

```
1. from tkinter import*
2. from tkinter import ttk
3. from tkinter import messagebox
4. from PIL import ImageTk
5. import pymysql
6.
7.
8. class Login:
9.     def __init__(self, root):
10.         self.root = root
11.         self.root.title = ("LOGIN AND REGISTER SYSTEM FOR APPS")
12.         self.root.geometry("1366x700+0+0")
13.         # self.root.resizable(False,False)
14.         self.loginform()
15.         # self.appscreen()
16.
17.
18.     def loginform(self):
19.         Frame_login = Frame(self.root, bg="white")
20.         Frame_login.place(x=0, y=0, height=700, width=1366)
21.
22.         self.img = ImageTk.PhotoImage(file="gambar2.JPG")
23.         img = Label(Frame_login, image=self.img).place(
24.             x=0, y=0, height=700, width=1366)
25.
26.         frame_input_login = Frame(self.root, bg='white')
27.         frame_input_login.place(x=320, y=130, height=450, width=350)
28.
29.         label1 = Label(frame_input_login, text="Login Disini", font=(
30.             'impact', 32, 'bold'), fg="black", bg="white") # Label yang bertuliskan login disini
31.         label1.place(x=75, y=20)
32.
33.         label2 = Label(frame_input_login, text="User Name", font=(
34.             'Goudy old style', 20, 'bold'), fg="orangered", bg="white") # Label yang bertuliskan
35.         Username
36.         label2.place(x=30, y=95)
37.
38.         self.email_text = Entry(frame_input_login, font=(
39.             "times new roman", 15, "bold"), bg='lightgray') # entry email atau username
40.         self.email_text.place(x=30, y=145, width=270, height=35)
41.
42.         label3 = Label(frame_input_login, text="Password", font=(
43.             'Goudy old style', 20, 'bold'), fg="orangered", bg="white") # Label password
44.         label3.place(x=30, y=195)
45.
46.         self.password_text = Entry(frame_input_login, show='*', font=(
47.             "times new roman", 15, "bold"), bg='lightgray') # entry password
48.         self.password_text.place(x=30, y=245, width=270, height=35)
49.
50.         btn1 = Button(frame_input_login, text="Lupa Password?", command=self.forget_pass,
51.             cursor='hand2', font=('calibri', 10), bg='white', fg='black', bd=0) # tombol untuk
52.         Lupa password
53.         btn1.place(x=125, y=305)
54.
55.         btn2 = Button(frame_input_login, text="LOGIN", command=self.login, cursor='hand2', font=(
```

```
54.         'times new roman', 15), bg='orangered', fg='white', bd=0, width=15, height=1)
55.     btn2.place(x=90, y=340)
56.
57.     btn3 = Button(frame_input_login, text="Belum Registrasi? Register dulu...",
58. command=self.register,
59.         cursor='hand2', font=('calibri', 10), bg='white', fg='black', bd=0) # tombol untuk
60.     register
61.     btn3.place(x=100, y=390)
62.
63.     def login(self):
64.         if self.email_text.get() == "" or self.password_text.get() == "":
65.             messagebox.showerror(
66.                 "Error", "Username dan Password Masih Kosong", parent=self.root)
67.         else:
68.             try:
69.                 con = pymysql.connect(
70.                     host='localhost', user='root', password='', database='aplikasidatamahasiswa')
71.                 cur = con.cursor()
72.                 cur.execute('select * from user where username=%s and password=%s',
73.                     (self.email_text.get(), self.password_text.get()))
74.                 row = cur.fetchone()
75.                 if row == None:
76.                     messagebox.showerror(
77.                         'Error', 'Username Atau Password Salah', parent=self.root)
78.                     self.login_clear()
79.                     self.email_text.focus()
80.                 else:
81.                     self.tambahData()
82.                     con.close()
83.             except Exception as es:
84.                 messagebox.showerror(
85.                     'Error', f'Error Due to : {str(es)}', parent=self.root)
86.
87.     def login_clear(self):
88.         self.email_text.delete(0, END)
89.         self.password_text.delete(0, END)
90.
91.     def tambahData(self):
92.         Frame_tambah_data = Frame(self.root, bg="white")
93.         # membuat frame utama baru sama seperti di Login form sebelum nya
94.         Frame_tambah_data.place(x=0, y=0, height=700, width=1366)
95.
96.         frame_input_tambahData = Frame(self.root, bg='white')
97.         # membuat frame inputan atau entry baru sama seperti di Login form sebelum nya
98.         frame_input_tambahData.place(x=75, y=60, height=700, width=1260)
99.
100.        label1 = Label(Frame_tambah_data, text="Aplikasi Data Mahasiswa",
101.            fg="black", bg="#70aacf", font=("times new roman", 35, "bold")) # membuat Label
102.        baru sama seperti di Login form sebelum nya
103.        label1.place(x=0, y=0, width=1366)
104.
105.        self.img = ImageTk.PhotoImage(file="11.jpg")
106.        img = Label(frame_input_tambahData, image=self.img).place(
```

```
105.         x=700, y=70, height=300, width=500) # membuat tampilan gambar baru sama seperti di login
form sebelum nya
106.
107.         label2 = Label(frame_input_tambahData, text="Nomor Induk Mahasiswa", font=(
108.             'times new roman', 20, 'bold'), fg="orangered", bg="white") # membuat Label baru sama
seperti di login form sebelum nya
109.         label2.place(x=30, y=150)
110.
111.         self.entry = Entry(frame_input_tambahData, font=(
112.             'times new roman', 15, 'bold'), bg="lightgray") # nim
113.         # membuat entry baru sama seperti di login form sebelum nya
114.         self.entry.place(x=370, y=150, width=270, height=35)
115.
116.         label3 = Label(frame_input_tambahData, text="Nama Lengkap", font=(
117.             'times new roman', 20, 'bold'), fg="orangered", bg="white") # membuat Label baru sama
seperti di login form sebelum nya
118.         label3.place(x=30, y=195)
119.
120.         self.entry2 = Entry(frame_input_tambahData, font=(
121.             'times new roman', 15, 'bold'), bg="lightgray") # nama
122.         # membuat entry baru sama seperti di login form sebelum nya
123.         self.entry2.place(x=370, y=195, width=270, height=35)
124.
125.         label4 = Label(frame_input_tambahData, text="Tempat & Tanggal Lahir", font=(
126.             'times new roman', 20, 'bold'), fg="orangered", bg='white') # membuat Label baru sama
seperti di login form sebelum nya
127.         label4.place(x=30, y=240)
128.
129.         self.entry3 = Entry(frame_input_tambahData, font=(
130.             'times new roman', 15, 'bold'), bg="lightgray") # tanggal Lahir
131.         # membuat entry baru sama seperti di login form sebelum nya
132.         self.entry3.place(x=370, y=240, width=270, height=35)
133.
134.         label5 = Label(frame_input_tambahData, text="Jenis Kelamin", font=(
135.             'times new roman', 20, 'bold'), fg="orangered", bg='white') # membuat Label baru sama
seperti di login form sebelum nya
136.         label5.place(x=30, y=285)
137.
