+'.txt', 'r') as user:
        for i in user:
            write(i.split(',')[0]) // will write the title
of document that have uploaded by nim that you search

def compareDocxMenu():
    pathOutsideDb : string
    pathInsideDb: string

    tempStatus : []

    open with('infoUser.txt', 'r') as user;
        for i in user:
            for open(read(pathOutsideDb)) as eachline: //
open the file
                if i.split('0')[0] in eachLine : // checking the nim on every each line
                    tempStatus = ['succes', i] // bring nim to tempStatus
                tempStatus = ['failed', 0] // if there is no nim on that document

            if tempStatus[0] is 'succes':
                pathInsideDb = gettingPathInsideDb(tempStatus[1],
path) // bring nim to another function
            else:
                write.console("failed")

            write(compareDocxFuction(pathInsideDb, pathOutsideDb))

def gettingPathInsideDb(nim, path):

    fixTitle = ''

    open with(User/nim+'.txt', 'r') as title: // inside file
every line like this : judul dokumen_112010XX_210130.txt, 210130
        for i in title:
            tempTitle = i.split(',')[0]

```

```

        if tempTitle.split('_')[0] in open(path): // finding title inside the path
            return i.split(',')[0]
            break
        else:
            return 'masukkan judul dokumen seperti yang ada dalam dokumen'

def compareDocxFunction(fileInside, fileOutside):
    // reminder for nawfal : compare function may change the method while code
    takeAllInside : list
    takeAllOutside: list
    count : int = 0
    truesn : int = 0

    takeAllInside = open(fileInside).split()
    takeAllOutside = open(fileOutside).split()

    lengthIn = length(open(fileInside).split()) // count every line of word in insideFile

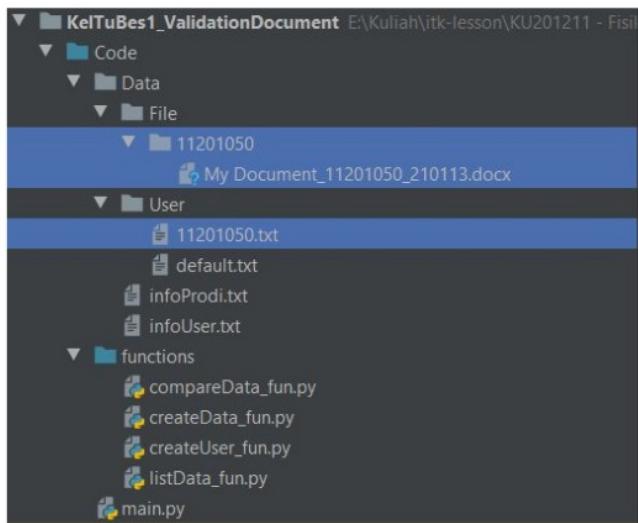
    for i in range(lengthIn):
        if takeAllInside[i] == takeAllOutside[i]:
            count += 1
            truesn +=1
        else:
            count += 1

    return (true/count)*100

write.console('system exit')

```

### 3.3 Directory



File yang diblok dengan warna biru itu adalah file/folder yang akan muncul setelah data di input misalkan pada User/11201050.txt itu akan muncul setelah memasukkan user sedangkan yang Data/File/11201050/My Document\_11201050\_210113.docx akan baru ada setelah file nya dimasukkan pada fitur masukkan dokumen.

### 3.4 Source Code

	Main.py
	<pre>import sys, docx import string  from PyQt5.QtWidgets import *  # Imported all fun that i made import createData_fun import listData_fun import compareData_fun import createUser_fun  22 class main(QWidget):     def __init__(self):         super().__init__()         self.nim = "default"          self.setWindowTitle("Document Validation")         self.setMinimumWidth(500)         self.setMaximumWidth(500)         self.setMinimumHeight(110)         self.setMaximumHeight(110)</pre>

```

        self.menu_layout()

        self.mainLayout = QHBoxLayout()

        self.mainLayout.addLayout(self.menuLayout1)
        self.btn.addData.clicked.connect(self.createUIs)
        self.btn_addUser.clicked.connect(self.createUserUIs)

        self.mainLayout.addWidget(self.menuLayout2)
        self.btn_listData.clicked.connect(self.listUIs)
        self.btn_cekData.clicked.connect(self.checkingUIs)

        # self.mainLayout.addWidget(self.menpuuLayout3)

        self.setLayout(self.mainLayout)
        self.show()

    def errorMsg(self, msg, layout):
        self.msg_box = QMessageBox(layout)
        self.msg_box.setText(msg)
        self.msg_box.exec_()

    def menu_layout(self):
        self.btn.addData = QPushButton('Masukkan\n Dokumen')
        self.btn_addUser = QPushButton('Masukkan\n User Baru')
        self.btn_listData = QPushButton('List\n Dokumen')
        self.btn_cekData = QPushButton('Cek\n Dokumen')

        self.menuLayout1 = QVBoxLayout()
        self.menuLayout1.addWidget(self.btn.addData)
        self.menuLayout1.addWidget(self.btn.addUser)

        self.menuLayout2 = QVBoxLayout()
        self.menuLayout2.addWidget(self.btn_listData)
        self.menuLayout2.addWidget(self.btn_cekData)

        # self.menuLayout3 = QVBoxLayout()
        # self.menuLayout3.addWidget(self.btn_cekData)

    # Create Data UI

    def createUIs(self):
        self.createWindow = QMainWindow(self)
        self.createWindow.setMinimumHeight(400)
        self.createWindow.setMaximumHeight(400)
        self.createWindow.setMinimumWidth(500)
        self.createWindow.setMaximumWidth(500)

        # On The Top

        self.text_title = QLabel(self.createWindow)
        self.text_title.setText('Judul :')
        self.text_title.move(18, 25)

        self.input_title = QLineEdit(self.createWindow)
        self.input_title.setText('Judul Dokumen')
        self.input_title.move(65, 25)
        self.input_title.setFixedWidth(400)

```

```

# On The Bottom

self.text_path = QLabel(self.createWindow)
self.text_path.setText('PATH :')
self.text_path.move(18, 270)

self.dic_name = QLineEdit(self.createWindow)
self.dic_name.setText("")
self.dic_name.move(65, 270)
self.dic_name.setFixedWidth(320)

self.input_file = QPushButton(self.createWindow)
self.input_file.move(400, 270)
self.input_file.setText("Browse... ")
self.input_file.setFixedWidth(80)

self.input_file.clicked.connect(self.getDic)

self.submit_file = QPushButton(self.createWindow)
self.submit_file.move(150, 320)
self.submit_file.setText("Upload")
self.submit_file.setFixedWidth(200)

self.submit_file.clicked.connect(lambda:
self.valid_vedCode(self.dic_name.text(), self.input_title.text()))
self.createWindow.show()

# Create User UI

def createUserUIs(self):
    self.cuserLayout = QMainWidget(self)
    self.cuserLayout.setMaximumSize(300, 260)
    self.cuserLayout.setMinimumSize(300, 260)

    self.text_nimUser = QLabel(self.cuserLayout)
    self.text_nimUser.setText("Nim : Maximal Character 7")
    self.text_nimUser.move(10, 10)
    self.text_nimUser.setFixedWidth(300)

    self.input_nimUser = QLineEdit(self.cuserLayout)
    self.input_nimUser.move(10, 40)
    self.input_nimUser.setFixedWidth(280)

    self.text_nameUser = QLabel(self.cuserLayout)
    self.text_nameUser.setText("Name : ")
    self.text_nameUser.move(10, 70)
    self.text_nameUser.setFixedWidth(300)

    self.input_nameUser = QLineEdit(self.cuserLayout)
    self.input_nameUser.move(10, 100)
    self.input_nameUser.setFixedWidth(280)

    self.text_codeUser = QLabel(self.cuserLayout)
    self.text_codeUser.move(10, 130)
    self.text_codeUser.setText("Verify Code : Minimal 4
Character")
    self.text_codeUser.setFixedWidth(300)

```

