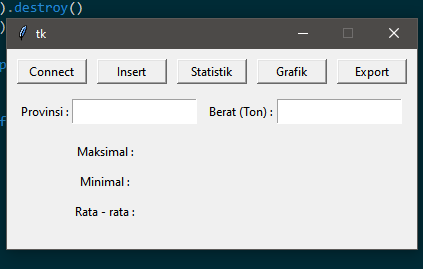
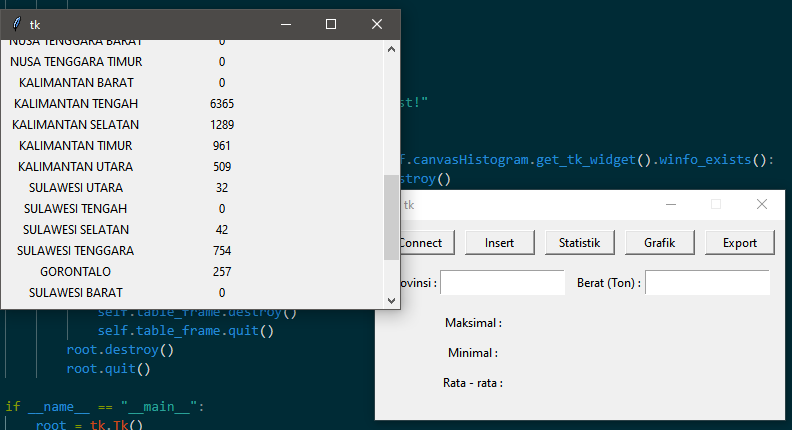
Penjelasan Singkat : Program ini menggunakan MySQL database menggunakan XAMPP dengan data yang sudah di berikan juga didalam zip filenya untuk data yang sudah di export berbentuk sql nanti bisa dimasukan ke dalam XAMPP dengan mengimportnya dan jalankan program

|  |  |  |
| --- | --- | --- |
| No | Line Code | Komentar |
| 1 | import tkinter as tk  import mysql.connector  import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  from tkinter import \*  from matplotlib.backends.backend\_tkagg import FigureCanvasTkAgg | Memerintahkan python untuk mengimpor seluruh modul pustaka ke dalam program dengan cara menyebutkan nama modul dan untuk from sendiri digunakan untuk mengimpor spesifik nama atau simbol tertentu dari modul |
| 2 | **class** Jengkol:  **class** App: | Memerintahkan Python untuk membuat class |
| 3 | **def** \_\_init\_\_(self, provinsi, jengkol):  **def** \_\_init\_\_(self, root): | Memerintahkan python untuk membuatkan konstruktor dari class yang pertama untuk class Jengkol dan kedua class App |
| 4 | self.provinsi = provinsi  self.jengkol = jengkol | Meminta python menginisialisasi parameter self yang mereference ke object yang sedang dibuat |
| 5 | self.table\_frame = None  self.pie\_frame = None  self.cnx = None  self.cursor = None  self.canvasHistogram = None | Meminta python membuatkan attriut attribut yang ada dan menginisialisasikan dengan nilai None |
| 6 | width=410  height=200 | Yang pertama untuk width dan heigh menginisialisasikannya dengan 410 dan 200 |
| 7 | screenwidth = root.winfo\_screenwidth()  screenheight = root.winfo\_screenheight() | Meminta python menginisialisasikan 2 variable ini dengan root.windo\_screenwidth() dan yang satunya screenheight() untuk mendapatkan lebar layar dan tinggi layar monitor gunakan |
| 8 | alignstr = '%dx%d+%d+%d' % (width, height, (screenwidth - width) / 2, (screenheight - height) / 2) | Meminta python menginisialisasi hasil yang berisi format posisi dan ukuran dengan nilai variabel sebelumnya |
| 9 | root.geometry(alignstr) | Memanggil method geometry() untuk mengatur geometri atau posisi dan ukuran jendela |
| 10 | root.resizable(width=False, height=False) | Memeinta python untuk membuat jendela agar tidak dapat diubah ukuranya oleh pengguna |
| 11 | LabelProvinsi = tk.Label(root)  LabelBerat = tk.Label(root)  elf.provinsi\_entry = tk.Entry(root)  self.jengkol\_entry = tk.Entry(root)  self.LabelMaks = tk.Label(root)  self.LabelMins = tk.Label(root)  self.LabelAvg = tk.Label(root)  self.LabelStatus = tk.Label(root)  ConnectButton=tk.Button(root)  InsertButton=tk.Button(root)  StatistikButton=tk.Button(root)  GrafikButton=tk.Button(root)  ExportButton=tk.Button(root) | Membuat objek objek datri tk dan dimasukan ke variabel ini atau pada kasus ini ada beberapa yang menggunakan slef dimana disitu akan dijadikan attribut yang akan diakses di semua functionya |