138.         j = ["Laki-Laki", "Perempuan"]
139.         self.jenisKel_entry = ttk.Combobox(
140.             frame_input_tambahData, values=j, font=("times new roman", 15, "bold"))
141.         # combobox penjelasan pada awal sintax program ini
142.         self.jenisKel_entry.place(x=370, y=285, width=270, height=35)
143.         self.jenisKel_entry.config(state="readonly")
144.         self.jenisKel_entry.set("Pilih Jenis Kelamin")
145.
146.         label6 = Label(frame_input_tambahData, text="Jurusan", font=(
147.             'times new roman', 20, 'bold'), fg="orangered", bg='white') # membuat Label baru sama
seperti di login form sebelum nya
148.         label6.place(x=30, y=330)
149.
150.         n = ["Matematika", "Informatika", "Teknik Mesin", "Teknik Electro"]
151.         self.jurusan_entry = ttk.Combobox(
152.             frame_input_tambahData, values=n, font=("times new roman", 15, "bold")) # jurusan
153.         # combo box untuk memilih jurusan
```

```

154.         self.jurusan_entry.place(x=370, y=330, width=270, height=35)
155.         self.jurusan_entry.config(state="readonly")
156.         self.jurusan_entry.set("Pilih Jurusan")
157.
158.         label7 = Label(frame_input_tambahData, text="Alamat", font=(
159.             'times new roman', 20, 'bold'), fg="orangered", bg='white') # membuat Label baru sama
seperti di login form sebelum nya
160.         label7.place(x=30, y=375)
161.
162.         self.entry6 = Entry(frame_input_tambahData, font=(
163.             'times new roman', 15, 'bold'), bg="lightgray") # alamat
164.         # membuat entry baru sama dengan login form sebelum nya
165.         self.entry6.place(x=370, y=375, width=800, height=35)
166.
167.         label8 = Label(frame_input_tambahData, text="Nomor Telepon", font=(
168.             'times new roman', 20, 'bold'), fg="orangered", bg='white') # membuat Label baru sama
seperti di login form sebelum nya
169.         label8.place(x=30, y=420)
170.
171.         self.entry7 = Entry(frame_input_tambahData, font=(
172.             'times new roman', 15, 'bold'), bg="lightgray") # no telepon
173.         # membuat entry baru sama dengan login form sebelum nya
174.         self.entry7.place(x=370, y=420, width=270, height=35)
175.
176.         btn1 = Button(frame_input_tambahData, text="Simpan Data", command=self.simpan, cursor="hand2",
font=(
177.             "times new roman", 15), fg="white", bg="orangered", bd=0, width=15, height=1) # membuat
tombol simpan data baru
178.         btn1.place(x=30, y=550)
179.
180.         btn2 = Button(frame_input_tambahData, text="Lihat Data", command=self.lihat, cursor="hand2",
font=(
181.             "times new roman", 15), fg="white", bg="orangered", bd=0, width=15, height=1) # membuat
tombol Lihat data
182.         btn2.place(x=260, y=550)
183.
184.         btn3 = Button(Frame_tambah_data, text="Logout", command=self.loginform, cursor="hand2", font=(
185.             "times new roman", 15), fg="white", bg="orangered", bd=0, width=15, height=1) # membuat
tombol Logout
186.         btn3.place(x=1170, y=10)
187.
188.         def simpan(self):
189.             if self.entry.get() == "" or self.entry2.get() == "" or self.entry3.get() == "" or
self.jenisKel_entry.get() == "Pilih Jenis Kelamin" or self.jurusan_entry.get() == "Pilih Jurusan" or
self.entry6.get() == "" or self.entry7.get() == "":
190.                 messagebox.showerror("Error", "Semua Field Harus di Isi")
191.             else:
192.                 try:
193.                     con = pymysql.connect(
194.                         host='localhost', user='root', password='', database='aplikasidatamahasiswa') #
koneksi database pythongui
195.                     # menyiapkan tempat berupa array untuk data data yang ada di dalam database
196.                     cur = con.cursor()
197.                     cur.execute("select * from data where nim=%s",

```

```

198.         self.entry.get()) # pengekseskuan data atau menselect atau memilih semua
data yang ada di dalam database dan kemudian di cocokan dengan entry nim
199.         row = cur.fetchone() # kemudian di ambil data nya tidak semua
200.         if row != None: # jika data nim yang di ketikan oleh user sudah ada di dalam data base
maka akan muncul pesan eror seperti di bawah
201.             messagebox.showerror(
202.                 "Error", "Nim Sudah Tendaftar,Silahkan Masukkan Dengan Nomor Induk Mahasiswa
Lainnya", parent=self.root)
203.         self.entry.focus() # artinya Letak pengetikan atau tempat input oleh user di
fokuskan ke entry nim yang sudah ada di dalam database tadi agar user tidak menggunakan nim tersebut dan
dapat membuat nim yang baru
204.         con.close() # perintah ini untuk menutup koneksi dari program ke database
205.         # jika nim yang dimasukkan di enrtty nim oleh user tidak sama dengan data yang ada di
database atau nim baru maka
206.         # program akan membuat koneksi dengan database yang bernama pythongui seperti syntax di
bawah
207.         else:
208.             con = pymysql.connect(
209.                 host='localhost', user='root', password='', database='aplikasidatamahasiswa')
210.             cur = con.cursor() # menyiapkan tempat untuk data yang akan di ambil dari database
211.             cur.execute("Insert into
data(nim,nama,tanggalTempat,jenisKelamin,jurusan,alamat,telepon)values('%s','%s','%s','%s','%s','%s','%s')
% (
212.                 self.entry.get(), self.entry2.get(), self.entry3.get(),
self.jenisKel_entry.get(), self.jurusan_entry.get(), self.entry6.get(), self.entry7.get())
213.                 con.commit()
214.                 con.close()
215.                 # laporan kepada user jika datanya sudah masuk atau sudah di simpan ke dalam database
216.                 messagebox.showinfo(
217.                     "Information", "Tambah Data Mahasiswa Berhasil", parent=self.root)
218.                 # memanggil function SimpanDataClear untuk menghapus inputan dari user sebelum nya
219.                 self.SimpanDataClear()
220.                 self.entry.focus()
221.
222.         except Exception as es:
223.             messagebox.showerror(
224.                 "Error", f"Error due to:{str(es)}", parent=self.root)
225.
226.     def SimpanDataClear(self): # jika di panggil maka function ini akan di eksekusi
227.         # nama variabel Entry akan di hapus inputan dari user sebelum nya
228.         self.entry.delete(0, END)
229.         # nama variabel Entry2 akan di hapus inputan dari user sebelum nya
230.         self.entry2.delete(0, END)
231.         # nama variabel Entry3 akan di hapus inputan dari user sebelum nya
232.         self.entry3.delete(0, END)
233.         # nama variabel jenisKel_entry yang berupa combo box akan di kembalikan ke menu awal yakni "pilih
gender"
234.         self.jenisKel_entry.set("Pilih Jenis Kelamin")
235.         # nama variabel jurusan_entry yang berupa combo box akan di kembalikan ke menu awal yakni "pilih
jurusan"
236.         self.jurusan_entry.set("Pilih Jurusan")
237.         # nama variabel Entry6 akan di hapus inputan dari user sebelum nya
238.         self.entry6.delete(0, END)
239.         # nama variabel Entry7 akan di hapus inputan dari user sebelum nya
240.         self.entry7.delete(0, END)

```

```

241.