```

        self.input_codeUser = QLineEdit(self.cuserLayout)
        self.input_codeUser.move(10, 160)
        self.input_codeUser.setFixedWidth(280)

        self.btn_newUser = QPushButton(self.cuserLayout)
        self.btn_newUser.setText("Masukkan Data")
        self.btn_newUser.move(10, 200)
        self.btn_newUser.setFixedWidth(280)

        self.btn_newUser.clicked.connect(self.valid_vehAdmin)

        self.cuserLayout.show()

# Validasi Admin Buat Masukkan Data Ke Info User

def valid_vehAdmin(self):
    self.vadminLayout = QMainWidget(self)
    self.vadminLayout.setMinimumSize(300, 200)
    self.vadminLayout.setMaximumSize(300, 200)

    self.text_usernameAdmin = QLabel(self.vadminLayout)
    self.text_usernameAdmin.setFixedWidth(300)
    self.text_usernameAdmin.setText("Username : ")
    self.text_usernameAdmin.move(10, 10)

    self.input_usernameAdmin = QLineEdit(self.vadminLayout)
    self.input_usernameAdmin.setFixedWidth(280)
    self.input_usernameAdmin.move(10, 40)

    self.text_vehCodeAdmin = QLabel(self.vadminLayout)
    self.text_vehCodeAdmin.setFixedWidth(300)
    self.text_vehCodeAdmin.setText("Password : ")
    self.text_vehCodeAdmin.move(10, 70)

    self.input_vehCodeAdmin = QLineEdit(self.vadminLayout)
    self.input_vehCodeAdmin.setFixedWidth(280)
    self.input_vehCodeAdmin.move(10, 100)

    self.btn_formAdmin = QPushButton(self.vadminLayout)
    self.btn_formAdmin.setFixedWidth(280)
    self.btn_formAdmin.move(10, 140)
    self.btn_formAdmin.setText("Submit Data")

    self.btn_formAdmin.clicked.connect(lambda:
self.verifyingAdmin(self.input_usernameAdmin.text().lower(),

self.input_vehCodeAdmin.text().lower()))

    lengthNim = len(self.input_nimUser.text())
    validationLength = 0
    for i in range(lengthNim):
        if self.input_nimUser.text()[i] in string.digits:
            validationLength += 1

        if len(self.input_nimUser.text().strip()) < 5 or
len(self.input_nimUser.text().strip()) > 8 or \
            len(self.input_codeUser.text()) < 4 or validationLength
!= lengthNim:
            self.errorMsg("Salah memasukkan NIM atau kurang memasukkan

```

```

    code", self.vadminLayout)
        self.vadminLayout.close()
    else:
        self.vadminLayout.show()

    # Validasi Vef Code

    def valid_veyCode(self, path, title):
        self.veyLayout = QMainWindow(self)
        self.veyLayout.setMinimumWidth(500)
        self.veyLayout.setMaximumWidth(500)
        self.veyLayout.setMinimumHeight(120)
        self.veyLayout.setMaximumHeight(120)

        self.text_vey = QLabel(self.veyLayout)
        self.text_vey.setText('Verify Your Code')
        self.text_vey.move(200, 15)

        self.text_vey = QLabel(self.veyLayout)
        self.text_vey.setText('Verification Code :')
        self.text_vey.move(18, 55)

        self.input_vey = QLineEdit(self.veyLayout)
        self.input_vey.setText('Judul Dokumen')
        self.input_vey.move(125, 55)
        self.input_vey.setFixedWidth(265)

        self.btn_vey = QPushButton(self.veyLayout)
        self.btn_vey.setText("Verif")
        self.btn_vey.move(400, 55)
        self.btn_vey.setFixedWidth(80)

    # Code =====

    if path == "" or len(path) < 6:
        self.errorMsg("Masukkan File Terlebih Dahulu",
self.createWindow)
        self.veyLayout.close()
    elif "docx" not in path:
        self.errorMsg("Harus menggunakan File .docx",
self.createWindow)
    elif path != "":
        if self.input_title.text() == "":
            self.errorMsg("Masukkan Judul Yang Ada Di Dokumen",
self.createWindow)
        else:
            gate1 = createData_fun.searchNim(self, path)
            if gate1[0] == 0:
                self.errorMsg(gate1[1], self.createWindow)
            else:
                doc = docx.Document(path).paragraphs
                gate2 = createData_fun.searchTitle(self,
self.input_title.text(), doc)
                if gate2[0] == 0:
                    self.errorMsg(gate2[1], self.errorMsg())
                elif gate2[0] == 1:
                    print(gate2)
                    tempNim = createData_fun.getNim(path)
                    print("Bismillah")

```

```

        self.btn_vey.clicked.connect(lambda:
self.vey_codew(title, tempNim, path))
        print(tempNim)
        self.veyLayout.show()

# Bismillah work :"D

# Code =====

# Start Of List UI

def listUIs(self):
    self.listWindow = QMainWindow(self)
    self.listWindow.setMinimumWidth(510)
    self.listWindow.setMaximumWidth(510)
    self.listWindow.setMinimumHeight(530)
    self.listWindow.setMaximumHeight(530)

# =====
# Center =====
# =====

self.text_searchDoc = QLabel(self.listWindow)
self.text_searchDoc.setText('NIM :')
self.text_searchDoc.move(18, 25)

self.input_searchDoc = QLineEdit(self.listWindow)
self.input_searchDoc.setText(f'{self.nim}')
self.input_searchDoc.move(65, 25)
self.input_searchDoc.setFixedWidth(320)

self.btn_searchDoc = QPushButton(self.listWindow)
self.btn_searchDoc.setText("Search")
self.btn_searchDoc.move(400, 25)
self.btn_searchDoc.setFixedWidth(80)
self.btn_searchDoc.clicked.connect(self.getNim)

self.btn_infoUser = QPushButton(self.listWindow)
self.btn_infoUser.setText("Info Mahasiswa")
self.btn_infoUser.move(5, 490)
self.btn_infoUser.setFixedWidth(500)
self.btn_infoUser.clicked.connect(lambda:
self.getInfo(self.nim))

# =====
# Below =====
# =====

self.formLayout = QFormLayout()
self.groupBox = QGroupBox(f"Dokumen {self.nim}")
self.btn_info = QPushButton

with open(f"Data/User/{self.nim}.txt", 'r') as nim:
    for i in nim:
        temp = i.split(',')
        slashN = temp[1][0:len(temp[1]) - 1]
        self.formLayout.addRow(QLabel(temp[0]), QLabel(f'|{slashN}|'))

```

```

        self.groupBox.setLayout(self.formLayout)
        self.groupBox.setMinimumWidth(490)
        self.scroll = QScrollArea(self.listWindow)
        self.scroll.setWidget(self.groupBox)
        self.scroll.setMinimumWidth(500)
        self.scroll.setMinimumHeight(400)
        self.scroll.move(5, 70)

        self.listWindow.show()

