**LAPORAN**

**PROJEK AKHIR**

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

****

**Disusun Oleh:**

|  |  |  |
| --- | --- | --- |
|  | **1. Siswadi Perdana Putra**  **2. Farhan Mirza Banazi** | **L200204049**  **L200204211** |

**TEKNIK INFORMATIKA**

**FAKULTAS KOMUNIKASI DAN INFORMATIKA**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2021/2022**

**Laporan Project Akhir Praktikum Algoritma dan Struktur Data**

|  |  |  |
| --- | --- | --- |
| No | Topics | Problem |
| 4. | Merge Sort | Creatae sorting program to sort:   1. Venue column in an alphabetical order 2. Time of the match, from the beginning to the end, and vice versa |

1. **Data**

**World Cup Timetable 2018**

|  |
| --- |
| Jadwal Pertandingan Piala Dunia 2018  1,Russia - Saudi Arabia,Moskwa Luzhniki,14 June 2018, 22.00  2,Egypt - Uruguay,Yekaterinburg,15 June 2018, 19.00  3,Russia - Egypt,St. Petersburg,20 June 2018, 01.00  4,Uruguay - Saudi Arabia,Rostov-on-Don,20 June 2018, 22.00  5,Arab Saudi - Egypt,Vologorad,25 June 2018, 21.00  6,Uruguay - Russia,Samara,25 June 2018, 21.00  7,Portugal - Spain,Sochi,16 June 2018, 01.00  8,Morocco - Iran,St. Petersburg,15 June 2018, 22.00  9,Portugal - Morocco,Moskwa Luzhniki,20 June 2018, 19.00  10,Iran - Spain,Kazan,21 June 2018, 01.00  11,Spain - Morocco,Kaliningrad,26 June 2018, 01.00  12,Iran - Portugal,Saransk,26 June 2018, 01.00  13,France - Australia,Kazan,16 June 2018, 17.00  14,Peru - Denmark,Saransk,16 June 2018, 23.00  15,France - Peru,Yekaterinburg,21 June 2018, 22.00  16,Denmark - Australia,Samara,21 June 2018, 19.00  17,Denmark - France,Moskwa Luzhniki,26 June 2018, 21.00  18,Australia - Peru,Sochi,26 June 2018, 21.00  19,Argentina - Iceland,Moskwa Spartak,16 June 2018, 20.00  20,Croatia - Nigeria,Kaliningrad,17 June 2018, 02.00  21,Argentina - Croatia,Nizhni Novgorod,22 June 2018, 01.00  22,Nigeria - Iceland,Volgograd,22 June 2018, 22.00  23,Iceland - Croatia,Rostov-on-Don,27 June 2018, 01.00  24,Nigeria - Argentina,St. Petersburg,27 June 2018, 01.00  25,Brazil - Switzerland,Rostov-on-Don,18 June 2018, 01.00  26,Costa Rica - Serbia,Samara,17 June 2018, 19.00  27,Brazil - Costa Rica,St. Petersburg,22 June 2018, 19.00  28,Serbia - Switzerland,Kaliningrad,23 June 2018, 01.00  29,Serbia - Brazil,Moskwa Spartak,28 June 2018, 01.00  30,Switzerland - Costa Rica,Nizhni Novgorod,28 June 2018, 01.00  31,Germany - Mexico,Moskwa Luzhniki,17 June 2018, 22.00  32,Sweden - South Korea,Nizhni Novgorod,18 June 2018, 19.00  33,Germany - Sweden,Sochi,24 June 2018, 01.00  34,South Korea - Mexico,Rostov-on-Don,23 June 2018, 22.00  35,South Korea - Germany,Kazan,27 June 2018, 21.00  36,Mexico - Sweden,Yekaterinburg,27 June 2018, 21.00  37,Belgium - Panama,Sochi,18 June 2018, 22.00  38,Tunisia - England,Volovograd,19 June 2018, 01.00  39,Belgium - Tunisia,Moskwa Spartak,23 June 2018, 19.00  40,England - Panama,Nizhni Novgorod,24 June 2018, 19.00  41,England - Belgium,Kaliningrad,29 June 2018, 01.00  42,Poland - Senegal,Moskwa Spartak,19 June 2018, 22.00  43,Colombia - Japan,Saransk,19 June 2018, 19.00  44,Japan - Senegal,Yekaterinburg,24 June 2018, 22.00  45,Poland - Colombia,Kazan,25 June 2018, 01.00  46,Japan - Poland,Volgograd,28 June 2018, 21.00  47,Senegal - Colombia,Samara,28 June 2018, 21.00 |

Pada data pertandingan piala dunia tahun 2018 kami memasukkan data di atas pada excel dengan format .csv yang nantinya dimana data tersebut akan dipanggil pada kodingan yang kami buat.