242.     def lihat(self):
243.
244.         # nama lengkap, tanggal lahir, tempat lahir, jenis kelamin, jurusan, alamat, no telepon,
245.         # pembuatan window utama di function ini
246.         self.frame = Frame(self.root, bg="white", relief=FLAT)
247.         self.frame.place(x=0, y=0, width=1366, height=700)
248.
249.         # self.heading_image=PhotoImage(file=)
250.         self.heading_label = Label(self.root, text="Aplikasi Data Mahasiswa", fg="black", bg="#70aacf",
font=(
251.             "times new roman", 35, "bold")) # pembuatan label untuk window ini
252.         self.heading_label.place(x=0, y=0, width=1530)
253.         # self.heading_label.config(image=self.heading_image, compound=LEFT)
254.
255.         # pembuatan window untuk beberapa entry dan button dalam window utama dalam function ini
256.         self.frame_input_lihatData = Frame(
257.             self.root, bg="#c5ded0", relief=FLAT)
258.         self.frame_input_lihatData.place(x=0, y=60, width=1366, height=330)
259.
260.         self.nim = Label(self.frame_input_lihatData, text="NIM : ", font=(
261.             "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label nim
262.         self.nim.place(x=20, y=50)
263.
264.         self.nim_entry = Entry(self.frame_input_lihatData, font=(
265.             "times new roman", 15, "bold"), relief=RIDGE) # pembuatan entry nim
266.         self.nim_entry.place(x=20, y=90)
267.
268.         self.nama = Label(self.frame_input_lihatData, text="Nama Lengkap :", font=(
269.             "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label nama lengkap
270.         self.nama.place(x=300, y=50)
271.
272.         self.nama_entry = Entry(self.frame_input_lihatData, font=(
273.             "times new roman", 15, "bold"), relief=RIDGE) # pembuatan entry nama lengkap
274.         self.nama_entry.place(x=300, y=90, width=250)
275.
276.         self.tanggal = Label(self.frame_input_lihatData, text="Tempat & Tanggal Lahir :", font=(
277.             "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label tanggal
278.         self.tanggal.place(x=610, y=50)
279.
280.         self.tanggal_entry = Entry(self.frame_input_lihatData, font=(
281.             "times new roman", 15, "bold"), relief=RIDGE) # pembuatan entry tanggal
282.         self.tanggal_entry.place(x=610, y=90, width=250)
283.
284.         self.jenisKel = Label(self.frame_input_lihatData, text="Jenis Kelamin :", font=(
285.             "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label jenis kelamin
286.         self.jenisKel.place(x=20, y=130)
287.
288.         j = ["Laki-Laki", "Perempuan"]
289.         self.jenisKel_entry = ttk.Combobox(
290.             self.frame_input_lihatData, values=j, font=("times new roman", 15, "bold")) # pembuatan
combobox jenis kelamin
291.         self.jenisKel_entry.place(x=20, y=170, width=250)
292.         self.jenisKel_entry.config(state="readonly")
293.         self.jenisKel_entry.set("Pilih Jenis Kelamin")

```

```
294.
295.     self.jurusan = Label(self.frame_input_lihatData, text="Jurusan : ", font=(
296.         "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label jurusan
297.     self.jurusan.place(x=300, y=130)
298.
299.     n = ["Matematika", "Informatika", "Teknik Mesin", "Teknik Electro"]
300.     self.jurusan_entry = ttk.Combobox(
301.         self.frame_input_lihatData, values=n, font=("times new roman", 15, "bold")) # pembuatan
combobox jurusan
302.     self.jurusan_entry.place(x=300, y=170, width=250)
303.     self.jurusan_entry.config(state="readonly")
304.     self.jurusan_entry.set("Pilih Jurusan")
305.
306.     self.telpon = Label(self.frame_input_lihatData, text="Nomor Telpon : ", font=(
307.         "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label telepon
308.     self.telpon.place(x=610, y=130)
309.
310.     self.telpon_entry = Entry(self.frame_input_lihatData, font=(
311.         "times new roman", 15, "bold"), relief=RIDGE) # pembuatan entry telepon
312.     self.telpon_entry.place(x=610, y=170, width=250)
313.
314.     self.alamat = Label(self.frame_input_lihatData, text="Alamat :", font=(
315.         "times new roman", 15, "bold"), fg="black", bg="#c5ded0") # pembuatan label alamat
316.     self.alamat.place(x=20, y=220)
317.
318.     self.alamat_entry = Entry(self.frame_input_lihatData, font=(
319.         "times new roman", 15, "bold"), relief=RIDGE) # pembuatan entry alamat
320.     self.alamat_entry.place(x=20, y=250, width=845)
321.
322.     self.id_entry = Entry(self.frame_input_lihatData, font=(
323.         "times new roman", 15, "bold")) # pembuatan entry id
324.
325.     self.updateButton = Button(self.frame_input_lihatData, cursor="hand2", text="Edit Data", font=(
326.         "times new roman", 15, "bold"), fg="black", bg="#70aacf", bd=4, relief=GROOVE,
command=lambda: self.update_student()) # pembuatan tombol update
327.     self.updateButton.place(x=890, y=80, width="190")
328.
329.     self.deleteButton = Button(self.frame_input_lihatData, cursor="hand2", text="Hapus Data", font=(
330.         "times new roman", 15, "bold"), fg="black", bg="#70aacf", bd=4, relief=GROOVE,
command=lambda: self.delete_student()) # pembuatan tombol delete
331.     self.deleteButton.place(x=890, y=140, width="190")
332.
333.     self.clearButton = Button(self.frame_input_lihatData, cursor="hand2", text="Refresh", font=(
334.         "times new roman", 15, "bold"), fg="black", bg="#70aacf", bd=4, relief=GROOVE,
command=lambda: self.UpdateDataClear()) # pembuatan tombol clear
335.     self.clearButton.place(x=890, y=200, width="190")
336.
337.     self.backButton = Button(self.frame_input_lihatData, cursor="hand2", text="Kembali", font=(
338.         "times new roman", 15, "bold"), fg="white", bg="orangered", bd=4, relief=GROOVE,
command=lambda: self.tambahData()) # pembuatan tombol back atau kembali
339.     self.backButton.place(x=890, y=260, width="190")
340.
341.     # membuat frame baru atau window baru di dalam function ini
342.     self.frame3 = Frame(self.root, relief=FLAT, bg="#c5ded0")
343.     self.frame3.place(x=10, y=400, width=1350, height=330)
```



```

344.
