STRUKTUR DATA

Dosen Pengampu :

**Edy Haryanto, S.Kom.,M.Pd**



Disusun Oleh:

Zofita Amalia Wati (23241085)

RaimunDus Ama Suba Sabon (23241112)

Dinda Damayanti putri (23241093)

Zainul Haqqi Mr (23241115)

**KELAS C**

**PROGRAM STUDI PENDIDIKAN PENDIDIKAN TEKNOLOGI INFORMASI**

**FAKULTAS SAINS, TEKNOLOGI DAN TERAPAN**

**UNIVERSITAS PENDIDIKAN MANDALIKA MATARAM**

**TAHUN 2023/2024**

### Pendahuluan

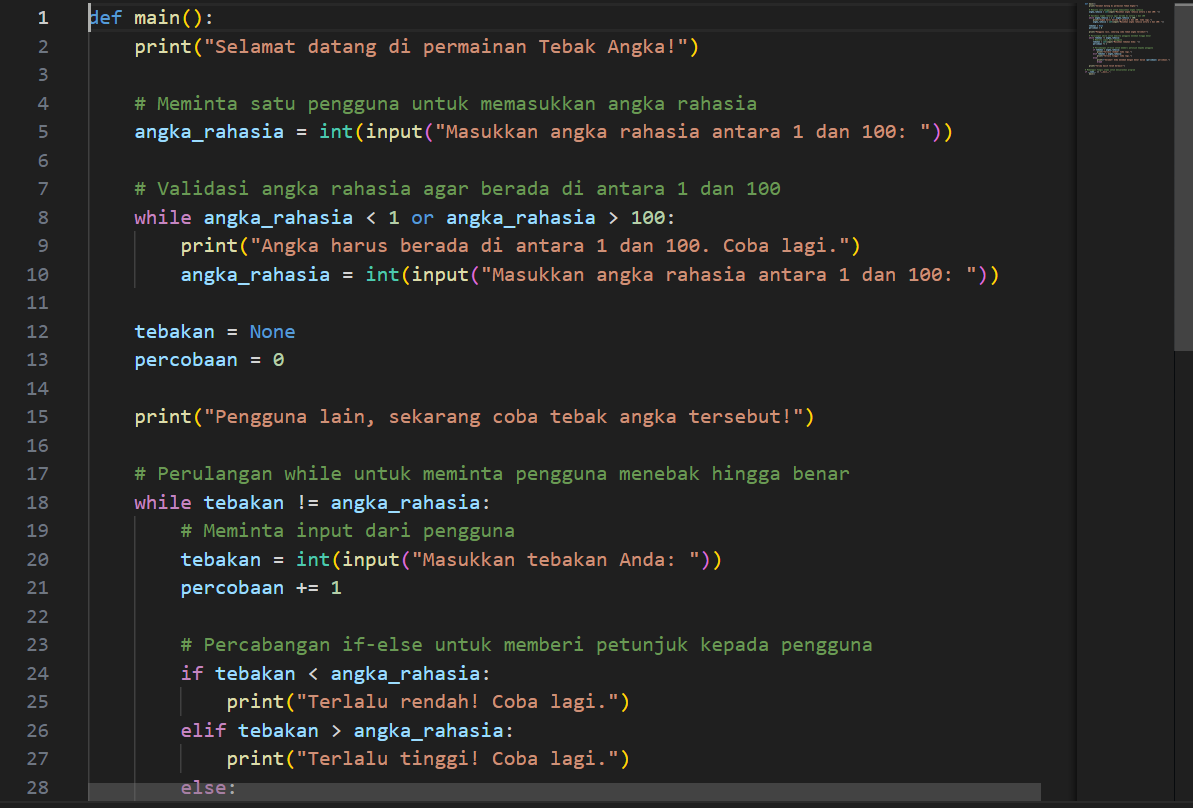
Struktur data adalah salah satu konsep dasar dalam ilmu komputer yang sangat penting untuk pengelolaan dan manipulasi data. Dua komponen utama dalam struktur data adalah percabangan (branching) dan perulangan (looping). Kedua konsep ini memainkan peran krusial dalam mengontrol alur eksekusi program, memungkinkan pengambilan keputusan dan pengulangan tugas tertentu.

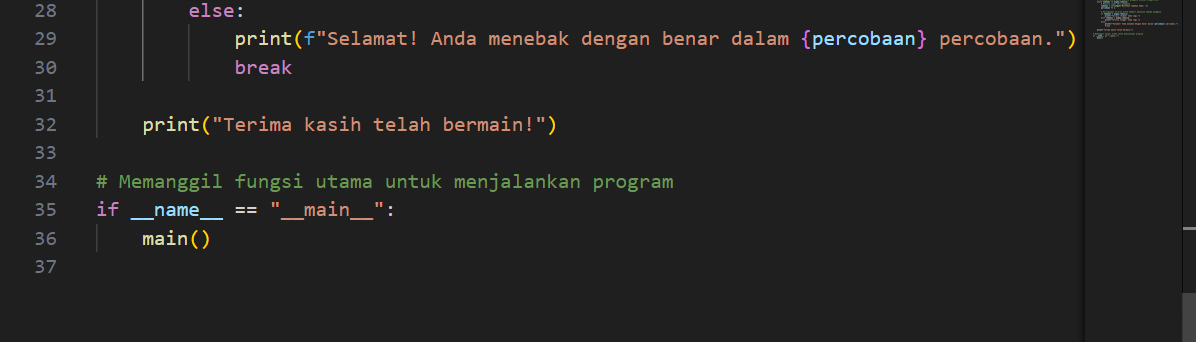
### Tujuan Makalah

Makalah ini bertujuan untuk menjelaskan konsep dasar dari percabangan dan perulangan dalam struktur data, mengilustrasikan penggunaannya dalam berbagai bahasa pemrograman, dan menunjukkan pentingnya dalam pengembangan perangkat lunak.

