

LAPORAN PRAKTIKUM
MACHINE LEARNING MINGGU KE-13

Seven Segment

Dosen Pengampu: Nur Rosyid Mubtadai S.Kom., M.T.



Oleh:

Bayu Kurniawan
(3322600019)

PROGRAM STUDI D4 SAINS DATA TERAPAN
DEPARTEMEN TEKNIK INFORMATIKA DAN KOMPUTER
POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2023

```

import tkinter as tk

# Segmen yang digunakan untuk setiap digit; 0, 1 = mati, hidup.
digits = (
    ( # 0
        ((1, 1, 1, 1, 1, 1, 0), (0, 0, 0, 0, 0, 0, 1)), # on_top, on_bottom
    ),
    ( # 1
        ((0, 1, 1, 0, 0, 0, 0), (1, 0, 0, 1, 1, 1, 1)),
    ),
    ( # 2
        ((1, 1, 0, 1, 1, 0, 1), (0, 0, 1, 1, 0, 1, 0)),
    ),
    ( # 3
        ((1, 1, 1, 1, 0, 0, 1), (0, 0, 1, 1, 1, 0, 0)),
    ),
    ( # 4
        ((0, 1, 1, 0, 0, 1, 1), (1, 0, 0, 1, 1, 0, 1)),
    ),
    ( # 5
        ((1, 0, 1, 1, 0, 1, 1), (0, 1, 1, 1, 1, 0, 0)),
    ),
    ( # 6
        ((1, 0, 1, 1, 1, 1, 1), (0, 1, 1, 1, 0, 1, 1)),
    ),
    ( # 7
        ((1, 1, 1, 0, 0, 0, 0), (0, 0, 0, 0, 1, 1, 1)),
    ),
    ( # 8
        ((1, 1, 1, 1, 1, 1, 1), (0, 1, 1, 1, 1, 1, 0)),
    ),
    ( # 9
        ((1, 1, 1, 1, 0, 1, 1), (0, 1, 1, 1, 0, 0, 1)),
    )
)

class Digit:
    def __init__(self, canvas, x=10, y=10, length=20, width=4):
        self.canvas = canvas
        l = length
        self.segs = []
        # Titik offset untuk menggambarkan segmen
        offsets = (
            (0, 0, 1, 0), # top
            (1, 0, 1, 1), # upper right
            (1, 1, 1, 2), # lower right
            (0, 2, 1, 2), # bottom
            (0, 1, 0, 2), # lower left

```

```

        (0, 0, 0, 1), # upper left
        (0, 1, 1, 1), # middle
    )
    # Membuat segmen dari titik offset yang diberikan
    for x0, y0, x1, y1 in offsets:
        self.segs.append(canvas.create_line(
            x + x0 * l, y + y0 * l, x + x1 * l, y + y1 * l,
            width=width, state='hidden'))

    def show(self, num):
        on_top, on_bottom = num
        for iid, on in zip(self.segs[:7], on_top): # Menampilkan segmen atas
            self.canvas.itemconfigure(iid, state='normal' if on else 'hidden')
        for iid, on in zip(self.segs[7:], on_bottom): # Menampilkan segmen
bawah
            self.canvas.itemconfigure(iid, state='normal' if on else 'hidden')

class SSDisplay:
    def __init__(self, root):
        self.root = root
        self.canvas = tk.Canvas(root, width=300, height=200)
        self.canvas.grid(row=5, column=0, columnspan=7)
        self.segment_labels = [Digit(self.canvas, x=30 + i * 70, y=50) for i
in range(2)]
        self.result_label = tk.Label(root, text="", font=('Arial', 14,
'bold'))
        self.result_label.grid(row=5, column=7, columnspan=2)

    def display_result(self, result):
        try:
            result_int = int(result)
            result_str = str(result_int).zfill(2) # Mengonversi hasil menjadi
string dengan minimal 2 digit

            if not 0 <= result_int <= 99: # Menyaring hasil yang tidak valid
                raise ValueError("Result out of range (0-99)")

            print(f"Result Integer: {result_int}")
            print(f"Result String: {result_str}")

            for i in range(2):
                digit_num = int(result_str[i])
                self.show_digit(i, digit_num)

            self.result_label.config(text=f"Result: {result_int}")
        except (ValueError, TypeError) as e:
            print(f"Error: {e}")
            self.result_label.config(text="Invalid input or calculation")

```

```

def show_digit(self, index, digit_num):
    # Menampilkan digit pada seven-segment display
    digit_data = digits[digit_num][0] # Ambil data segmen untuk digit
    digit_segment = self.segment_labels[index]
    digit_segment.show(digit_data)

def calculate():
    try:
        num1 = int(entry_num1.get())
        num2 = int(entry_num2.get())
        operator = operator_var.get()

        # Operasi berdasarkan operator yang dipilih
        if operator == "+":
            result = num1 + num2
        elif operator == "-":
            result = num1 - num2
        elif operator == "*":
            result = num1 * num2
        elif operator == "/":
            result = num1 / num2 if num2 != 0 else "Infinity"
        else:
            result = "Invalid operator"

        print(f"Operator: {operator}")
        print(f"Result: {result}")

        ss_display.display_result(result)
    except ValueError:
        print("Invalid input")
        ss_display.display_result("Invalid input")
    except ZeroDivisionError:
        print("Cannot divide by zero")
        ss_display.display_result("Cannot divide by zero")

root = tk.Tk()

# Entry untuk angka pertama
label_num1 = tk.Label(root, text="Number 1:")
label_num1.grid(row=0, column=0)
entry_num1 = tk.Entry(root)
entry_num1.grid(row=0, column=1)

# Entry untuk angka kedua
label_num2 = tk.Label(root, text="Number 2:")
label_num2.grid(row=1, column=0)
entry_num2 = tk.Entry(root)