| 12 | LabelProvinsi["text"] = "Provinsi : "  LabelBerat["text"] = "Berat (Ton) : "  self.LabelMaks["text"] = "Maksimal : "  self.LabelMins["text"] = "Minimal : "  self.LabelAvg["text"] = "Rata - rata : "  self.LabelStatus["text"] = ""  ConnectButton["text"] = "Connect"  InsertButton["text"] = "Insert"  StatistikButton["text"] = "Statistik"  GrafikButton["text"] = "Grafik"  ExportButton["text"] = "Export" | Mengatur teks atau label yang akan ditampilkan bedanya hanya pada self disini yang digunakan sebagai attribut bukan variabel |
| 13 | LabelProvinsi.place(x=5,y=50,width=70,height=25)  self.provinsi\_entry.place(x=65, y=50, width=125, height=25)  LabelBerat.place(x=200,y=50,width=70,height=25)  self.jengkol\_entry.place(x=270, y=50, width=125, height=25)  self.LabelMaks.place(x=50, y=90, width=100, height=25)  self.LabelMins.place(x=50, y=120, width=100, height=25)  self.LabelAvg.place(x=50, y=150, width=100, height=25)  self.LabelStatus.place(x=5, y=180, width=100, height=25)  ConnectButton.place(x=10,y=10,width=70,height=25)  InsertButton.place(x=90,y=10,width=70,height=25)  StatistikButton.place(x=170,y=10,width=70,height=25)  GrafikButton.place(x=250,y=10,width=70,height=25)  ExportButton.place(x=330,y=10,width=70,height=25) | Menempatkan posisi objek pada posisi tertentu di jendela dan perbedaanya hanya pada untuk self disini menetapkan menjadi sebagai attribut |
| 14 | self.LabelStatus["fg"] = "red" | Meminta tkinter python untuk mewarnai merah label ini |
| 15 | ConnectButton["command"] = self.ConnectButton\_command  InsertButton["command"] = self.InsertButton\_command  StatistikButton["command"] = self.StatistikButton\_command  GrafikButton["command"] = self.GrafikButton\_command  ExportButton["command"] = self.ExportButton\_command | Memasangkan button dengan function command dari function yang dimasukan |
| 16 | **def** CreateTable(self, data):  **def** getData(self):  **def** ConnectButton\_command(self):  **def** InsertButton\_command(self):  **def** StatistikButton\_command(self):  **def** GrafikButton\_command(self):  **def** ExportButton\_command(self):  **def** on\_window\_close(self): | Meminta Python untuk membuat suatu function yang berisikan suatu parameter yang berada didalam class App dikasus ini |
| 17 | total\_rows = len(data)  total\_columns = len(data[0]) | Meminta python untuk mengambil panjang array dan memasukanya ke dalam variabel |