1. **Kode yang di buat**
2. **Membuat class JadwalPertandinganBola()**

from prettytable import PrettyTable

# from pyrsistent import l

class JadwalPertandinganBola(object):

    def \_\_init\_\_(self, data):

        f = open(data, "r")

        self.jadwal = []

        for x in f.read().split('\n'):

            pecah = x.split(",")

            if len(pecah) > 1:

                self.jadwal.append(

                    {"No": pecah[0], "Match": pecah[1], "Venue": pecah[2], "Time": pecah[3] + pecah[4]})

# urutkan data berdasarkan targetUrut dan urut

kami membuat sebuat class yang menampung beberapa method yang dimana di dalam class tersebut memiliki 3 buah method, dimana method def \_\_init\_\_ diatas berfungsi untuk membagi sebuah data dengan fungsi split.

Kami membuat sebuah variable pecah yang menampung nilai x serta menggunakan fungsi split pada kode dimana nanti akan dibagi berdasarkan tanda koma (“,”) dan akan di masukkan pada variable jadwal.

1. **Membuat method di dalam class dengan nama merge()**

    def merge(self, targetUrut, urut, l, m, r):

        n1 = m - l + 1

        n2 = r - m

        L = [0] \* (n1)

        R = [0] \* (n2)

        for i in range(0, n1):

            L[i] = self.jadwal[l + i]

        for j in range(0, n2):

            R[j] = self.jadwal[m + 1 + j]

        i = 0

        j = 0

        k = l

        while i < n1 and j < n2:

            if urut == "ASC":

                if L[i][targetUrut] <= R[j][targetUrut]:

                    self.jadwal[k] = L[i]

                    i += 1

                else:

                    self.jadwal[k] = R[j]

                    j += 1

            elif urut == "DESC":

                if L[i][targetUrut] >= R[j][targetUrut]:

                    self.jadwal[k] = L[i]

                    i += 1

                else:

                    self.jadwal[k] = R[j]

                    j += 1

            else:

                pass

            k += 1

        while i < n1:

            self.jadwal[k] = L[i]

            i += 1

            k += 1

        while j < n2:

            self.jadwal[k] = R[j]

            j += 1

            k += 1

pada kode diatas kami menambahkan sebuah method dialam class jadwalPertandinganBola() dimana kami beri nama merge(). Methode merge(0 ini nantinya menampung kode yang dimana berfungsi untuk mengurutkan berdasarkan inputan yang diinginkan. Didalam method ini terdapat ASC atau ascending yang dimana mengurutkan dari yang terkecil hingga terbesar sedangkan DESC atau descending kebalikan dari ascending yang dimana mengurutkan dari yang terbesar hingga yang terkecil.

1. **Membuat method di dalam class dengan nama mergeSortPertandinganBola()**

    def mergeSortPertandinganBola(self, targetUrut, urut, l, r):

        if l < r:

            # print(l, r)

            m = l + (r - l) // 2

            self.mergeSortPertandinganBola(targetUrut, urut, l, m)

            self.mergeSortPertandinganBola(targetUrut, urut, m + 1, r)

self.merge(targetUrut, urut, l, m, r)

Pada kode diatas kami membuat method dengan bernama mergeSortPertandinganBola() yang nantinya berfungsi membagi inputan.

1. **Run kode**

data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")

print("-------------------------------- Data sebelum diolah.CSV --------------------------------")

tabel = PrettyTable(["No", "Match", "Venue", "Time"])

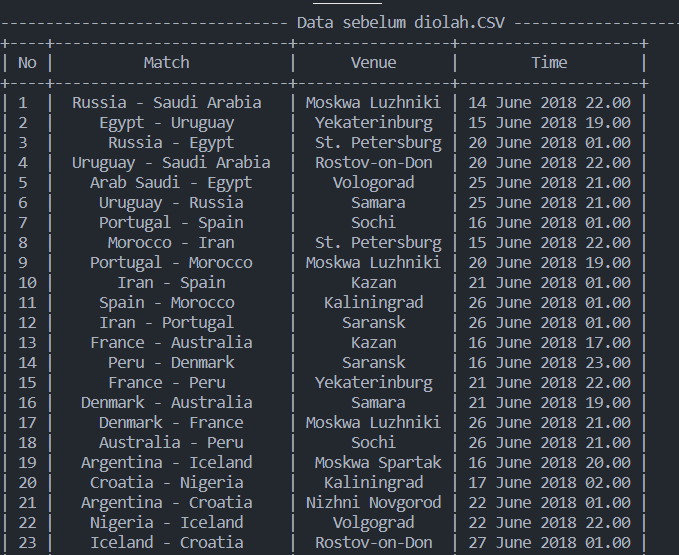
for x in data.jadwal:

    tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])

print(tabel)