    # Info Mahasiswa

    def infoWindow(self, nim, name, prodi, status):
        self.infomahasiswaLayout = QMainWidget()
        self.infomahasiswaLayout.setWindowTitle(f'Data {status} {nim}')
        self.infomahasiswaLayout.setMinimumSize(250, 130)
        self.infomahasiswaLayout.setMaximumSize(250, 130)

        self.infoNim = QLabel(self.infomahasiswaLayout)
        self.infoNim.setText(nim)
        self.infoNim.move(10, 10)

        self.infoName = QLabel(self.infomahasiswaLayout)
        self.infoName.setText(f'Name : {name}')
        self.infoName.setFixedWidth(300)
        self.infoName.move(10, 40)

        self.infoProdi = QLabel(self.infomahasiswaLayout)
        self.infoProdi.setText(f'Prodi : {prodi}')
        self.infoProdi.setFixedWidth(300)
        self.infoProdi.move(10, 70)

        self.infoStatus = QLabel(self.infomahasiswaLayout)
        self.infoStatus.setText(f'Status : {status}')
        self.infoStatus.setFixedWidth(300)
        self.infoStatus.move(10, 100)

        if nim == 'default':
            self.errorMsg("Masukkan NIM yang ada di Database",
self.listWindow)
            self.infomahasiswaLayout.close()
        else:
            self.infomahasiswaLayout.show()

    def checkingUis(self):
        self.checkingLayout = QMainWidget(self)
        self.checkingLayout.setMinimumWidth(920)
        self.checkingLayout.setMaximumWidth(920)
        self.checkingLayout.setMinimumHeight(330)
        self.checkingLayout.setMaximumHeight(330)

    # Left Section

        self.text_dicL = QLabel(self.checkingLayout)
        self.text_dicL.setText("C:/.../fileName.docx")
        self.text_dicL.move(30, 259)
        self.text_dicL.setFixedWidth(1000)

        self.btn_dicL = QPushButton(self.checkingLayout)

```

```

        self.btn_dicL.setText("Search")
        self.btn_dicL.move(325, 260)

    # Right Section

        self.dic_get = QLineEdit(self.checkingLayout)
        self.dic_get.setText('C:/...')
        self.dic_get.move(438, 260)
        self.dic_get.setFixedWidth(365)

        self.text_dic = QLabel(self.checkingLayout)
        self.text_dic.setText("fileName.docx")
        self.text_dic.move(580, 225)
        self.text_dic.setFixedWidth(1000)

        self.btn_dic = QPushButton(self.checkingLayout)
        self.btn_dic.setText("Get File")
        self.btn_dic.move(810, 260)
        self.btn_dic.clicked.connect(self.getDic2)

        self.checkingLayout.show()

    # All Function
    # Create Function

    def getDic(self):
        createData_fun.getDic(self, self.createWindow, self.dic_name)

    def vef_codew(self, title, nim, path):
        wait = createData_fun.movingFile(self, title, nim, path)
        if wait[0] == 0:
            self.errorMsg(wait[1], self.createWindow)
        else:
            self.errorMsg(wait[1], self.createWindow)
            self.vefLayout.close()
            self.createWindow.close()

    # List Function

    def getNim(self):
        listData_fun.getList(self, self.listWindow, self.nim,
        self.input_searchDoc)
        self.listUIs()

    def getInfo(self, nims):
        data = listData_fun.getNim(self, nims)
        print(data)
        if data[0] == 0:
            self.errorMsg(data[1], self.listWindow)
        else:          25
            self.infoWindow(data[0], data[1], data[3], data[4])

    # Compare Function

    def getDic2(self):
        createData_fun.getDic(self, self.checkingLayout, self.dic_get)
        getNameFile = self.dic_get.text().split('/')
        self.text_dic.setText(getNameFile[len(getNameFile) - 1])

```

```

        doc = docx.Document(self.dic_get.text()).paragraphs
        nimCheck = compareData_fun.searchNim(self, doc)
        print(nimCheck)
        titleCheck = compareData_fun.searchTitle(self, nimCheck, doc)
        print(titleCheck)
        fileCheck = compareData_fun.getFileDB(self, nimCheck,
        titleCheck)
        self.text_dicL.setText(fileCheck.split(",") [0])
        fileDbPath =
docx.Document(f'Data/File/{nimCheck}/{fileCheck.split(",") [0]}').para
        presentase = compareData_fun.comparing(self, doc, fileDbPath)
        self.errorMsg(presentase, self.checkingLayout)

# admin verification Function

def verifyingAdmin(self, usernm, passw):
    status = createUser_fun.verification(self, usernm, passw)
    message = ""
    if status == "succes":
        checkNim = createUser_fun.checkingNim(self,
self.input_nimUser.text())
        print('allhamdulillah')
        if checkNim[0] == 0:
            self.errorMsg("Nim sudah ada / Salah memasukkan
format\n cek tahun"
                           "dan kode prodi", self.vadminLayout)
        else:
            create = createUser_fun.sendToDb(self, checkNim[0],
self.input_nameUser.text(),
self.input_codeUser.text(), checkNim[1])
            print(create)
            self.vadminLayout.close()
            self.cuserLayout.close()
    else:
        message = "failed"
        self.errorMsg("Salah masukkan password/username",
self.vadminLayout)

    print(message)

20
if __name__ == '__main__':
    app = QApplication(sys.argv)
    ex = main()
    sys.exit(app.exec_())

```

functions/compareData\_fun.py

```

from PyQt5.QtWidgets import *

def searchNim(self, paragraphs):
    print("Nimming...")
    tempList = []

```

```

with open('Data/infoUser.txt') as user:
    for nim in user:
        temp = nim.split(',')
        tempList.append(temp[0])

tempSort = sorted(tempList)
for nims in tempSort:
    for i in range(len(paragraphs)):
        if nims in paragraphs[i].text:
            return nims

def searchTitle(self, nim, parag):
    tempTitle = []
    with open(f'Data/User/{nim}.txt', 'r') as titling:
        for a in titling:
            tempTitle.append(a.split(',') [0])
        print(tempTitle)

    sortSelish = []
    for b in tempTitle:
        splittingTitle = b.split('_')[0]
        splitText = splittingTitle.split()
        print(splitText)

        dif = 0 # Selisih
        if len(parag) > 15:
            cntr = 0
            for i in range(16):
                while True:
                    if cntr >= len(parag):
                        break
                    if len(parag[cntr].text) > 0:
                        for j in splitText:
                            if j.lower() in parag[cntr].text.lower():
                                dif += 1
                                print(dif)
                                sortSelish.append([dif, splittingTitle])
                        cntr += 1
                        break
                    else:
                        cntr += 1
                if cntr > len(parag):
                    break
            else:
                cntr = 0
                for g in parag:
                    while True:
                        if cntr >= len(parag):
                            break
                        if len(parag[cntr].text) > 0:
                            for k in splitText:
                                if k.lower() in parag[cntr].text.lower():
                                    dif += 1
                                    print(dif)
                                    sortSelish.append([dif, splittingTitle])
                            cntr += 1
                            break
                        else:

```

```

        cntr += 1
    if cntr > len(parag):
        break
sorting = sorted(sortSelish)
return sorting[len(sorting) - 1][1]

def getFileDB(self, nim, title):
    with open(f'Data/User/{nim}.txt', 'r') as titles:
        for i in titles:
            if title in i:
                return i

def comparing(self, fileOutside, fileDb):
    insideDb = []
    OutsideDb = []

    for inside in fileDb:
        if len(inside.text) > 0:
            temp = inside.text
            insideDb.append(temp.split())
        else:
            pass

    for outside in fileOutside:
        if len(outside.text) > 0:
            tempO = outside.text
            OutsideDb.append(tempO.split())
        else:
            pass

    total = 0
    trues = 0
    falses = 0

    for parag in range(len(insideDb)):
        for words in range(len(insideDb[parag])):
            if len(insideDb[parag]) > 0:
                try:
                    if OutsideDb[parag][words] in insideDb[parag]:
                        insideDb[parag].pop(0)
                        total += 1
                        trues += 1
                    else:
                        total += 1
                        falses += 1
                except IndexError:
                    falses += 1
                    total += 1
                    print("astaghfirullah")
            else:
                break
    print(insideDb)
    print("allhamdulillah")