```

```

entry_num2.grid(row=1, column=1)

# Operator selection
operator_var = tk.StringVar()
operator_var.set("+")
operator_label = tk.Label(root, text="Operator:")
operator_label.grid(row=2, column=0)
operator_menu = tk.OptionMenu(root, operator_var, "+", "-", "*", "/")
operator_menu.grid(row=2, column=1)

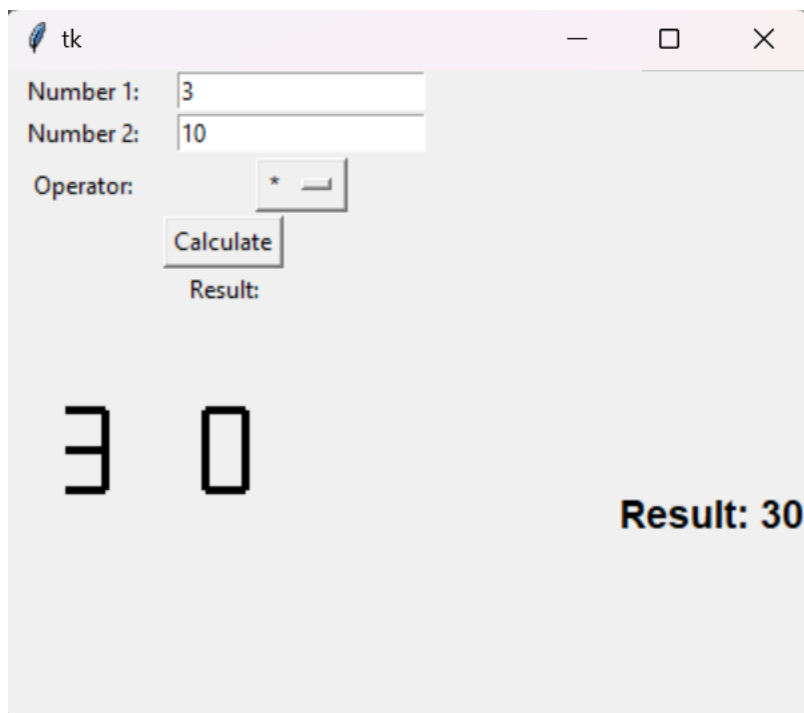
# Calculate button
calculate_button = tk.Button(root, text="Calculate", command=calculate)
calculate_button.grid(row=3, column=0, columnspan=2)

# Result label
result_label = tk.Label(root, text="Result:")
result_label.grid(row=4, column=0, columnspan=2)

# Seven-segment display
ss_display = SSDisplay(root)

# Run GUI loop
root.mainloop()

```



tk

Number 1:

Number 2:

Operator:

Result:

1 3

Result: 13

tk

Number 1:

Number 2:

Operator:

Result:

2 0

Result: 20

Kode ini adalah program Python yang menggunakan modul Tkinter untuk membuat antar muka pengguna grafis (GUI) sederhana. Program ini memungkinkan pengguna untuk memasukkan dua bilangan dan sebuah operator matematika (+, -, *, /), kemudian melakukan operasi matematika pada dua bilangan tersebut dan menampilkan hasilnya menggunakan tampilan tujuh segmen (seven-segment display).

Berikut adalah analisis komponen utama dari kode tersebut:

Kelas ``Digit``: Kelas ini bertanggung jawab untuk menggambar digit menggunakan segmen-segmen pada seven-segment display.

Kelas ``SSDisplay``: Kelas ini digunakan untuk menampilkan hasil operasi matematika pada seven-segment display.

- ``__init__``: Membuat GUI dengan menggunakan Tkinter.
- ``display_result``: Menerima hasil dari operasi matematika dan menampilkannya pada seven-segment display.
- ``show_digit``: Mengatur tampilan digit pada seven-segment display.

Fungsi ``calculate``: Menangani operasi matematika (penambahan, pengurangan, perkalian, pembagian) berdasarkan input yang dimasukkan oleh pengguna.

GUI: Terdapat input untuk dua bilangan (``entry_num1`` dan ``entry_num2``), pemilihan operator menggunakan ``OptionMenu``, tombol ``Calculate``, dan label untuk menampilkan hasil operasi.

Eksekusi Program:

- Membuat instance dari ``Tk`` (root window) untuk menampilkan GUI.
- Memulai GUI loop dengan menggunakan ``root.mainloop()``.

Selain itu, terdapat penggunaan fungsi ``print`` untuk debugging dan penanganan kesalahan seperti ``ValueError`` dan ``ZeroDivisionError`` untuk menangani situasi di mana input tidak valid atau terjadi pembagian dengan nol.