Pada kode diatas kita membuat objek untuk memanggil class JadwalPertandinganBola() yang sudah dibuat serta di dalam class tersebut kita menginputkan file data jadwalpertandinganpialadunia2018.csv yang sudah dibuat, dimana nantinya file tersebut dipanggil dan akan di eksekusi. Kami juga membuat sebuah variable tabel yang menampung fungsi PrettyTable yang dimana nanti akan mengeksekusikan sebuah tabel.

Pada kode diatas akan menampilkan sebuah data yang belum diolah dari file .csv tersebut.

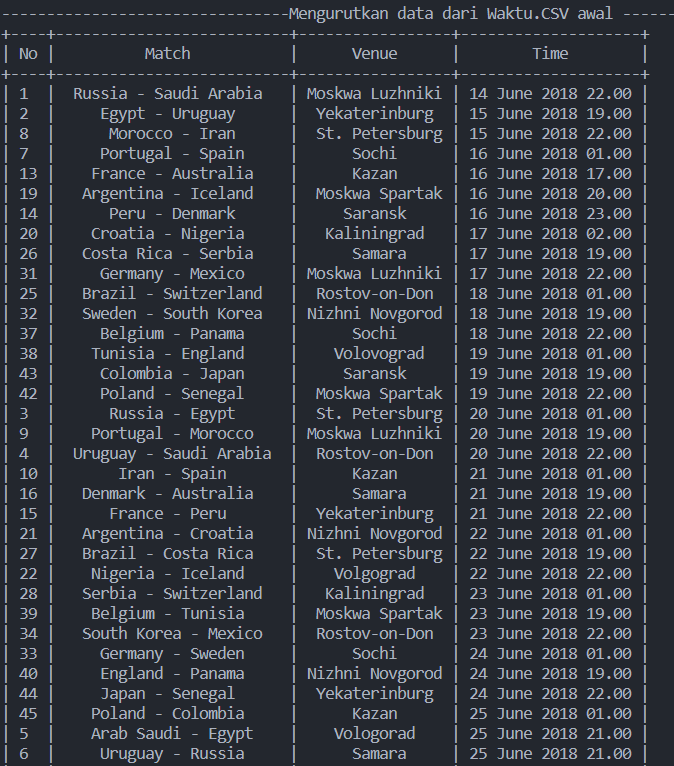


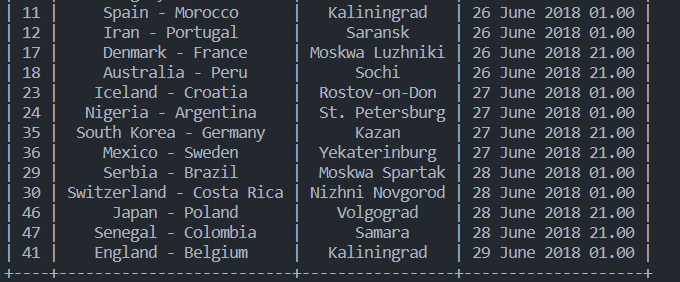


*Gambar diatas hasil run dari kode diatas dimana menampilkan sebuah file data jadwalpertandinganpialadunia.csv yang belum diolah.*

|  |
| --- |
| data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print("--------------------------------Mengurutkan data dari Waktu.CSV awal --------------------------------------")  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  data.mergeSortPertandinganBola("Time", "ASC", 0, len(data.jadwal)-1)  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel) |

Kode diatas hampir sama dengan kode sebelumnya namun perbedaan nya terletak pada inputan serta pemanggilan method mergeSortPertandinganBola dan didalam method tersebut terdapat value “Time” serta “ASC” yang dimana nantinya akan di urutkan berdasarkan tanggal yang tercepat atau pertandingan yang dimulai lebih dulu sampai pertandingan yang tanggal nya lama.