    print(int(((trues - falses)/total)*100))
    return f"Presentase Validasi File : {int(((trues - falses)/total)*100)}%"

```

	functions/createUser_fun.py
	<pre> from PyQt5.QtWidgets import * import datetime  def verification(self, usernm, passw):     if usernm == "admin" and passw == "admin":         return "succes"     else:         return "failed"  def prodiChecking(nim):     prodi = ''      if len(nim) == 7:         with open('Data/infoProdi.txt', 'r') as prods:             for prodih in prods:                 if nim[0] in prodih:                     prodi = prodih.split(',') [1]                     break                 else:                     prodi = [0, f"Tidak ada prodi {nim[0:2]}"]     return prodi      elif len(nim) == 8:         with open('Data/infoProdi.txt', 'r') as prods:             for prodih in prods:                 if nim[0:2] in prodih:                     prodi = prodih.split(',') [1]                     break                 else:                     prodi = [0, f"Tidak ada prodi {nim[0:2]}"]     return prodi  def yearChecking(nim):     year = datetime.datetime.now().strftime("%Y")      if len(nim) == 7:         if int(nim[1:3]) &lt;= int(year):             return nim[1:3]         else:             print(f"ini masih tahun {datetime.datetime.now().strftime('%Y')} ")             return "failed"      elif len(nim) == 8:         if int(nim[2:4]) &lt;= int(year):             return nim[2:4]         else:             print(f"ini masih tahun {datetime.datetime.now().strftime('%Y')} ")             return "failed" </pre>

```

def checkingNim(self, nim):
    tempNim = []
    message = 0
    prodi = ''
    with open('Data/infoUser.txt', 'r') as nima:
        for nims in nima:
            tempNim.append(nims.split(",") [0])

    if nim not in tempNim:
        message = 1
    else:
        message = 0

    if message == 1:
        prodi = prodiChecking(nim)
        if prodi[0] == 0:
            return prodi[1]
        year = yearChecking(nim)
        if year == 'failed':
            return [0, f"ini masih tahun
{datetime.datetime.now().strftime('%Y')}]"
        return [nim, f"{prodi[0]:len(prodi) - 1}]{year}"]
    else:
        return [0, 'nim sudah ada di dalam database']

def sendToDb(self, nim, name, passw, prodi, status='mahasiswa'):
    validating = 0
    with open('Data/infoUser.txt', 'r') as userR:
        for i in userR:
            validating += 1

    with open('Data/infoUser.txt', 'a') as users:
        if validating != 0:
            users.write(f'\n{nime},{name},{passw},{prodi},{status}')
        elif validating == 0:
            users.write(f'{nime},{name},{passw},{prodi},{status}')

    with open(f'Data/User/{nim}.txt', 'w') as newUser:
        return "Successfull"

```

#### functions/createData\_fun.py

```

from PyQt5.QtWidgets import *
import docx, os, shutil, datetime, string, time

def getDic(self, windowApp, textDic=QLabel):
    self.takeDic = QFileDialog(windowApp)
    opens = self.takeDic.getOpenFileName(self, 'Open File', 'D:/',
'Document (*.docx)')
    textDic.setText(opens[0])

def getNim(path):
    tempDoc = docx.Document(path)
    lenParDoc = len(tempDoc.paragraphs)

```

```

tempNimList = []
with open('Data/infoUser.txt') as user:
    for i in user:
        temp = i.split(',')
        tempNimList.append(temp[0])

tempSorted = sorted(tempNimList)
for i in tempSorted:
    for j in range(lenParDoc):
        if i in tempDoc.paragraphs[j].text:
            return i

def searchTitle(self, title, parag):
    thisTitle = title.split()
    print("Bismillah")
    if len(parag) > 15:
        print("bismillah..")
        cntr = 0
        print("bismillah..")
        for i in range(15):
            if len(parag[i].text) > 0:
                for j in thisTitle:
                    print("bismillah..")
                    if j in parag[i].text:
                        cntr += 1
                if cntr > 4:
                    return [1, 'succes']
            if cntr > 0:
                return [1, 'succes']
            elif cntr < 0:
                return [0, 'Tolong gunakan judul dokumen yang ada di dalam
file.']

    else:
        cntr = 0
        for i in parag:
            if len(i.text) > 0:
                for j in thisTitle:
                    if j in i.text:
                        cntr += 1
                if cntr > 4:
                    return [1, 'succes']
            if cntr > 0:
                return [1, 'succes']
            elif cntr < 0:
                return [0, 'Tolong gunakan judul dokumen yang ada di dalam
file.']

def searchNim(self, doc):
    docing = docx.Document(doc).paragraphs

    temp = []
    with open('Data/infoUser.txt', 'r') as nims:
        for nim in nims:
            temp.append(nim.split(',')[0])

    sorted(temp)

```

```

count = 0
for i in docing:
    for j in temp:
        for k in list(i.text):
            if k in string.digits:
                count += 1
                # Processing
            if len(j) == len(list(i.text)):
                if j in i.text:
                    return [1, j]
return [0, "Nim yang didalam dokumen tidak valid\n atau tidak dada didalam database"]

# This Night
def movingFile(self, title, nim, path):
    print(f"The title : {title}")
    print(f"Your nim : {nim}")

    vef_code = ''
    with open('Data/infoUser.txt') as user:
        for i in user:
            if nim in i:
                temp = i.split(',')
                vef_code = temp[2].strip()

    print(f"You Got Your pass : {vef_code}")
    if self.input_vef.text() == vef_code:
        print("bismillah")
        x = datetime.datetime.now()
        day = x.strftime("%y%m%d")

        newName = f"{path}/../{title}_{nim}_{day}.docx"
        print("Checking...")
        newFolder = f"Data/File/{nim}"
        createFolder(newFolder)
        print("Checking...")

        print(path)
        print(newName)

        os.rename(path, newName)
        print("Checking...")
        shutil.copy(newName, newFolder)
        print("Checking...")

        validating = 0
        documents = []
        with open(f'Data/User/{nim}.txt', 'r') as apper:
            for i in apper:
                validating += 1
                documents.append(i.split(',')[0])

        with open(f'Data/User/{nim}.txt', 'a') as appe:
            title2 = newName.split('/')[-1]
            if validating != 0:
                if title2 in documents:
                    no = 2

```

```

        while True:
            new_title = f'{title2} ({no})'
            if new_title not in documents:
                break
            no += 1
            appe.write(f'\n{n{new_title}}, {day} ')
            appe.close()
        else:
            appe.write(f'\n{title2}, {day} ')
            appe.close()
    elif validating == 0:
        appe.write(f'{title2}, {day} ')
        appe.close()

    print("Checking...")

    os.rename(newName, path)
    print("Checking...")
    print("allhamdulillah")
    return [1, f'Dokumen
{newName.split("/")[-1]} {len(newName.split("/")) - 1}] sudah dimasukkan']
    elif self.input_vez.text() != vez_code:
        return [0, 'Verify code yang anda masukkan salah\n '
                  'tidak sesuai dengan nim yang ada di dokumen'] #
    Change to pop up error message
else:
    print("Hah?!")

```

16

```

def createFolder(directory):
    try:
        if not os.path.exists(directory):
            os.makedirs(directory)
    except OSError:
        print('Error: Creating directory. ' + directory)

```

	functions/listData_fun.py
--	---------------------------