345.     # membuat frame baru untuk head bagian pencarian atau window baru di dalam function ini
346.     self.subf = Frame(self.frame3, relief=FLAT, bg="white")
347.     self.subf.place(x=5, y=5, width=1350, height=55)
348.
349.     self.search_by = Label(self.subf, text="Cari Data", font=(
350.         "times new roman", 20, "bold"), fg="black", bg="white") # membuat Label search By
351.     self.search_by.place(x=100, y=10)
352.
353.     k = ["NIM", "Nama Lengkap", "Jurusan"]
354.     self.scombo = ttk.Combobox(
355.         self.subf, value=k, font=("times new roman", 18, "bold")) # membuat combo box pemilihan
option dalam mencari data
356.     self.scombo.place(x=250, y=10, width=200)
357.     self.scombo.set("Berdasarkan")
358.     self.scombo.config(state="readonly")
359.
360.     self.search_entry = Entry(self.subf, font=(
361.         "times new roman", 18, "bold"), relief=RIDGE, bg="#697e80", fg="white") # membuat entry
search untuk mengimputkan data apa yang di cari oleh user yang di gunakan oleh program untuk mencari data
di dalam database
362.     self.search_entry.place(x=470, y=10, width=300)
363.
364.     self.search_button = Button(self.subf, cursor="hand2", text="Cari", font=(
365.         "times new roman", 15, "bold"), fg="black", bg="#70aacf", bd=4, relief=GROOVE,
command=lambda: self.search()) # membuat tombol search untuk memulai menbcari data yang telah di ketikan
oleh user di entry search
366.     self.search_button.place(x=808, y=5, width="200")
367.
368.     self.show_button = Button(self.subf, cursor="hand2", text="Tampil Semua", font=(
369.         "times new roman", 15, "bold"), fg="black", bg="#70aacf", bd=4, relief=GROOVE,
command=lambda: self.showall()) # membuat tombol untuk menampilkan semua data yang ada di database
370.     self.show_button.place(x=1090, y=5, width="200")
371.
372.     self.scroll_x = ttk.Scrollbar(self.frame3, orient=VERTICAL)
373.     self.scroll_y = ttk.Scrollbar(self.frame3, orient=HORIZONTAL)
374.     self.treeview = ttk.Treeview(self.frame3, columns=(1, 2, 3, 4, 5, 6, 7, 8), show="headings",
375.         height=11, xscrollcommand=self.scroll_x.set,
yscrollcommand=self.scroll_y.set)
376.     self.scroll_x.pack(side=RIGHT, fill=Y)
377.     self.scroll_y.pack(side=BOTTOM, fill=X)
378.     '''
379.     scroll_x yang artinya variabel yang bernama scroll_x menggunakan widget scrollbar dari ttk dengan
parameter(namavariabel window, kemudian orientasi nya apa (vertikal/horizontal))
380.     kemudian sama dengan yang di bawah nya yaitu scrooll_y yang artinya variabel yang bernama scroll_y
menggunakan widget sama seperti scroll_x dan parameter yang di gunakan nya sama akan tetapi value yang di
gunakan berbeda
381.     kemudian dalam tabel ini di tentukan ada berapa kolom dan heading nya di tampilkan sehingga dapat
di lihat oleh user dan tak lupa juga di atur ketinggian dan
382.     scroll_x dan scroll_y nya dengan command seperti sintax di atas
383.     dan untuk scroll_x berada di kanan dan scroll_y berada di bawah table
384.     '''
385.
386.     # pada heading atau kepala table di kolom 1 di beri nama ID
387.     self.treeview.heading(1, text="ID")

```



```

388.         # pada heading atau kepala table di kolom 1 di beri nama nim
389.         self.treeview.heading(2, text="NIM")
390.         # pada heading atau kepala table di kolom 1 di beri nama "Nama Lengkap"
391.         self.treeview.heading(3, text="Nama Lengkap")
392.         # pada heading atau kepala table di kolom 1 di beri nama tempat & tanggal lahir
393.         self.treeview.heading(4, text="Tempat & Tanggal Lahir")
394.         # pada heading atau kepala table di kolom 1 di beri nama jenis kelamin
395.         self.treeview.heading(5, text="Jenis Kelamin")
396.         # pada heading atau kepala table di kolom 1 di beri nama jurusan
397.         self.treeview.heading(6, text="Jurusan")
398.         # pada heading atau kepala table di kolom 1 di beri nama alamat
399.         self.treeview.heading(7, text="Alamat")
400.         # pada heading atau kepala table di kolom 1 di beri nama nomor telepon
401.         self.treeview.heading(8, text="Nomor Telpon")
402.
403.         # treeview atau tabel pada colom 1 menggunakan ukuran lebar 100 dan posisi tulisan nya di
tengah/center
404.         self.treeview.column(1, width=100, anchor=CENTER)
405.         # treeview atau tabel pada colom 2 menggunakan ukuran lebar 100 dan posisi tulisan nya di
tengah/center
406.         self.treeview.column(2, width=100, anchor=CENTER)
407.         # treeview atau tabel pada colom 3 menggunakan ukuran lebar 120 dan posisi tulisan nya di
tengah/center
408.         self.treeview.column(3, width=120, anchor=CENTER)
409.         # treeview atau tabel pada colom 4 menggunakan ukuran lebar 120 dan posisi tulisan nya di
tengah/center
410.         self.treeview.column(4, width=120, anchor=CENTER)
411.         # treeview atau tabel pada colom 5 menggunakan ukuran lebar 160 dan posisi tulisan nya di
tengah/center
412.         self.treeview.column(5, width=160, anchor=CENTER)
413.         # treeview atau tabel pada colom 6 menggunakan ukuran lebar 120 dan posisi tulisan nya di
tengah/center
414.         self.treeview.column(6, width=120, anchor=CENTER)
415.         # treeview atau tabel pada colom 7 menggunakan ukuran lebar 140 dan posisi tulisan nya di
tengah/center
416.         self.treeview.column(7, width=140, anchor=CENTER)
417.         # treeview atau tabel pada colom 8 menggunakan ukuran lebar 100 dan posisi tulisan nya di
tengah/center
418.         self.treeview.column(8, width=100, anchor=CENTER)
419.
420.         # ini adalah treeview atau bila di lihat oleh user bentuk nya seperti table yang didalam nya
mengandung data data yang ada di database yang sudah di urutkan
421.         self.treeview.place(x=5, y=60, width=1330)
422.         self.tree()
423.         # kemudian jika ada salah satu baris di klik maka akan menjaalan kan function click_insert()
424.         self.treeview.bind("<ButtonRelease-1>", self.click_insert)
425.