| 18 | if self.table\_frame is not None and self.table\_frame.winfo\_exists():              self.table\_frame.destroy() | Memeriksa apakah table\_frame sudah ada dan masih ada dalam GUI dan jika iya maka akan menhancurkan table\_frame |
| 19 | self.table\_frame = tk.Toplevel(root) | Membuat jendela baru (Toplevel) |
| 20 | canvas = Canvas(self.table\_frame) | Membuat object canvas dan memasukannya didalam canvas dalam table\_frame untuk menampung konten |
| 21 | canvas.pack(side=LEFT, fill=BOTH, expand=True) | Mengatur tata letak dan mengisian pada objek canvas |
| 22 | scrollbar = Scrollbar(self.table\_frame, command=canvas.yview) | Membuatobjek scrollbar dengan nama scrollbar dalam table\_frame |
| 23 | scrollbar.pack(side=RIGHT, fill=Y) | Mengatur tata letak dan pengisian pada objek scroll bar |
| 24 | canvas.configure(yscrollcommand=scrollbar.set) | Mengkonfigurasi canvas agar menggunakan scrollbar secara vertikal |
| 25 | canvas.bind('<Configure>', **lambda** e: canvas.configure(scrollregion=canvas.bbox("all"))) | Mengikat atau bidning event <configure> pada canvas |
| 26 | frame = Frame(canvas) | Membuat objek frame dengan nama frame di dalam canvas |
| 27 | canvas.create\_window((0, 0), window=frame) | Window baru dalam canvas dan menghubungkanya dengan frame |
| 28 | for i in range(total\_rows):              for j in range(total\_columns):                  value = data[i][j]                  label = Label(frame, width=20, text=value)                  label.grid(row=I,column=j) | Perulangan for yang digunakan untuk menampilkan data yang disimpan dan ditampilan ke dalam frame dengan menggunakan Label |
| 29 | self.cursor.execute("SELECT \* FROM jengkol")  self.cursor.execute("INSERT INTO `jengkol`(`provinsi`, `ton`) VALUES (%s, %s)", (provinsi\_value, jengkol\_value)) | Mengeksekusi Query ini dengan menggunakan objeck cursor yang tersambung dengan koneksi database |
| 30 | rows = self.cursor.fetchall() | Mengambil semua data hasil eksekusi dan disimpan dalam variabel |
| 31 | jengkol\_objects = [] | Membuat list kosong |
| 32 | for row in rows:              provinsi = row[0]              jengkol = row[1]              jengkol\_obj = Jengkol(provinsi, jengkol)              jengkol\_objects.append(jengkol\_obj) | Melakukan perulangan dan memasukan data yang didapatkan sebelumnya yang berada didalam rows dan memasukanya ke dalam variabel untuk dibuat ke dalam objek yang nantinya dibuat ke dalam array of objek dari jengkol\_objects |
| 33 | return jengkol\_objects | Mengembalikan nilai jengkol\_objects kemana dia dipanggil |
| 34 | try:  except mysql.connector.Error as error: | Mencoba menjalankjan code dalam blok try dan akan ditangkap oleh except jika ada error dan akan dieksekusi oleh block except ini |