```

from PyQt5.QtWidgets import *

def getList(self, windowApp, nim, nimG):
    windowApp.close()
    self.nim = nimG.text()
    tempList = []
    with open('Data/infoUser.txt') as user:
        for i in user:
            temp = i.split(',')
            tempList.append(temp[0])
    if self.nim in tempList:
        self.nim = self.nim
    else:
        self.nim = 'default'

def getNim(self, nim):
    if nim == 'default':

```

```

        return [0, 'Tidak mendapatkan nim yang valid']

    temp = []
    with open('Data/infoUser.txt', 'r') as dataUser:
        for nims in dataUser:
            temp.append(nims.split(',') [0])

    for i in temp:
        if nim == i:
            with open('Data/infoUser.txt', 'r') as dataUser2:
                for nims2 in dataUser2:
                    if nim == nims2.split(',') [0]:
                        return nims2.split(',')

```

### 3.5 Additional Files

File ini untuk membaca nim yang ditulis dengan pola seperti 11201050 yang 11 awalnya itu adalah informatika.

	Data/infoProdi.txt
	1,Fisika 19,Matematika 3,Teknik Mesin 4,Teknik Elektro 5,Teknik Kimia 6,Teknik Material dan Metalurgi 7,Teknik Sipil 8,Perencanaan Wilayah dan Kota 9,Teknik Perkapalan 10,Sistem Informasi 11,Informatika 12,Teknik Industri 13,Teknik Lingkungan 14,Teknik Kelautan 15,Statistika 16,Arsitektur 17,Ilmu Aktuaria

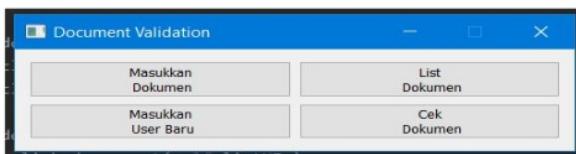
File default ini untuk mengalihkan saat pencarian nim pada list dokumen tidak valid maka akan menggunakan file ini akan memberitahukan dengan menampilkan tulisan ini di table

	Data/User/default.txt
	Nim yang anda masukkan tidak valid atau tidak ada di database, inputkan lagi

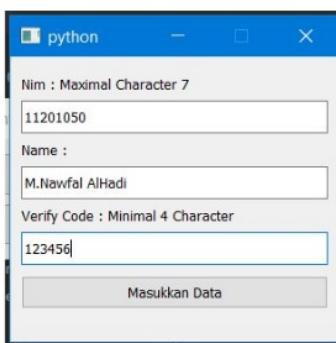
### 3.6 Run Program

#### 3.6.1 Masukkan User Baru

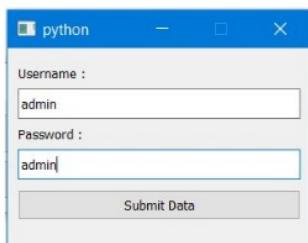
1. Pilih menu “Masukkan user baru”



2. Masukkan nim dengan minimal 7 Character dan Maximal 8 Character

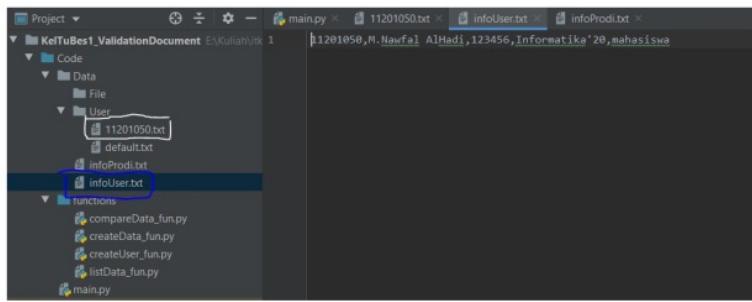


3. Masukkan username dan password admin



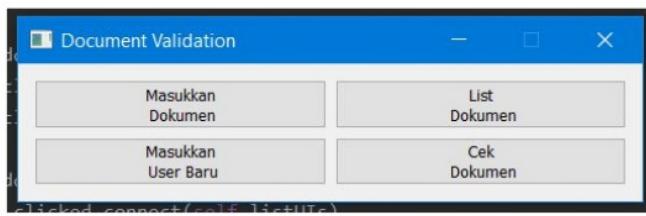
4. Kotak putih menandakan file untuk penyimpanan setiap judul yang disimpan oleh nim tersebut dan yang kotak biru adalah untuk menyimpan list semua user dengan penulisan :

Nim,nama,vefcode,prodi dan angkatan,status

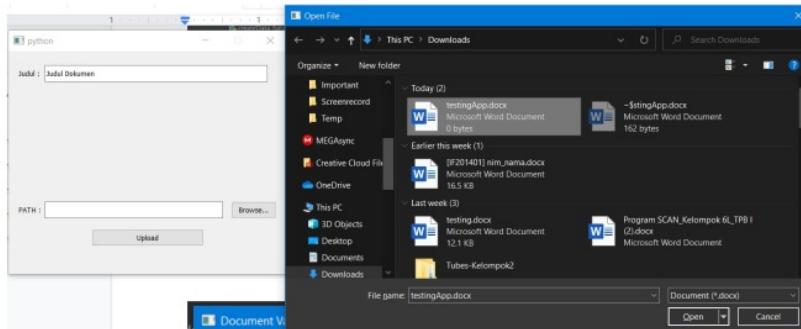


### 3.6.2 Masukkan Dokumen Baru

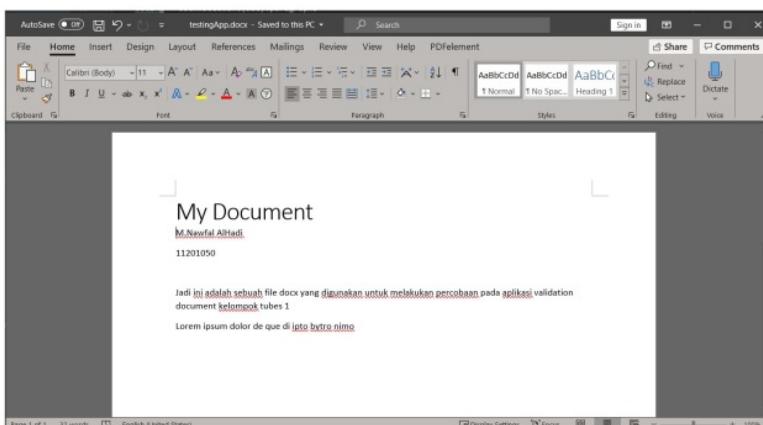
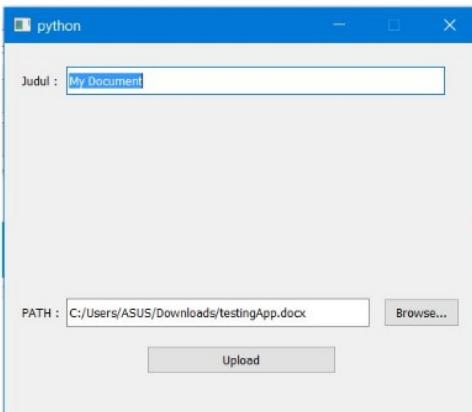
1. Pilih “Masukkan Dokumen”



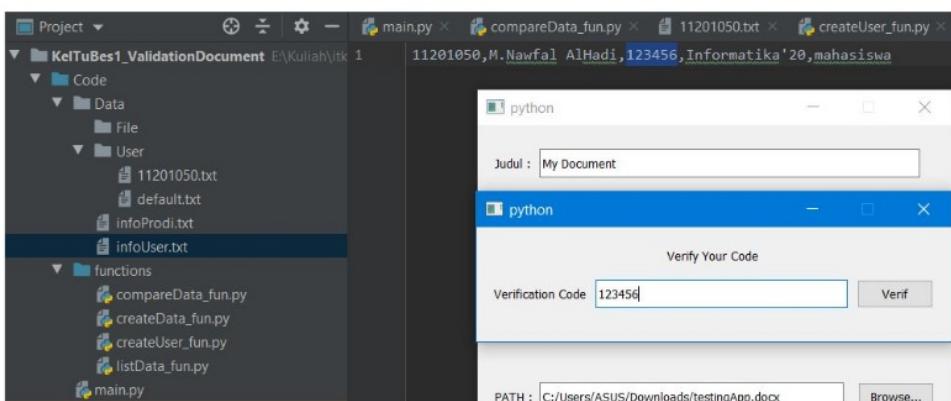
2. Tekan button “Browse..” untuk cari file, dan pilih file yang ada nim yang valid dan sudah dimasukkan ke dalam database



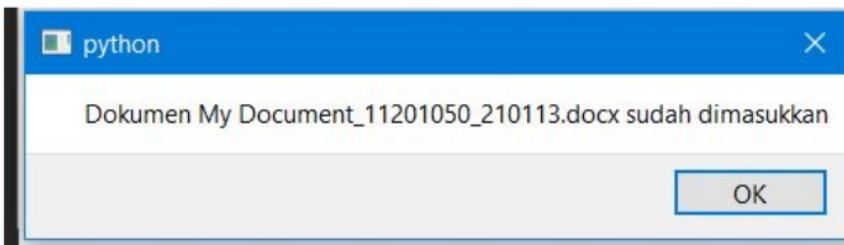
3. Ganti Judul dokumen yang sama dengan didalam filenya



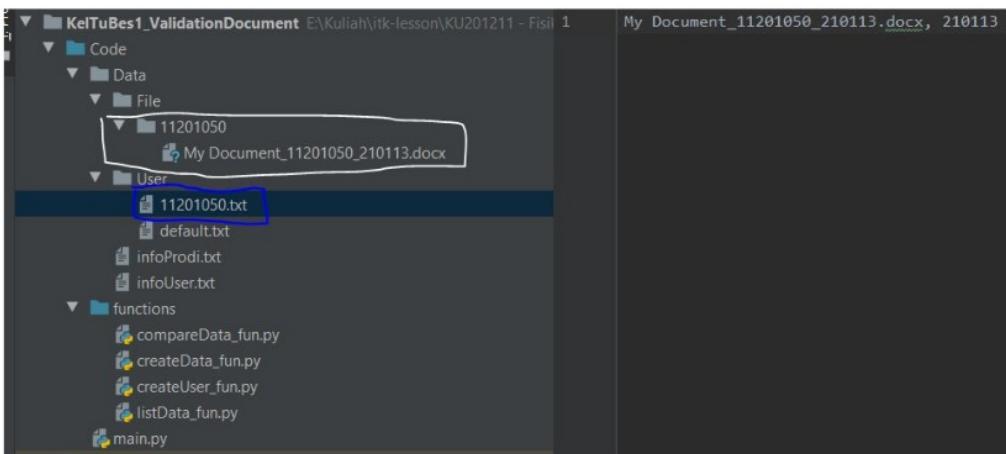
4. Sistem akan membaca nim yang didalam dokumen itu lalu meminta verifikasi code untuk memasukkan nya



5. Dokumen sudah dimasukkan ke dalam database

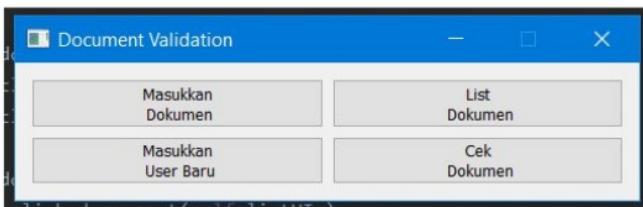