426.         def update_student(self):
427.             if self.nim_entry.get() == "" or self.nama_entry.get() == "" or self.tanggal_entry.get() ==
"Select Class" or self.jenisKel_entry.get() == "Pilih Jenis Kelamin" or self.jurusan_entry.get() ==
"Pilih Jurusan" or self.telpon_entry.get() == "" or self.alamat_entry.get() == "":
428.                 messagebox.showerror("Error", "Semua Field Harus Di Isi")
429.             else:
430.                 con = pymysql.connect(
431.                     host='localhost', user='root', password='', database='aplikasidatamahasiswa')

```

```

432.         cur = con.cursor()
433.         cur.execute("update data set
nim=%s,nama=%s,tanggalTempat=%s,jenisKelamin=%s,jurusan=%s,alamat=%s,telepon=%s where id=%s",
(self.nim_entry.get(
434.             ), self.nama_entry.get(), self.tanggal_entry.get(), self.jenisKel_entry.get(),
self.jurusan_entry.get(), self.alamat_entry.get(), self.telpon_entry.get(), self.id_entry.get()))
435.         con.commit()
436.         con.close()
437.         messagebox.showinfo("Success", "Data Mahasiswa Sudah Di Update")
438.         self.tree()
439.         self.UpdateDataClear()
440.
441.     def delete_student(self):
442.         if self.nim_entry.get() == "":
443.             messagebox.showerror(
444.                 "Error", "Silahkan Masukkan NIM Untuk Menghapus Data Mahasiswa Atau Bisa Pilih Pada Kolom
Pencarian")
445.         else:
446.             con = pymysql.connect(
447.                 host='localhost', user='root', password='', database='aplikasidatamahasiswa')
448.             cur = con.cursor()
449.             cur.execute("delete from data where nim=%s", self.nim_entry.get())
450.             con.commit()
451.             con.close()
452.             messagebox.showinfo("Success", "Data Mahasiswa Sudah Terhapus")
453.             self.tree()
454.             self.UpdateDataClear()
455.
456.     def UpdateDataClear(self):
457.         # function ini akan di eksekusi jika ada function lain yang memanggil atau pun tombol clear di
click oleh user
458.         # menghapus entry field yang nama variabel nya nim_enrty
459.         self.nim_entry.delete(0, END)
460.         # menghapus entry field yang nama variabel nya nama_entry
461.         self.nama_entry.delete(0, END)
462.         # menghapus entry field yang nama variabel nya tanggal_enrty
463.         self.tanggal_entry.delete(0, END)
464.         # mengeset kembali combobox pada pilihan pertama yang nama variabel nya jenisKel_entry
465.         self.jenisKel_entry.set("Pilih Jenis Kelamin")
466.         # mengeset kembali combobox pada pilihan pertama yang nama variabel nya jurusan_entry
467.         self.jurusan_entry.set("Pilih Jurusan")
468.         # menghapus entry field yang nama variabel nya
469.         self.telpon_entry.delete(0, END)
470.         # menghapus entry field yang nama variabel nya
471.         self.alamat_entry.delete(0, END)
472.         # menghapus entry field yang nama variabel nya
473.         # menghapus entry field yang nama variabel nya
474.         self.id_entry.delete(0, END)
475.
476.     def tree(self):
477.         # funcion ini sama seperti 2 function yang sebelum nya function lain harus memanggil function ini
agar di eksekusi oleh program
478.         # membuat koneksi ke database yang bernama pythongui
479.         con = pymysql.connect(host='localhost', user='root',
480.                               password='', database='aplikasidatamahasiswa')

```

```
481. cur = con.cursor() # menyiapkan tempat untuk syntax sql
482. # syntax sql untuk mengambil semua data untuk di tampilkan seperti dalam table
483. cur.execute("select * from data")
484. # artinya semua row yang ada di dalam database akan di tampilkan di table
485. self.row = cur.fetchall()
486. if len(self.row) != 0: # jika dalam row ada dataya maka
487.     # program akan menjalankan function treeview.delete()
488.     self.treeview.delete(*self.treeview.get_children())
489.     for i in self.row: # for untuk menampilkan ssemua data yang ada di database seperti table
sama seperti yang ada di database
490.         # semua data di tiap kolom akan di masukkan ke dalam value i
491.         self.treeview.insert('', 'end', values=i)
492.         con.commit() # kemudian data data tersebut di commit agar dapat Lasngung di olah oleh
function lainnya seperti function insert dimana function tersebut memasukkan data
493.         # kedalam field sesuai dengan kolom yang ada di database
494.
495. def click_insert(self, ev):
496.     # function ini akan di eksekusi jika user mengclick salah satu baris yang ada di tampilan tabel
atau treeview yang menampilkan database jika user mengclick salah satu baris secara orto matis
497.     # function ini akan di eksekusi dan semua entry akan di isi dengan data yang sesuai dengan data
yang ada di database akan tetapi sesuai dengan bagian baris mana user mencllick
498.     # jika pada baris dengan id 1 maka data yang sebaris dengan id 1 akan di inputkan atau di
tampilkan ke dalam entry agar user dapat merubah data tersebut melalu entry field
499.     self.UpdateDataClear() # memanggil function UpdateDataClear()
500.     # data row di fokuskan pada data yang di click oleh user
501.     self.cur_row = self.treeview.focus()
502.     # isi atau item semua data di cur_row yang di click user akan di simpan ke dalam content
503.     self.content = self.treeview.item(self.cur_row)
504.     # dengan info content nya adalah values
505.     self.info = self.content['values']
506.
507.     # entry fields dengan nama variabel id_entry diisi dengan data yang ada di kolom id sesuai dengan
data yang di click oleh user dan self_info itu di artikan dari index seberapa data tersebut berasal
508.     self.id_entry.insert(0, self.info[0])
509.     # entry fields dengan nama variabel nim_entry diisi dengan data yang ada di kolom id sesuai
dengan data yang di click oleh user dan self_info itu di artikan dari index seberapa data tersebut
berasal
510.     self.nim_entry.insert(0, self.info[1])
511.     # entry fields dengan nama variabel nama_entry diisi dengan data yang ada di kolom id sesuai
dengan data yang di click oleh user dan self_info itu di artikan dari index seberapa data tersebut
berasal
512.     self.nama_entry.insert(0, self.info[2])
513.     # entry fields dengan nama variabel tanggal_entry diisi dengan data yang ada di kolom id sesuai
dengan data yang di click oleh user dan self_info itu di artikan dari index seberapa data tersebut
berasal
514.     self.tanggal_entry.insert(0, self.info[3])
515.     # entry fields dengan nama variabel jenisKel_entry diset sesuai dengan data yang ada di kolom id
sesuai dengan data yang di click oleh user dan self_info itu di artikan dari index seberapa data
tersebut berasal
516.     self.jenisKel_entry.set(self.info[4])
517.     # entry fields dengan nama variabel jurusan_entry diset sesuai dengan data yang ada di kolom id
sesuai dengan data yang di click oleh user dan self_info itu di artikan dari index seberapa data
tersebut berasal
518.     self.jurusan_entry.set(self.info[5])
```

```
519.         # entry fields dengan nama variabel alamat_entry diisi dengan data yang ada di kolom id sesuai
dengan data yang di click oleh user dan self_info itu di artikan dari index beberapa data tersebut
berasal
520.         self.alamat_entry.insert(0, self.info[6])
521.         # entry fields dengan nama variabel telpon_entry diisi dengan data yang ada di kolom id sesuai
dengan data yang di click oleh user dan self_info itu di artikan dari index beberapa data tersebut
berasal
522.         self.telpon_entry.insert(0, self.info[7])
523.
