Algoritma dan Struktur Data

Apa itu Algoritma?

Algoritma adalah sekumpulan langkah-langkah sistematis untuk menyelesaikan masalah atau tugas.

Mengapa Algoritma Penting?

- Algoritma membantu kita menjalankan tugas-tugas dengan efisien.
- Meningkatkan pemahaman pemrograman.
- Penting dalam pengembangan perangkat lunak dan rekayasa sistem.

Karakteristik Algoritma

- **Masukan (Input)**: Algoritma menerima data sebagai masukan, yang mungkin berupa nol, satu, atau lebih.
- **Keluaran (Output)**: Algoritma menghasilkan data sebagai keluaran, yang juga bisa berupa nol, satu, atau lebih.
- **Kejelasan (Definiteness)**: Setiap langkah algoritma harus jelas dan spesifik.
- **Terbatas (Finiteness)**: Algoritma harus memiliki jumlah langkah yang terbatas.
- **Efektivitas (Effectiveness)**: Algoritma harus dapat diimplementasikan dalam praktek dengan sumber daya yang tersedia.

Jenis-Jenis Algoritma

1. **Algoritma Pencarian**: Mencari elemen tertentu dalam data.
2. **Algoritma Pengurutan**: Mengurutkan data dalam urutan tertentu.
3. **Algoritma Rekursi**: Algoritma yang memanggil dirinya sendiri.
4. **Algoritma Greedy**: Memilih opsi terbaik di setiap langkah.
5. **Algoritma Dinamis**: Memecah masalah menjadi submasalah yang lebih kecil.

Contoh Algoritma: Penambahan Dua Angka

Masukan: Dua angka, misalnya 5 dan 3.
Keluaran: Hasil penambahan, yaitu 8.

Langkah-langkah Algoritma:

1. Mulai.
2. Masukkan angka pertama (contoh: 5).
3. Masukkan angka kedua (contoh: 3).
4. Hitung hasil penambahan dari kedua angka: 5 + 3 = 8.
5. Tampilkan hasilnya (8).
6. Selesai.

Contoh Algoritma: Menentukan Bilangan Genap atau Ganjil

Masukan: Sebuah angka, misalnya 7.
Keluaran: Pesan "Bilangan Ganjil" atau "Bilangan Genap."

Langkah-langkah Algoritma:

1. Mulai.
2. Masukkan angka (contoh: 7).
3. Periksa apakah angka tersebut habis dibagi 2.
4. Jika habis, tampilkan pesan "Bilangan Genap."
5. Jika tidak habis, tampilkan pesan "Bilangan Ganjil."
6. Selesai.

Apa itu Struktur Data

- **Struktur Data** adalah cara untuk menyimpan dan mengatur data dalam komputer.
- Contoh struktur data: array, list, queue, stack, dan lainnya.

Struktur Data Umum

1. **Array**: Kumpulan elemen data dengan indeks.
2. **List**: Kumpulan elemen data yang dapat diubah.
3. **Queue**: Antrian data dengan prinsip FIFO.
4. **Stack**: Tumpukan data dengan prinsip LIFO.
5. **Linked List**: Kumpulan elemen yang saling terhubung.
6. **Tree**: Struktur hirarkis dengan akar dan cabang.
7. **Graph**: Kumpulan simpul yang terhubung dengan tepi.

Contoh Struktur Data: Array

Deskripsi: Array adalah struktur data yang berisi elemen-elemen dengan indeks numerik.
Contoh: Array berisi angka-angka `[3, 1, 4, 1, 5]`.

Operasi Pada Array:

1. Mengakses elemen berdasarkan indeks (misalnya, elemen ke-2 adalah 4).
2. Menambahkan elemen baru (misalnya, menambahkan 9 ke belakang array).
3. Menghapus elemen (misalnya, menghapus elemen pertama).
4. Menghitung jumlah elemen dalam array (jumlah elemen: 5).

Contoh Struktur Data: Antrian (Queue)

Deskripsi: Antrian adalah struktur data yang mengikuti prinsip FIFO (First-In-First-Out).
Contoh: Antrian pelanggan di loket pelayanan.

Operasi Pada Antrian:

1. Menambahkan elemen ke belakang antrian (misalnya, pelanggan baru masuk).
2. Menghapus elemen dari depan antrian (pelanggan pertama dilayani).
3. Melihat pelanggan yang berada di depan antrian (pelanggan berikutnya yang akan dilayani).

Code Examples

```
function addNumbers(num1, num2) {
  return num1 + num2;
}

const firstNumber = 5;
const secondNumber = 10;

const sum = addNumbers(firstNumber, secondNumber);

console.log(`The sum of ${firstNumber} and ${secondNumber} is: ${sum}`);
```

Code Examples

```
function concatenateStrings(str1, str2) {
  return str1 + " " + str2;
}

const firstName = "John";
const lastName = "Doe";

const fullName = concatenateStrings(firstName, lastName);

console.log(`Full name: ${fullName}`);
```

Code Examples

```
function findMaximum(num1, num2) {
  return Math.max(num1, num2);
}

const number1 = 7;
const number2 = 12;

const maxNumber = findMaximum(number1, number2);

console.log(`The maximum of ${number1} and ${number2} is: ${maxNumber}`);
```

Code Examples

```
// Define an array of names
const names = ["Alice", "Bob", "Charlie", "David", "Eve"];

// Access and manipulate the array
console.log("List of Names:", names);

// Add a new name to the end of the array
names.push("Frank");
console.log("After Adding 'Frank':", names);

// Remove the last name from the array
names.pop();
console.log("After Removing the Last Name:", names);

// Access a specific name by index
const secondName = names[1];
console.log("The Second Name in the List is:", secondName);

// Find the number of names in the array
const numberOfNames = names.length;
console.log("Number of Names in the List:", numberOfNames);
```

Latihan

Latihan 1: Membalikkan String

Tulis sebuah fungsi yang mengambil sebuah string sebagai masukan dan mengembalikan string tersebut dalam keadaan terbalik.

Latihan 2: Pemeriksa Palindrom

Buat sebuah fungsi yang memeriksa apakah string yang diberikan adalah palindrom (dibaca sama ke depan dan ke belakang). Abaikan spasi, tanda baca, dan huruf besar.

Latihan 3: Deret Fibonacci

Tuliskan sebuah fungsi untuk menghasilkan deret angka Fibonacci ke-n.