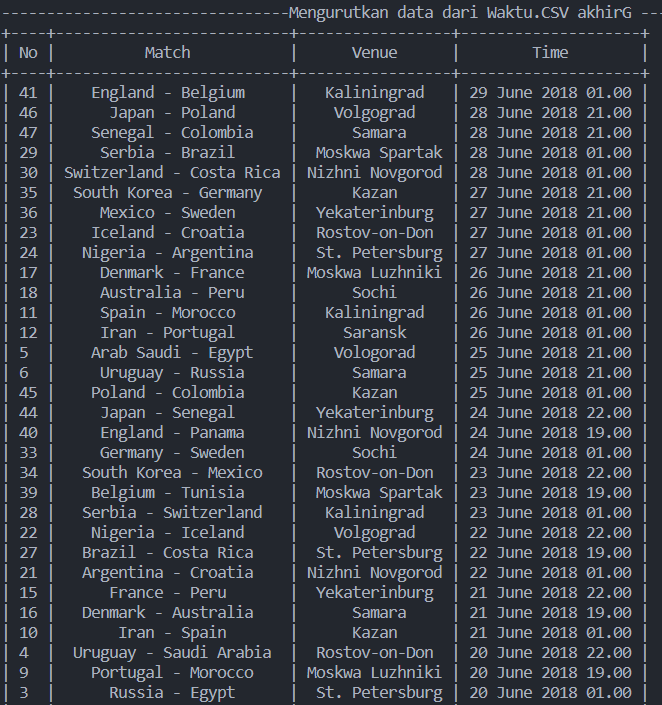


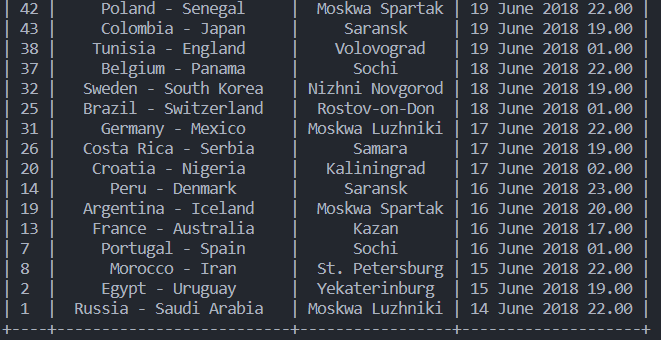


*Gambar diatas menampilkan data yang dimana pada row Time diurutkan berdasarkan tanggal pertandingan yang dimulai lebih dulu hingga sampai ke bawah data tanggal pertandingan yang dimulai terakhir.*

|  |
| --- |
| data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print("--------------------------------Mengurutkan data dari Waktu.CSV akhirG --------------------------------------")  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  data.mergeSortPertandinganBola("Time", "DESC", 0, len(data.jadwal)-1)  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel) |

Pada kode diatas kembalikan dari kode sebelumnya dimana kode tersebut menampilkan data pertandingan dengan waktu yang terakhir bertanding terdahulu dengan inputan DESC yang dimana terbesar ke terkecil.