524.     def showall(self): # function ini di di eksekusi jika tombol show di click oleh user
525.         self.tree() # memanggil function tree untuk menampilkan seluruh data tanpa kesuali
526.
527.     def search(self):
528.         # cari data berdasarkan nim
529.         if self.scombo.get() == "NIM":
530.             con = pymysql.connect(host='localhost', user='root',
531.                                     password='', database='aplikasidatamahasiswa') # koneksi ke database
532.             cur = con.cursor()
533.             cur.execute("select * from data where nim=%s",
534.                         self.search_entry.get()) # syntax sql
535.             self.row = cur.fetchall() # menampilkan seluruh data berdasarkan nim
536.
537.             if len(self.row) != 0: # jika data nya ada maka
538.                 # tabel akan seperti di reset atau di kosongkan
539.                 self.treeview.delete(*self.treeview.get_children())
540.
541.                 for i in self.row: # kemudian di tampilkan sesuai dengan data yang ada di database
dengan cara for atau looping
542.                     # menginsert data ke dalam tabel agar user dapat melihat data yang dia cari
543.                     self.treeview.insert('', 'end', values=i)
544.                     con.commit() # kemudian koneksi di comit agar semua data dapat di keluarkan
545.             else: # jika tidak ada data maka
546.                 # muncul pesan untuk user
547.                 messagebox.showinfo(
548.                     "Information", "Tidak Ada Informasi dari NIM Yang anda Masukkan")
549.         # cari data berdasarkan nama
550.         elif self.scombo.get() == "Nama Lengkap":
551.             con = pymysql.connect(
552.                 host='localhost', user='root', password='', database='aplikasidatamahasiswa')
553.             cur = con.cursor()
554.             cur.execute("select * from data where nama=%s",
555.                         self.search_entry.get())
556.             self.row = cur.fetchall()
557.
558.             if len(self.row) != 0:
559.                 self.treeview.delete(*self.treeview.get_children())
560.
561.                 for i in self.row:
562.                     self.treeview.insert('', 'end', values=i)
563.                     con.commit()
564.             else:
565.                 messagebox.showinfo(
566.                     "Information", "Tidak Ada Informasi dari Nama Yang anda Masukkan")
567.
568.         # cari data berdasarkan jurusan
```

```
569. elif self.scombo.get() == "Jurusan":
570.     con = pymysql.connect(
571.         host='localhost', user='root', password='', database='aplikasidatamahasiswa')
572.     cur = con.cursor()
573.     cur.execute("select * from data where jurusan=%s",
574.                 self.search_entry.get())
575.
576.     self.row = cur.fetchall()
577.
578.     if len(self.row) != 0:
579.         self.treeview.delete(*self.treeview.get_children())
580.
581.         for i in self.row:
582.             self.treeview.insert('', 'end', values=i)
583.             con.commit()
584.     else:
585.         messagebox.showinfo(
586.             "Information", "Tidak Ada Informasi dari nama Jurusan Yang anda Masukkan")
587.
588. else:
589.     messagebox.showerror(
590.         "Error", "Silahkan Pilih Opsi Terlebih Dahulu Sebelum Mencari Data")
591.
592. def forget_pass(self):
593.     # function ini akan di eksekusi jika user mengclick tombol forget password pada form Login
594.     # membuat fram baru atau wondow baru untuk form Lupa password atau reset password
595.     Frame_ResetPassword = Frame(self.root, bg="#c5ded0")
596.     Frame_ResetPassword.place(x=0, y=0, height=700, width=1366)
597.
598.     self.img = ImageTk.PhotoImage(file="gambar2.JPG")
599.     img = Label(Frame_ResetPassword, image=self.img).place(
600.         x=0, y=0, height=700, width=1366)
601.
602.     # membuat frame untuk input atau tempat seperti entry, label, combobox dan lainnya di dalam frame
atau window inti dalam function ini
603.     frame_input_resetPass = Frame(self.root, bg='white')
604.     frame_input_resetPass.place(x=320, y=130, height=450, width=630)
605.
606.     # membuat Label yang bernama Form Reset Password
607.     label1 = Label(frame_input_resetPass, text=" Form Reset Password", font=(
608.         'impact', 32, 'bold'), fg="black", bg="white")
609.     label1.place(x=45, y=20)
610.
611.     # membuat Label yang bernama Username
612.     label2 = Label(frame_input_resetPass, text="User Name", font=(
613.         'Goudy old style', 20, 'bold'), fg="orangered", bg="white")
614.     label2.place(x=30, y=95)
615.
616.     # membuat entry untuk username
617.     self.entry = Entry(frame_input_resetPass, font=(
618.         'times new roman', 15, 'bold'), bg="lightgray")
619.     self.entry.place(x=30, y=145, width=270, height=35)
620.
621.     # membuat Label yang bernama Password
622.     label3 = Label(frame_input_resetPass, text="Password", font=(
```

```

623.         'Goudy old style', 20, 'bold'), fg="orangered", bg="white")
624. label3.place(x=30, y=195)
625.
626.     # membuat entry untuk password
627.     self.entry2 = Entry(frame_input_resetPass, show='*', font=(
628.         'times new roman', 15, 'bold'), bg="lightgray")
629.     self.entry2.place(x=30, y=245, width=270, height=35)
630.
631.     # membuat Label yang bernama Email
632.     label4 = Label(frame_input_resetPass, text="Email",
633.         font=('Goudy old style', 20, 'bold'), fg="orangered", bg="white")
634.     label4.place(x=330, y=95)
635.
636.     # membuat entry untuk Email
637.     self.entry3 = Entry(frame_input_resetPass, font=(
638.         'times new roman', 15, 'bold'), bg="lightgray")
639.     self.entry3.place(x=330, y=145, width=270, height=35)
640.
641.     # membuat Label yang bernama Konfirmasi Password
642.     label5 = Label(frame_input_resetPass, text="Konfirmasi Password", font=(
643.         'Goudy old style', 20, 'bold'), fg="orangered", bg="white")
644.     label5.place(x=330, y=195)
645.
646.     # membuat entry untuk Konfirmasi password
647.     self.entry4 = Entry(frame_input_resetPass, show='*', font=(
648.         'times new roman', 15, 'bold'), bg="lightgray")
649.     self.entry4.place(x=330, y=245, width=270, height=35)
650.
651.     # membuat tombol yang bernama Reset Password
652.     btn2 = Button(frame_input_resetPass, text="Reset Password", command=self.reset, cursor='hand2',
font=(
653.         'times new roman', 15), bg='orangered', fg='white', bd=0, width=15, height=1)
654.     btn2.place(x=90, y=340)
655.
656.     # membuat tombol yang bernama Cek Username
657.     btn2 = Button(frame_input_resetPass, text="Cek Username", command=self.cekUsername,
cursor='hand2', font=(
658.         'times new roman', 15), bg='orangered', fg='white', bd=0, width=15, height=1)
659.     btn2.place(x=330, y=340)
660.