| 35 | self.cnx = mysql.connector.connect(                  host="localhost",                  user="root",                  password="",                  database="jengkol"              ) | Membuat koneksi ke database MySQL dengan mengatur parameter nya dan hasilnya disimpan dalam cnx dalam object self |
| 36 | self.cursor = self.cnx.cursor() | Membuat objek cursor |
| 37 | data = self.getData() | Memanggil function getData dan disimpan didalam variabel |
| 38 | data\_tuples = [(obj.provinsi, obj.jengkol) for obj in data] | Membuat list yang berisi nilai provinsi dan jengkol yang adad didalam variabel data |
| 39 | self.CreateTable(data\_tuples) | Memanggil function createTable dan mengisi parameternya |
| 40 | provinsi\_value = self.provinsi\_entry.get()  jengkol\_value = self.jengkol\_entry.get() | Mengambil nilai yang ada didalam entry field dan disimpan ke dalam variabel |
| 41 | if self.cursor: | Mengecek apakah cursor telah terinisialisasi |
| 42 | if provinsi\_value == "" or jengkol\_value == "":      self.LabelStatus["text"] = "No Data!" | Memeriksa apakah nilai provinsi dan jengkol kosong jika ya maka menampilkan “no data” jika tidak mengeksekusi blok else |
| 43 | self.cnx.commit() | Commit yang digunakan untuk menyimpan perubahan datanya |
| 44 | if len(jengkol\_values) > 0:  max\_value = np.max(jengkol\_values)  min\_value = np.min(jengkol\_values)  avg\_value = np.mean(jengkol\_values) | Memeriksa apakah value tidak kosong, jika iya maka melakukan pengolahan data dengan numpy dan memasukanya ke dalam variabel |
| 45 | self.fig, ax = plt.subplots() | Membuat objek menggunakan sublots dengan matplotlib |
| 46 | labels = [entry[0] for entry in data\_tuples]  jengkol\_values = [entry[1] for entry in data\_tuples] | Membuat list yang berisi nilai data dari setiap tuple dalam variabel data\_tuples |
| 47 | x\_pos = np.arange(len(labels)) | Membuat array dengan panjang sesuai dengan jumlah labels |
| 48 | ax.bar(x\_pos, jengkol\_values, align='center') | Membuat grafik batang menggunakan ax.bar() dengan posisi x\_pos dan nilai y jengkol\_values |
| 49 | ax.set\_xticks(x\_pos) | Mengatur posisi sumbu x |
| 50 | ax.set\_xticklabels(labels, rotation=90) | Mengatur label pada sumbu x dan memutar 90 derajat |
| 51 | ax.set\_xlabel('Provinsi')  ax.set\_ylabel('Berat (Ton)')  ax.set\_title('Berat Jengkol per Provinsi') | Mangatur label sumbux, sumbu y dan mengatur judulnya |
| 52 | if self.pie\_frame is not None and self.pie\_frame.winfo\_exists():                  self.pie\_frame.destroy() | Memeriksa apakah objek pie\_frame terinisialisasi jika masih ada maka akan destory |
| 53 | self.pie\_frame = tk.Toplevel(root) | Membuat frame baru menggunakan top level tkinter |