6. Pada kotak biru akan dimasukkan ke dalam file user/nimnya.txt dan memasukkan judul dan tanggal nya sedangkan kotak putih itu akan memasukkan file document yang sudah di copy lalu direname sesuai system yang dibuat.

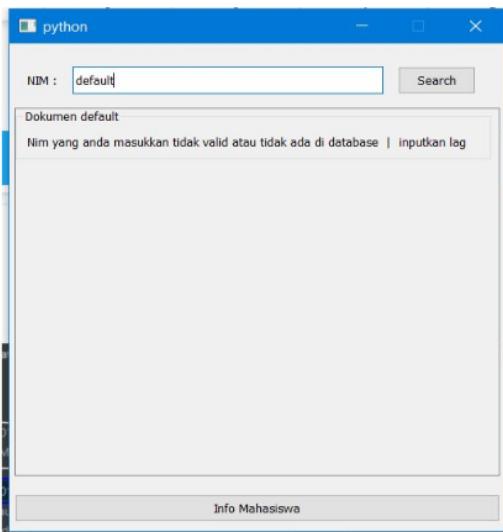


### 3.6.3 Melihat list dokumen

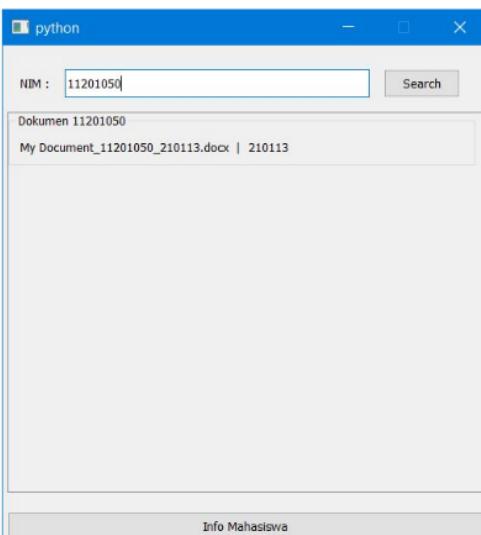
1. Pilih “List Dokumen”



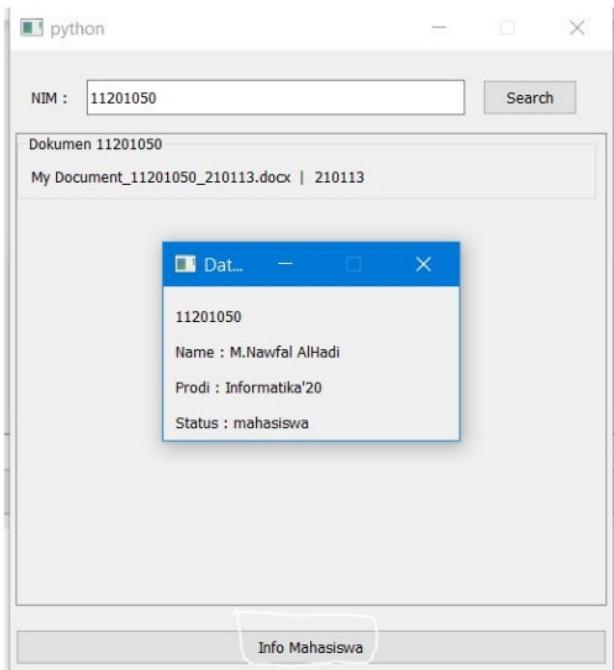
2. Pertama masuk maka akan menampilkan default dengan info kalau “nim yang anda masukkan tidak valid atau tidak ada di database”.



3. Setelah user memasukkan nim yang valid user bisa membuka info mahasiswa dan bisa melihat dokumen dokumen yang telah dimasukkan ke database oleh nim tersebut.

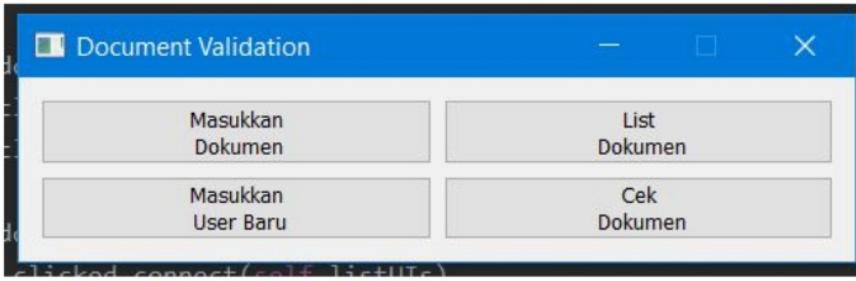


4. Jika menekan button info mahasiswa maka akan menampilkan data mahasiswa nim tersebut yang disimpan pada User/nim.txt untuk menyimpan judul tadi

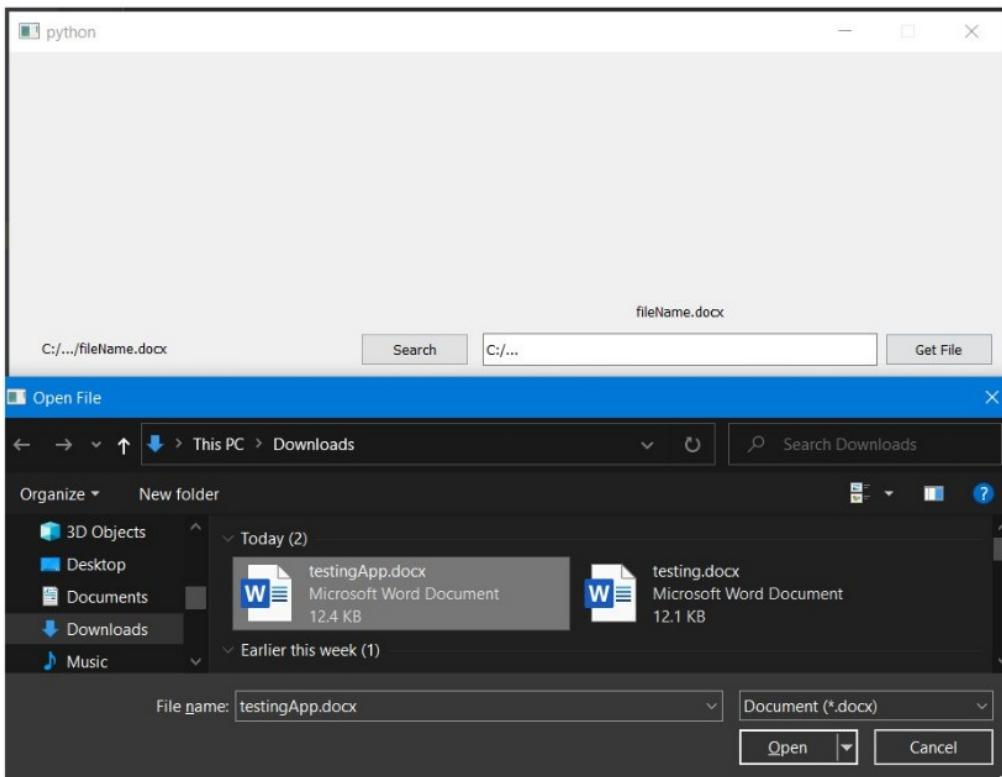


#### 3.6.4 Cek Dokumen

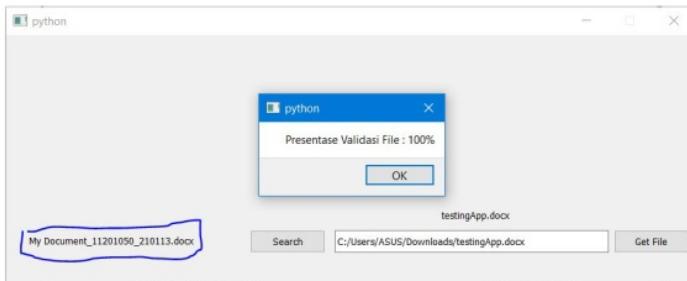
1. Pilih "Cek Dokumen"



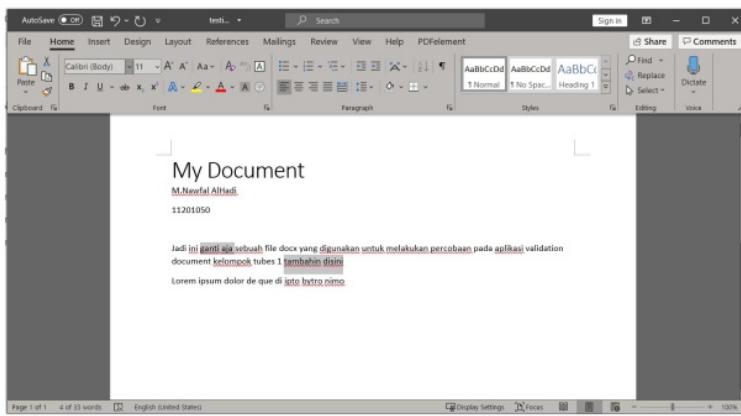
2. Tekan button get file dan cari file yang ingin dibandingkan dengan yang ada di database, harus ada nim valid dan didalamnya berjudul sama dengan yang ada di dalam database



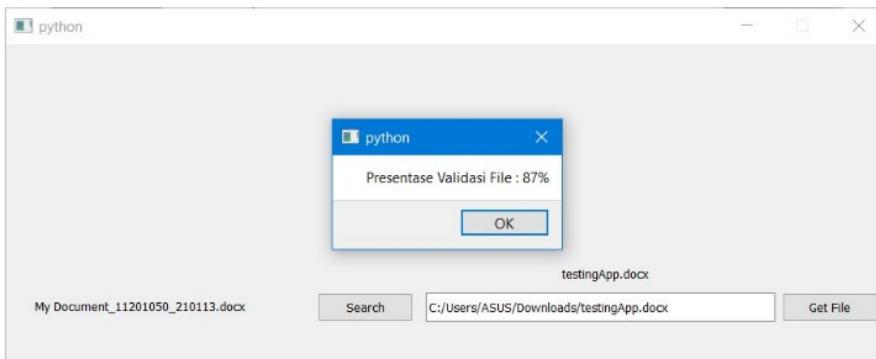
3. System akan membandingkan dan mempersenkan hasil perbandingan nya dengan formula (wordSama/totalCompare \* 100%)



4. Lalu coba tambahin dan ganti pada file nya untuk mendapatkan presentase yang berbeda



5. Dan mendapatkan persen yang lebih kecil karena file nya sudah di ubah ubah.



## **Bab IV PENUTUP**

### 4.1 Kesimpulan

Program Document Validation merupakan program yang dibuat untuk dapat menangani pemalsuan dan keamanan data dalam mencari informasi yang tersimpan di dalam data diantaranya :

1. Membantu untuk menyimpan data dokumen mahasiswa atau karyawan yang tersimpan dengan aman
2. Membantu pengecekan dokumen dalam dengan berupa hasil persentase perbandingan dokumen yang tersedia di database
3. Membantu untuk melihat data dokumen yang sesuai pada mahasiswa dengan memasukkan nim atau id, dan pada karyawan menggunakan identitas(id) karyawan.

### 4.2 Saran

Maraknya pemalsuan pada dokumen yang tersimpan dan pengamanan data dokumen, diharapkan dapat mengurangi aksi tersebut sehingga para penggunanya merasa aman ketika memasukkan data privasinya ke suatu sistem.

## **Daftar Pustaka**

- Sari, R. F., & Azizah, N. (2020). Sistem Validasi Keaslian Dokumen Digital Berbasis QR-Code. *JurTI (Jurnal Teknologi Informasi)*, 4(2), 321-327.
- <https://www.pythonindo.com/penangananeksepsi/#:~:text=Python%20menyediakan%20metode%20penanganan%20eksepsi,yang%20kemungkinan> (akses pada tanggal 1 Januari 2021)
- Perkasa, T. R. (2014). *TA: Rancang Bangun Pendekripsi Gerak Menggunakan Metode Image Subtraction pada Single Board Computer (SBC)* (Doctoral dissertation, Institut Bisnis dan Informatika Stikom Surabaya).
- Sulistyo-Basuki, M. A. D., & Memahami, P. (2003). mengelola Informasi dan Dokumen. *Jakarta: Gramedia*.