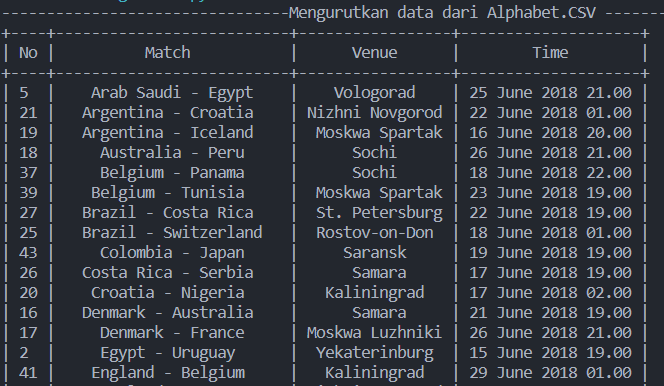


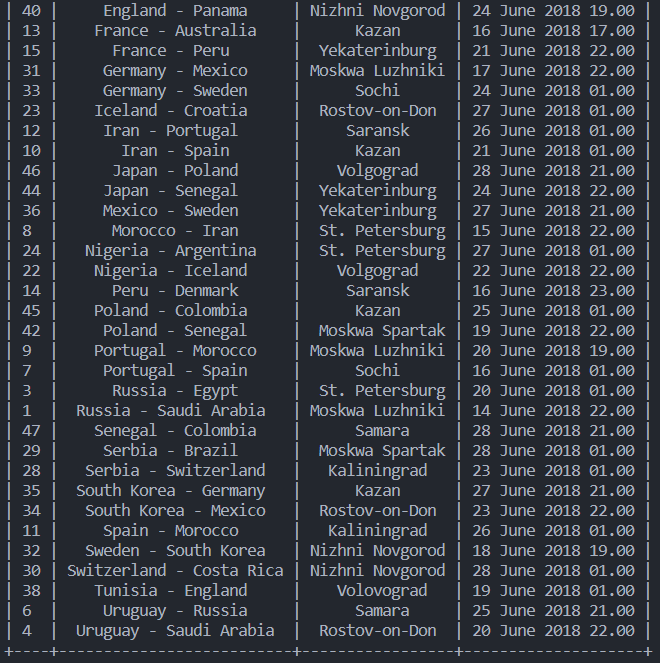


*Gambar diatas menampilkan hasil kode dimana pada column Time diurutkan berdasarkan angka yang lebih besar ke terkecil atau pertandingan yang dimulai terakhir hingga pertandingan yang awal dimulai*

|  |
| --- |
| data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print('--------------------------------Mengurutkan data dari Alphabet.CSV --------------------------------------')  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  data.mergeSortPertandinganBola("Match", "ASC", 0, len(data.jadwal)-1)  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel) |

Pada kode diatas dimana hampir sama dengan kode-kode sebelumnya yang dimana kode sebelumnya mengurutkan dari column Time namun kode ini diurutkan berdasarkan pertandingan atau Match menggunakan fungsi ASC (Ascending) yang akan menampilkan sesuai urutan alphabet.





*Gambar tersebut hasil eksekusi dari kode diatas yang dimana column pertandingan diurutkan berdasarkan Alphabet.*

**Kode lengkapnya**

|  |
| --- |
| from prettytable import PrettyTable  # from pyrsistent import l  class JadwalPertandinganBola(object):      def \_\_init\_\_(self, data):          f = open(data, "r")          self.jadwal = []          for x in f.read().split('\n'):              pecah = x.split(",")              if len(pecah) > 1:                  self.jadwal.append(                      {"No": pecah[0], "Match": pecah[1], "Venue": pecah[2], "Time": pecah[3] + pecah[4]})      # urutkan data berdasarkan targetUrut dan urut      def merge(self, targetUrut, urut, l, m, r):          n1 = m - l + 1          n2 = r - m          L = [0] \* (n1)          R = [0] \* (n2)          for i in range(0, n1):              L[i] = self.jadwal[l + i]          for j in range(0, n2):              R[j] = self.jadwal[m + 1 + j]          i = 0          j = 0          k = l          while i < n1 and j < n2:              if urut == "ASC":                  if L[i][targetUrut] <= R[j][targetUrut]:                      self.jadwal[k] = L[i]                      i += 1                  else:                      self.jadwal[k] = R[j]                      j += 1              elif urut == "DESC":                  if L[i][targetUrut] >= R[j][targetUrut]:                      self.jadwal[k] = L[i]                      i += 1                  else:                      self.jadwal[k] = R[j]                      j += 1              else:                  pass              k += 1          while i < n1:              self.jadwal[k] = L[i]              i += 1              k += 1          while j < n2:              self.jadwal[k] = R[j]              j += 1              k += 1      def mergeSortPertandinganBola(self, targetUrut, urut, l, r):          if l < r:              # print(l, r)              m = l + (r - l) // 2              self.mergeSortPertandinganBola(targetUrut, urut, l, m)              self.mergeSortPertandinganBola(targetUrut, urut, m + 1, r)              self.merge(targetUrut, urut, l, m, r)  data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print("-------------------------------- Data sebelum diolah.CSV --------------------------------")  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel)  # data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print("--------------------------------Mengurutkan data dari Waktu.CSV awal --------------------------------------")  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  data.mergeSortPertandinganBola("Time", "ASC", 0, len(data.jadwal)-1)  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel)  # data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print("--------------------------------Mengurutkan data dari Waktu.CSV akhirG --------------------------------------")  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  data.mergeSortPertandinganBola("Time", "DESC", 0, len(data.jadwal)-1)  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel)  # data = JadwalPertandinganBola("jadwalpertandinganpialadunia2018.csv")  print('--------------------------------Mengurutkan data dari Alphabet.CSV --------------------------------------')  tabel = PrettyTable(["No", "Match", "Venue", "Time"])  data.mergeSortPertandinganBola("Match", "ASC", 0, len(data.jadwal)-1)  for x in data.jadwal:      tabel.add\_row([x["No"], x["Match"], x["Venue"], x["Time"]])  print(tabel) |