661.     # membuat tombol yang bernama "Selesai Reset Password?Login..." agar user dapat kembali ke form
Login
662.     btn3 = Button(frame_input_resetPass, text="Selesai Reset Password?Login...",
command=self.loginform,
663.         cursor='hand2', font=('calibri', 10), bg='white', fg='black', bd=0)
664.     btn3.place(x=110, y=390)
665.
666.     def reset(self):
667.         # function ini akan di eksekusi jika user mengclick tombol reset Password
668.         # pengecekan apakah semua entry masih kosong atau belum jika masih ada yang kosong maka
669.         if self.entry.get() == "" or self.entry2.get() == "" or self.entry3.get() == "" or
self.entry4.get() == "":
670.             # muncul eror agar user mengisi semua entry
671.             messagebox.showerror(
672.                 "Error", "Semua Field Harus Di isi", parent=self.root)

```

```
673.         elif self.entry2.get() != self.entry4.get(): # jika semua entry sudah di isi kemudian akan
            dilakukan pengecekan apakah password baru yang dibuat user sama dengan konfirmasi password nya jika tidak
            sama maka
674.             # muncul eror agar user memasukkan password dan konfirmasi password dengan isian yang sama
675.             messagebox.showerror(
676.                 "Error", "Password Dan Konfirmasi Password Harus Sama", parent=self.root)
677.         else: # jika lolos dari pengecekan di atas maka
678.             try:
679.                 # membuat koneksi database
680.                 con = pymysql.connect(
681.                     host='localhost', user='root', password='', database='aplikasidatamahasiswa')
682.                 cur = con.cursor() # tempat untuk data yang akan di eksekusi oleh syntax sql
683.                 cur.execute("update user set emailid=%s,password=%s,konfirmasiPassword=%s where
            username=%s", (self.entry.get(), self.entry2.get(), self.entry3.get(
684.                     ), self.entry4.get())) # syntax sql untuk memasukkan data yang baru ke dalam data yang
            ada di database agar semua data yang ada di database di perbarui
685.                 # dalam sistem ini adalah penipaan data yang lama dengan data yang baru agar data yang
            ada di database di perbaharui dan dapat di gunakan oleh user
686.                 con.commit() # syntax sql di comit agar data nya di masukkan ke dalam database
687.                 con.close() # koneksi ke database di tutup
688.                 # muncul pesan jika reset pasword sukses
689.                 messagebox.showinfo("Success", "Reset Password Sukses")
690.                 self.resetclear() # memanggil function resetclear()
691.             except Exception as es: # jika ada kesalahan yang terjadi maka akan di tampilkan sebagai
            pesan eror
692.                 messagebox.showerror(
693.                     "Error", f"Error due to:{str(es)}", parent=self.root)
694.
695.         def resetclear(self):
696.             # funtion ini akan di eksekusi jika di panggil oleh function lain
697.             # menghapus semua isi atau ketikan di entry oleh user
698.             self.entry.delete(0, END)
699.             # menghapus semua isi atau ketikan di entry2 oleh user
700.             self.entry2.delete(0, END)
701.             # menghapus semua isi atau ketikan di entry3 oleh user
702.             self.entry3.delete(0, END)
703.             # menghapus semua isi atau ketikan di entry4 oleh user
704.             self.entry4.delete(0, END)
705.
706.         def cekUsername(self):
707.             # function ini akan di eksekusi jika tombol cek username di click oleh user yang berguna agar
            user dapat mengecek apakah username milik nya masih ada di dalam database atau tidak
708.             # pengecekan pada entry untuk username apakah masih kosong atau sudah di isi oleh user, jika
            masih kosong maka
709.             if self.entry.get() == "":
710.                 # muncul eror agar user mengisi entry untuk username jika user lupa atau pun sengaja
711.                 messagebox.showerror(
712.                     "Error", "Username Field Masih Kosong", parent=self.root)
713.             else:
714.                 try:
715.                     # membuat koneksi ke database pythongui
716.                     con = pymysql.connect(
717.                         host="localhost", user="root", password="", database="aplikasidatamahasiswa")
718.                     cur = con.cursor() # membuat tempat untuk data yang akan di eksekusi dengan syntax sql
```



```

719.         # syntax sql untuk mengambil semua data di dalam table register berdasarkan kolom
username untuk menyamakan apakah ada data yang sama dengan data yang ada di entry username
720.         cur.execute(
721.             "select * from user where username=%s", self.entry.get())
722.         row = cur.fetchone() # jika ada maka data tersebut di ambil 1 baris
723.         self.data = row # kemudian data tersebut di simpan ke dalam variabel data
724.         if self.data == None: # jika data nya tidak ada yang sama dengan data yang ada di entry
username maka
725.             # muncul eror agar user dapat mengetahui username yang dia cari tidak ada di dalam
database
726.             messagebox.showerror(
727.                 "Error", "Username Tidak Di Temukan, Silahkan Mencoba Untuk Menggunakan Username
Lain", parent=self.root)
728.             self.entry.focus() # arah ketikan atau crusor di fokuskan pada entry username agar
user dapat menghapus ketikan sebelum nya dan dapat mengetik username yang lainnya untuk di cari
729.             con.close() # koneksi ke database di tutup
730.             else: # jika data yang Di cari oleh user ada di database dan sama maka
731.                 # muncul informasi agar user membuat password baru untuk menggantikan password lama
yang ada di database
732.                 messagebox.showinfo(
733.                     "Information", "Username Ditemukan, Silahkan Membuat Password Baru",
parent=self.root)
734.                 self.resetclear() # memanggil reset clear() agar di pastikan semua entry kosong dan
di isikan dengan data yang di temukan sesuai dengan username
735.                 # netry pertama di inputkan data dari index ke 1 karna index 0 itu adalah id
736.                 self.entry.insert(END, self.data[1])
737.                 # entry kedua di inputkan data dari index ke 3
738.                 self.entry2.insert(END, self.data[3])
739.                 # entry ketiga di inputkan data dari index ke 2
740.                 self.entry3.insert(END, self.data[2])
741.                 # entry ke empat di inputkan data dari index ke 4
742.                 self.entry4.insert(END, self.data[4])
743.                 con.close() # koneksi ke database di tutup
744.             except Exception as es: # jika ada kesalahan maka akan di munculkan sebagai pesan eror
745.                 messagebox.showerror(
746.                     "Error", f"Error due to:{str(es)}", parent=self.root)
747.
748.         # form register
749.
750.         def register(self):
751.             # function ini akan di eksekusi jika user mengclick regiter pada form Login
752.             # membuat frame baru atau window baru untuk form registrasi
753.             Frame_Register = Frame(self.root, bg="#c5ded0")
754.             Frame_Register.place(x=0, y=0, height=700, width=1366)
755.
756.             self.img = ImageTk.PhotoImage(file="gambar2.JPG")
757.             img = Label(Frame_Register, image=self.img).place(
758.                 x=0, y=0, height=700, width=1366)
759.