- Widarma, A., & Kumala, H. (2018). Perancangan aplikasi gaji karyawan pada pt. pp london sumatra indonesia tbk. gunung melayu estate-kabupaten asahan. *JurTI (Jurnal Teknologi Informasi)*, 1(2), 166-173.
- <http://hikikootaku.blogspot.com/2014/10/pengertian-write-writeln-read-dan.html> ( diakses pada tanggal 21 desember 2020 )
- <https://ngodingdata.com/struktur-list-di-python/> ( diakses pada tanggal 21 Desember 2020)
- <https://kopiding.in/perulangan-python-dasar/> ( diakses pada tanggal 21 Desember 2020 )
- <https://id.wikipedia.org/>
- <https://id.wikipedia.org/wiki/PyQt> ( diakses pada tanggal 21 Desember 2020)
- <https://www.pythonindo.com/menjumlahkan-dua-bilangan/> ( diakses pada tanggal 27 Desember 2020 )
- <https://www.petanikode.com/python-variabel-dan-tipe-data/> ( diakses pada tanggal 27 Desember 2020 )
- <https://www.pythonindo.com/string/> ( diakses pada tanggal 27 Desember 2020 )

## Lampiran

### Biodata Anggota Kelompok

Nama	:	Jackson Pangesturi	
NIM	:	10201050	
Program Studi	:	Sistem Informasi	
Nomor Telepon	:	085290453275	
E-mail	:	jacksonpangesturi@gmail.com	
Tugas Dalam Proyek	:	Flowchart, Pemeriksaan Teori dalam Laporan, Penulisan Pembahasan, Nyari Konsep.	

Nama	:	Muhammad Nur Rahman	
NIM	:	10201062	
Program Studi	:	Sistem Informasi	
Nomor Telepon	:	081250197684	
E-mail	:	10201062@student.itk.ac.id	
Tugas Dalam Proyek	:	Flowchart, Pemeriksaan Teori dalam Laporan, Penulisan Pembahasan, Nyari Konsep.	

Nama	:	Muhammad Huzaimi	
NIM	:	11201062	
Program Studi	:	Informatika	
Nomor Telepon	:	0895409066345	
<i>E-mail</i>	:	11201062@students.itk.ac.id	
Tugas Dalam Proyek	:	Bantu code, flowchart, nyari inti konsep yang abstrak	

Nama	:	M. Nawfal Al Hadi	
NIM	:	11201050	
Program Studi	:	Informatika	
Nomor Telepon	:	082143883436	
<i>E-mail</i>	:	11201050@students.itk.ac.id	
Tugas Dalam Proyek	:	Focus Code, Algoritma, & Pseudocode	

Nama	:	Ahmad Gilang Ramadhan	
NIM	:	12201002	
Program Studi	:	Teknik Industri	
Nomor Telepon	:	081346361073	
E-mail	:	12201002@student.itk.ac.id	
Tugas Dalam Proyek	:	Memperbaiki kesalahan dalam laporan	

Nama	:	Rismawati	
NIM	:	12201058	
Program Studi	:	Teknik Industri	
Nomor Telepon	:	085753059979	
E-mail	:	12201058@student.itk.ac.id	
Tugas Dalam Proyek	:	Pembuatan Laporan	

Nama	:	Syarifah Chairunnisa
NIM	:	12201066
Program Studi	:	Teknik Industri
Nomor Telepon	:	085386429931
<i>E-mail</i>	:	12201066@student.itk.ac.id
Tugas Dalam Proyek	:	Pembuatan Laporan & Powerpoint



## Lembar Asistensi

The screenshot shows a Google Sheets document with the following details:

**Document Title:** LPTB ALPRO E

**Sheet Name:** Asistensi Tugas besar (Konsep, dan latar belakang)

**Table 1: Lembar Progres Tugas Besar (LPTB)**

Mata Kuliah	Algoritme Pemrograman	Semester	Ganjil/Genap
Kode MK	-	Tahun Akademik	2020/2021
Kredit	3 SKS	Dosen Pengampu	Boly Mugi Pratama, S.Si, M.Han
		Asisten	Indra Lukman Hakim Rani Yuniar Putriyanti

