



Praktikum Kecerdasan Buatan

(Praktikum)

TI – B4

FAKULTAS VOKASI
UNIVERSITAS AIRLANGGA

Oleh :
Andyka Salom 434221054

Cara bermain Game:

1. PerCmbangkan semua kemungkinan jalan
2. Berikan nilai pada semua kemungkinan jalan
3. Jalankan kemungkinan terbaik
4. Tunggu giliran pihak lawan
5. Ulangi tahapan-tahapan diatas

Penjelasan :

1. Import Library :

```
import tkinter as tk
import random
```

Dua library yang diimpor di sini adalah `tkinter` untuk membuat antarmuka pengguna grafis (GUI) dan `random` untuk menghasilkan pilihan acak.

2. Fungsi print_board :

```
def print_board(board):
    for row in board:
        print(" | ".join(row))
        print("-" * 9)
```

Fungsi ini digunakan untuk mencetak papan permainan Tic Tac Toe ke konsol. Ini mencetak setiap baris dari papan dengan pemisah '|' antara sel dan garis horizontal untuk memisahkan setiap baris.

3. Fungsi check_winner :

```
def check_winner(board):
    for i in range(3):
        if board[i][0] == board[i][1] == board[i][2] != " ":
            return board[i][0]
        if board[0][i] == board[1][i] == board[2][i] != " ":
            return board[0][i]

    if board[0][0] == board[1][1] == board[2][2] != " ":
        return board[0][0]
    if board[0][2] == board[1][1] == board[2][0] != " ":
        return board[0][2]

    return None
```

Fungsi ini memeriksa papan permainan untuk menentukan apakah ada pemenang atau tidak. Ini memeriksa baris, kolom, dan diagonal untuk kondisi menang.

4. Fungsi `get_empty_cells` :

```
def get_empty_cells(board):  
    empty_cells = []  
    for i in range(3):  
        for j in range(3):  
            if board[i][j] == " "  
                empty_cells.append((i, j))  
    return empty_cells
```

Fungsi ini mengembalikan sel-sel kosong di papan permainan, yang memungkinkan pemain untuk melakukan langkah selanjutnya.

5. Fungsi `handle_click` :

```
def handle_click(row, col):  
    global current_player, board_buttons, game_over  
  
    if board[row][col] != " " or game_over:  
        return  
  
    board[row][col] = current_player  
    board_buttons[row][col].config(text=current_player)  
  
    winner = check_winner(board)  
    if winner:  
        status_label.config(text=f"Pemain {winner} menang!")  
        game_over = True  
        return  
  
    if len(get_empty_cells(board)) == 0:  
        status_label.config(text="Permainan seri!")  
        game_over = True  
        return  
  
    current_player = "X" if current_player == "O" else "O"  
    status_label.config(text=f"Giliran pemain {current_player}")
```

Fungsi ini dipanggil ketika seorang pemain mengklik salah satu tombol pada papan permainan. Ini mengatur simbol pemain yang sesuai pada sel yang diklik, memeriksa apakah ada pemenang setelah setiap langkah, dan mengubah giliran pemain setelah setiap langkah.

6. Fungsi restart_game :

```
def restart_game():
    global current_player, board, board_buttons, game_over

    current_player = random.choice(["X", "O"])
    board = [[" " for _ in range(3)] for _ in range(3)]
    game_over = False

    for i in range(3):
        for j in range(3):
            board_buttons[i][j].config(text=" ", state="normal")

    status_label.config(text=f"Giliran pemain {current_player}")
```

Fungsi ini digunakan untuk me-reset permainan, mengatur ulang papan permainan dan variabel-variabel terkait ke keadaan awal.

7. Inisialisasi GUI :

```
root = tk.Tk()
root.title("Tic Tac Toe")
```

Inisialisasi jendela utama untuk permainan Tic Tac Toe.

8. Inisialisasi Papan Permainan dan Variabel :

```
board = [[" " for _ in range(3)] for _ in range(3)]
current_player = random.choice(["X", "O"])
game_over = False
```

Membuat papan permainan awal dengan seluruhnya kosong, memilih pemain pertama secara acak, dan mengatur status permainan menjadi belum berakhir.

9. Membuat Tombol-tombol Papan Permainan :

```
board_buttons = [[None for _ in range(3)] for _ in range(3)]
for i in range(3):
    for j in range(3):
        board_buttons[i][j] = tk.Button(root, text=" ",
                                         font=("Helvetica", 20), width=4, height=2,
                                         command=lambda row=i, col=j:
handle_click(row, col))
        board_buttons[i][j].grid(row=i, column=j)
```

Membuat tombol-tombol yang mewakili sel-sel pada papan permainan, dengan setiap tombol dipasangkan dengan fungsi `handle_click` ketika diklik.

10. Menampilkan Label Status :

```
status_label = tk.Label(root, text=f"Giliran pemain {current_player}",  
font=("Helvetica", 12))  
status_label.grid(row=3, columnspan=3)
```

Label ini menampilkan status permainan, seperti giliran pemain atau hasil permainan.

11. Tombol Restart :

```
restart_button = tk.Button(root, text="Restart", font=("Helvetica",  
12), command=restart_game)  
restart_button.grid(row=4, columnspan=3)
```

Tombol ini memungkinkan pemain untuk me-reset permainan jika ingin memulai ulang dari awal.

12. Memulai Loop Utama Tkinter :

```
root.mainloop()
```

Memulai loop utama Tkinter yang akan menampilkan jendela GUI dan menangani interaksi pengguna.