760.             # membuat frame input di dalam window inti atau fram inti pada function ini untuk tempat Label,
entry, tombol dan lainnya
761.             frame_input_register = Frame(self.root, bg='white')
762.             frame_input_register.place(x=320, y=130, height=450, width=630)
763.
764.             # membuat Label yang bernama Form Registrasi

```

```
765. label1 = Label(frame_input_register, text="Form Registrasi", font=(
766.     'impact', 32, 'bold'), fg="black", bg="white")
767. label1.place(x=45, y=20)
768.
769. # membuat Label yang bernama username
770. label2 = Label(frame_input_register, text="User Name", font=(
771.     'Goudy old style', 20, 'bold'), fg="orangered", bg="white")
772. label2.place(x=30, y=95)
773.
774. # membuat entry yang untuk username
775. self.entry = Entry(frame_input_register, font=(
776.     'times new roman', 15, 'bold'), bg="lightgray")
777. self.entry.place(x=30, y=145, width=270, height=35)
778.
779. # membuat Label yang bernama Password
780. label3 = Label(frame_input_register, text="Password", font=(
781.     'Goudy old style', 20, 'bold'), fg="orangered", bg="white")
782. label3.place(x=30, y=195)
783.
784. # membuat entry yang untuk password
785. self.entry2 = Entry(frame_input_register, show='*', font=(
786.     'times new roman', 15, 'bold'), bg="lightgray")
787. self.entry2.place(x=30, y=245, width=270, height=35)
788.
789. # membuat Label yang bernama Email
790. label4 = Label(frame_input_register, text="Email",
791.     font=('Goudy old style', 20, 'bold'), fg="orangered", bg="white")
792. label4.place(x=330, y=95)
793.
794. # membuat entry yang untuk email
795. self.entry3 = Entry(frame_input_register, font=(
796.     'times new roman', 15, 'bold'), bg="lightgray")
797. self.entry3.place(x=330, y=145, width=270, height=35)
798.
799. # membuat Label yang bernama Konfirmasi Password
800. label5 = Label(frame_input_register, text="Konfirmasi Password", font=(
801.     'Goudy old style', 20, 'bold'), fg="orangered", bg="white")
802. label5.place(x=330, y=195)
803.
804. # membuat entry yang untuk konfirmasi password
805. self.entry4 = Entry(frame_input_register, show='*', font=(
806.     'times new roman', 15, 'bold'), bg="lightgray")
807. self.entry4.place(x=330, y=245, width=270, height=35)
808.
809. # membuat tombol yang bernama Register
810. btn2 = Button(frame_input_register, text="REGISTER", command=self.regis, cursor='hand2', font=(
811.     'times new roman', 15), bg='orangered', fg='white', bd=0, width=15, height=1)
812. btn2.place(x=90, y=340)
813.
814. # membuat tombol yang bernama "Sudah Registrasi? Kembali Ke Login...." untuk kembali ke form
Login
815. btn3 = Button(frame_input_register, text="Sudah Registrasi? Kembali Ke login....",
command=self.loginform,
816.     cursor='hand2', font=('calibri', 10), bg='white', fg='black', bd=0)
817. btn3.place(x=110, y=390)
```

```

818.
819.     # fungsi untuk memasukkan data ke database
820.
821.     def regis(self): # function akan di eksekusi jika tombol registrasi di click oleh user
822.         # pengecekan apakah semua entry masih kosong atau belum jika masih ada yang kosong maka
823.         if self.entry.get() == "" or self.entry2.get() == "" or self.entry3.get() == "" or
self.entry4.get() == "":
824.             # akan muncul eror
825.             messagebox.showerror(
826.                 "Error", "Semua Field Harus Di isi", parent=self.root)
827.         elif self.entry2.get() != self.entry4.get(): # jika semua entry telah di isi oleh user maka
pengecekan untuk password dan konformasi password sudah sama atau belum jika belum sama maka
828.             # akan muncul eror agar user memasukan password dan konfirmasi pasword dengan isian sama
829.             messagebox.showerror(
830.                 "Error", "Password And Konfirmasi Password Harus Sama", parent=self.root)
831.         else: # jika sudah terisi semua dan password dan konfirmasi password sudah sama maka
832.             try:
833.                 # membuat koneksi ke database bernama pythongui
834.                 con = pymysql.connect(
835.                     host="localhost", user="root", password="", database="aplikasidatamahasiswa")
836.                 cur = con.cursor() # membuat tempat
837.                 # syntax sql untuk mengecek username sudah ada atau belum di database dengan di samakan
data yang di isi di entry
838.                 cur.execute(
839.                     "select * from user where username=%s", self.entry.get())
840.                 row = cur.fetchone() # memunculkan semua data di dalam 1 row username
841.                 if row != None: # jika username yang di tulis oleh user maka
842.                     messagebox.showerror(
843.                         "Error", "Username Sudah Ada,Silahkan Buat Username Baru", parent=self.root) #
muncul pesan eror agar user mengisi username yang baru
844.                     self.entry.focus() # di fokuskan pada entry username agar user dapat menghapus isi
yang telah di ketik sebelum nya agar user dapat membuat username baru
845.                     else: # jika tidak ada username yang sama dengan yang ada di database atau username baru
maka
846.                         cur.execute("Insert into
user(username,emailid,password,konfirmasiPassword)values('%s','%s','%s','%s') " % (
847.                             self.entry.get(), self.entry3.get(), self.entry2.get(), self.entry4.get())) #
syntax sql untuk memasukkann semua data yang ada di entry ke dalam table register sesuai dengan kolom yang
ada
848.                             con.commit() # syntax sql di comit agar data di simpan ke dalam database
849.                             con.close() # koneksi ke database di tutup
850.                             # kemudian akan muncul pesan sukses yang artinya satu akun sudah di buat dan user
dapat masuk ke dalam database sesuai dengan akun yang dia buat
851.                             messagebox.showinfo(
852.                                 "Success", "Registrasi Berhasil", parent=self.root)
853.                             self.regclear() # memanggil function regclear()
854.                             except Exception as es: # jika ada eror maka akan di munculkan apa eror nya
855.                                 messagebox.showerror(
856.                                     "Error", f"Error due to:{str(es)}", parent=self.root)
857.
858.         def regclear(self): # function ini di eksekusi jika function lain memanggil
859.             # menghapus isian atau data yang ada di entry dengan nama variabel entry
860.             self.entry.delete(0, END)
861.             # menghapus isian atau data yang ada di entry dengan nama variabel entry2
862.             self.entry2.delete(0, END)

```

```
863.         # menghapus isian atau data yang ada di entry dengan nama variabel entry3
864.         self.entry3.delete(0, END)
865.         # menghapus isian atau data yang ada di entry dengan nama variabel entry4
866.         self.entry4.delete(0, END)
867.
868.
869. root = Tk()
870. Aplikasi = Login(root)
871. # root pada kode syntax ini mengacu pada window utama (toplevel) yang bertugas untuk mencegah jendela
    aplikasi langsung ditutup.
872. root.mainloop()
```