**Table 2: Nama Anggota Kelompok**

NIM	NAMA
10201050	Jackson Pangesturi
10201052	Muhammad Nur Rahman
11201050	M Nawfal AlHadi
11201062	Muhammad Huzaimi
12201068	Ramawati
12201069	Syarifah Chairunisa
12201080	Ahmad Giang Ramadhan

**Table 3: Log of Attendance Entries**

Hari/Tanggal	Materi/Pokok Bahasan	Tanda Tangan Asisten	Tanda Tangan Mahasiswa	Keterangan
5 Desember 2020	Asistensi Tugas besar (Konsep, dan latar belakang)	[Signature]	[Signature]	
3 Desember 2020	Asistensi progres program dan proposal	[Signature]	[Signature]	
1 Januari 2021	Asistensi Laporan dan program	[Signature]	[Signature]	

**Bottom Navigation:** Kelompok 1, Kelompok 2, Kelompok 3, Kelompok 4, Kelompok 5, Kelompok 6

## **Lembar Pertanyaan**

Pertanyaan 1

Apakah hanya bisa file berupa docs saja, tidak bisakah memakai file yang lain seperti pdf atau zip?

Dari : Satya Yazid Rizqullah NIM 12201062 Kelompok 6

Jawaban Dari : M. Nawfal Alhadi (11201050)

“Saya membuat file yang bisa dimasukkan hanya untuk file docx, karena biasanya sebelum file fix dan di konversi ke pdf, orang mungkin akan cek dulu dengan format docx jika kurang memuaskan maka orang masih akan merubah-rubah isinya pada file.”

# Kelompok 1

## ORIGINALITY REPORT



## PRIMARY SOURCES

---

1	<a href="http://student.blog.dinus.ac.id">student.blog.dinus.ac.id</a> Internet Source	3%
2	<a href="http://github.com">github.com</a> Internet Source	3%
3	<a href="http://www.pythonindo.com">www.pythonindo.com</a> Internet Source	2%
4	<a href="http://repository.unhas.ac.id">repository.unhas.ac.id</a> Internet Source	2%
5	<a href="http://www.coursehero.com">www.coursehero.com</a> Internet Source	2%
6	<a href="http://id.wikipedia.org">id.wikipedia.org</a> Internet Source	1%
7	Submitted to Universitas Jenderal Achmad Yani Student Paper	1%
8	<a href="http://media.neliti.com">media.neliti.com</a> Internet Source	1%
9	<a href="http://lib.ui.ac.id">lib.ui.ac.id</a> Internet Source	1%

---

10	kelompok14ti2013.blogspot.com Internet Source	1 %
11	docplayer.info Internet Source	1 %
12	antoken.wordpress.com Internet Source	1 %
13	kopiding.in Internet Source	1 %
14	ngodingdata.com Internet Source	<1 %
15	reshaaa-dchasha.blogspot.com Internet Source	<1 %
16	Submitted to University of Messina Student Paper	<1 %
17	Submitted to Indian Institute of Technology, Madras Student Paper	<1 %
18	jurupenaseorangjurnalis.blogspot.com Internet Source	<1 %
19	addinaldollar.blogspot.com Internet Source	<1 %
20	Submitted to University of Leeds Student Paper	<1 %
	wayanrudiarta.blogspot.com	

21

Internet Source

<1 %

22

[bugs.launchpad.net](#)

Internet Source

<1 %

23

[library.gunadarma.ac.id](#)

Internet Source

<1 %

24

[fr.scribd.com](#)

Internet Source

<1 %

25

[www.scribd.com](#)

Internet Source

<1 %

26

[ilearning.me](#)

Internet Source

<1 %

27

[widuri.raharja.info](#)

Internet Source

<1 %

28

[www.configrouter.com](#)

Internet Source

<1 %

Exclude quotes

On

Exclude matches

Off

Exclude bibliography

On