### Percabangan (Branching)

Percabangan adalah struktur kontrol yang memungkinkan program untuk memilih di antara beberapa jalur eksekusi berdasarkan kondisi tertentu. Terdapat beberapa bentuk percabangan, antara lain:

****

****

**PENJELASAN**

Berikut adalah penjelasan setiap baris dari program "Tebak Angka Sederhana dengan Inputan Angka Rahasia":

* Baris 1: def main():

- Mendefinisikan fungsi `main()`, yang merupakan fungsi utama yang akan dijalankan ketika program dimulai.

* Baris 2: print("Selamat datang di permainan Tebak Angka!")

- Menampilkan pesan selamat datang kepada pengguna.

* Baris 5: angka\_rahasia = int(input("Masukkan angka rahasia antara 1 dan 100: "))

- Meminta satu pengguna untuk memasukkan angka rahasia antara 1 dan 100. Input dari pengguna dikonversi menjadi integer dan disimpan dalam variabel `angka\_rahasia`.

* Baris 8: while angka\_rahasia < 1 or angka\_rahasia > 100:
* Baris 9: print("Angka harus berada di antara 1 dan 100. Coba lagi.")
* Baris 10: angka\_rahasia = int(input("Masukkan angka rahasia antara 1 dan 100: "))

- Memastikan bahwa angka rahasia berada di antara 1 dan 100. Jika angka yang dimasukkan tidak valid, program akan meminta input ulang hingga angka yang valid dimasukkan.

* Baris 12: tebakan = None
* Baris 13: percobaan = 0

- Menginisialisasi variabel `tebakan` dengan nilai `None` (belum ada tebakan) dan `percobaan` dengan nilai 0 (belum ada percobaan).

* Baris 15: print("Pengguna lain, sekarang coba tebak angka tersebut!")

- Menampilkan pesan untuk pengguna lain yang akan menebak angka rahasia.

* Baris 18: while tebakan != angka\_rahasia:

- Memulai perulangan `while` yang akan terus berjalan selama `tebakan` tidak sama dengan `angka\_rahasia`.

* Baris 20: tebakan = int(input("Masukkan tebakan Anda: "))

- Meminta pengguna untuk memasukkan tebakan mereka, mengonversi input menjadi integer, dan menyimpannya dalam variabel `tebakan`.

* Baris 21: percobaan += 1

- Menambah nilai `percobaan` sebesar 1 setiap kali pengguna melakukan tebakan.

* Baris 24: if tebakan < angka\_rahasia:

- Memeriksa apakah tebakan pengguna lebih kecil dari angka rahasia.

* Baris 25: print("Terlalu rendah! Coba lagi.")

- Jika tebakan lebih kecil, menampilkan pesan "Terlalu rendah! Coba lagi."

* Baris 26: elif tebakan > angka\_rahasia:

- Memeriksa apakah tebakan pengguna lebih besar dari angka rahasia.

* Baris 27: print("Terlalu tinggi! Coba lagi.")

- Jika tebakan lebih besar, menampilkan pesan "Terlalu tinggi! Coba lagi."

* Baris 28: else:

- Jika tebakan sama dengan angka rahasia.

* Baris 29: print(f"Selamat! Anda menebak dengan benar dalam {percobaan} percobaan.")

break

- Menampilkan pesan selamat dengan jumlah percobaan yang dilakukan pengguna, dan keluar dari perulangan `while` dengan menggunakan `break`.

* Baris 32: print("Terima kasih telah bermain!")

- Menampilkan pesan terima kasih setelah pengguna berhasil menebak angka rahasia.

* Baris 35: if \_\_name\_\_ == "\_\_main\_\_":

main()

- Memeriksa apakah skrip dijalankan langsung (bukan diimpor sebagai modul). Jika iya, maka memanggil fungsi `main()` untuk menjalankan program.

**KESIMPULAN**

Percabangan dan perulangan adalah konsep fundamental dalam struktur data yang memungkinkan program untuk membuat keputusan dan mengulang tugas dengan efisien. Memahami dan menguasai penggunaan percabangan dan perulangan adalah langkah penting dalam pengembangan perangkat lunak yang efektif dan efisien

**Daftar Pustaka**

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). Introduction to Algorithms. MIT Press.

Kernighan, B. W., & Ritchie, D. M. (1988). The C Programming Language. Prentice Hall.

Lutz, M. (2013). Learning Python. O'Reilly Media.