| 54 | self.canvasHistogram = FigureCanvasTkAgg(self.fig, master=self.pie\_frame) | Membuat objek sebagai wadah untuk objek gambar dari self.fig pada frame pie\_frame |
| 55 | self.canvasHistogram.get\_tk\_widget().config(width=900, height=700) | Mengatur dimensi dari canvas |
| 56 | self.canvasHistogram.draw() | Menggambar objek gambar |
| 57 | self.canvasHistogram.get\_tk\_widget().pack() | Memasang canvas pada frame |
| 58 | self.fig.tight\_layout() | Mengatur layout agar lebih ketat |
| 59 | column1 = [entry[0] for entry in data\_tuples]  column2 = [entry[1] for entry in data\_tuples] | Membuat list yang berisi nilai dari setiap tuple data data\_tuples |
| 60 | data = {'Provinsi': column1, 'Jengkol (Ton)': column2} | Membuat disctionary data dengan nama kolom sebagai kunci dan list sebagai nilai |
| 61 | df = pd.DataFrame(data) | Mengkonversi dictionary data menjadi dataframe |
| 62 | file\_path = 'data.xlsx' | Menentukan tempat file yang akan dieksport |
| 63 | df.to\_excel(file\_path, index=False) | Mengubah datafram ke excel dengan menggunakan to\_excel |
| 64 | if self.canvasHistogram is not None and self.canvasHistogram.get\_tk\_widget().winfo\_exists():  if self.pie\_frame is not None and self.pie\_frame.winfo\_exists():  if self.table\_frame is not None and self.table\_frame.winfo\_exists(): | Mengecek apakah frame atau canvas atau bisa dikatakn objek ini ada dan tidak kosong isinya |
| 65 | self.canvasHistogram.get\_tk\_widget().destroy()  self.canvasHistogram.get\_tk\_widget().quit()  self.pie\_frame.destroy()  self.pie\_frame.quit()  self.table\_frame.destroy()  self.table\_frame.quit()  root.destroy()  root.quit() | Menghentikan untuk quit dan menghancurkan widgetnya tkinter sehingga widget tidak ditampilkan dan untuk quit sendiri dapat berhenti berinteraksi dengan pengguna |
| 66 | plt.close(self.fig) | Menutup gambar dengan objek dari fig |
| 67 | if \_\_name\_\_ == "\_\_main\_\_": | Meminta untuk memeriksa apakah dieksekusi secara lansung sebagai program utama dan bukan diimport sebagai modul |
| 68 | root = tk.Tk() | Membuat instance dari kelas TK() |
| 69 | app = App(root) | Membuat instance dari kelas App dengan root sebagai argumen |
| 70 | root.protocol("WM\_DELETE\_WINDOW", app.on\_window\_close) | Mengatur aturan untuk menangai ketika jendela ditutup |
| 71 | root.mainloop() | Memulai loop tkinter yang berjalan |

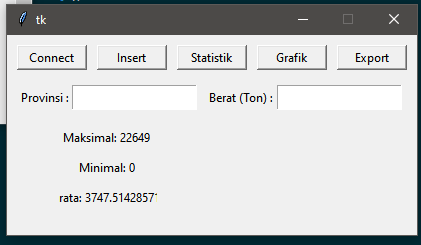
Hasil Running Program



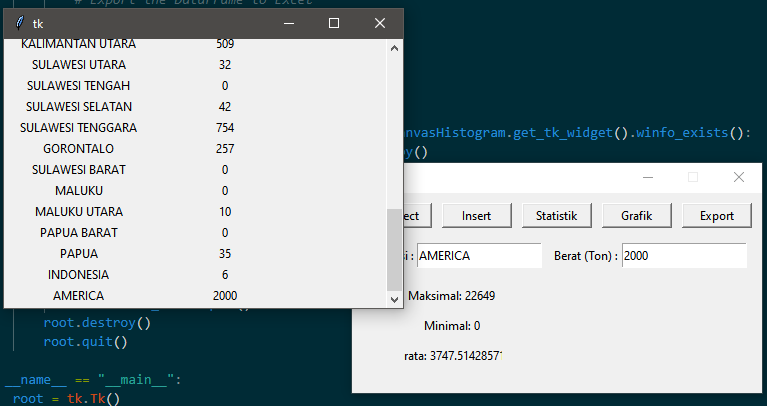
Tombol Connect



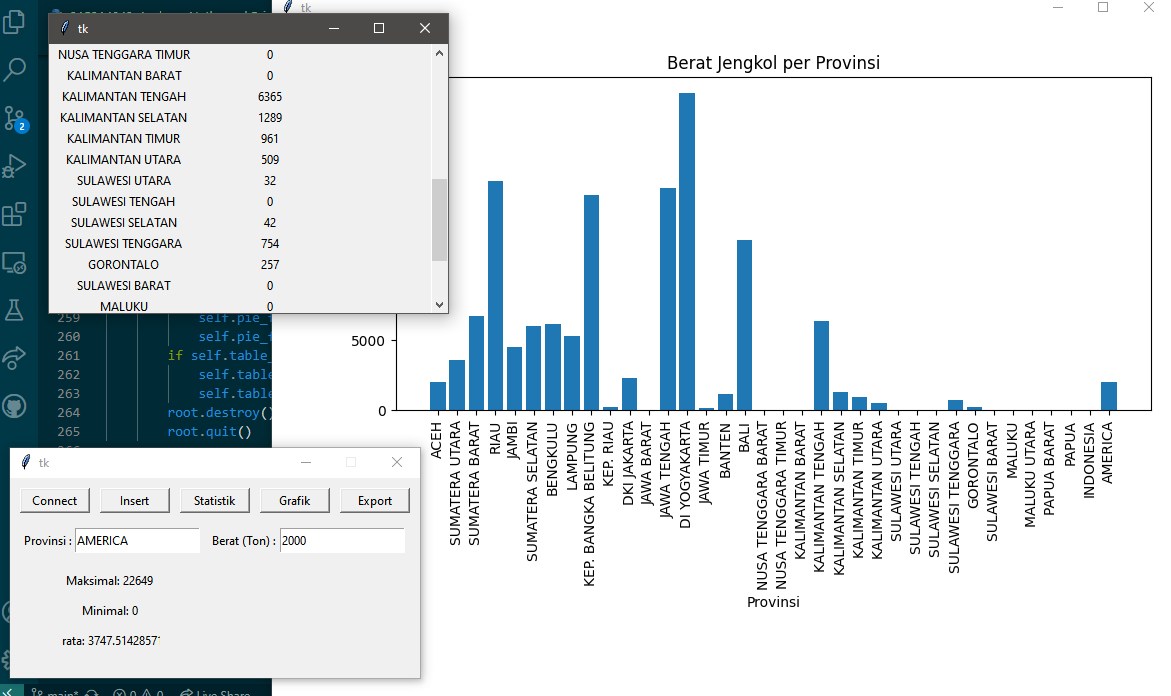
Tombol statistik



Tombol Insert



Tombol Grafik



Tombol eksport sendiri mengeksport datanya menjadi excel