

Nama: Anam Sadat (14522708)

Kelas: Teknik Informatika

Mata Kuliah: Struktur Data

## Modul 1 Array, Poiter dan Stuktur data

1. Jelaskan perbedaan array, pointer dan struktur data!
2. Jelaskan deklarasi array dan cara mengakses array
3. Jelaskan deklarasi pointer dan cara mengakses pointer
4. Jelasnkan deklarasi struktur dan cara mengakses struktur

Jawab:

### 1. Array:

Array adalah kumpulan elemen data yang memiliki tipe data yang sama dan diidentifikasi oleh indeks atau subskrip.

Ciri-ciri:

1. Elemen-elemen array disimpan secara berurutan di dalam memori.
2. Ukuran array tetap setelah deklarasinya.
3. Menggunakan indeks numerik untuk mengakses elemen-elemen di dalamnya.

Pointer:

Definisi: Pointer adalah variabel yang menyimpan alamat memori dari suatu variabel atau objek.

Ciri-ciri:

1. Mengacu pada lokasi memori variabel atau objek lain.
2. Memungkinkan manipulasi langsung terhadap nilai di lokasi memori tersebut.
3. Tipe data pointer harus sesuai dengan tipe data variabel yang ditunjuk.

Struktur Data:

Definisi: Struktur data adalah cara menggabungkan beberapa variabel dengan tipe data berbeda di bawah satu nama.

Ciri-ciri:

1. Memungkinkan pengelompokan data yang terkait menjadi satu kesatuan.
2. Setiap elemen diakses melalui nama struktur dan operator titik.
3. Digunakan untuk merepresentasikan objek yang lebih kompleks.

2. `int angka[5];`

```
// Mengisi array
```

```
angka[0] = 10;
```

```
angka[1] = 20;
```

```
angka[2] = 30;
```

```
angka[3] = 40;
```

```
angka[4] = 50;
```

```
// Mencetak array
```

```
for (int i = 0; i < 5; i++) {
```

```
    cout << angka[i] << endl;
```

```
}
```

3. `int a = 10;`

```
int *p = &a;
```

```
// Mengakses nilai variabel melalui pointer
```

```
cout << *p << endl;
```

```
struct mahasiswa {
```

```
    int nim;
```

```
    char nama[30];
```

```
    float ipk;
```

```
};
```

4. struct mahasiswa mhs1;

// Mengisi variabel struktur data

mhs1.nim = 12345;

strcpy(mhs1.nama, "Andi");

mhs1.ipk = 3.5;

// Mencetak variabel struktur data

cout << "NIM: " << mhs1.nim << endl;

cout << "Nama: " << mhs1.nama << endl;

## Modul 2 Stuck

1. Buatlah konversi bilangan decimal ke biner dan heksa menggunakan stack
2. Buatlah pembalik kalimat menggunakan stack

Contoh : Struktur data

ataD rutkurtS

buatlah pengecekan palindrom atau bukan, Contoh: sugus bila dibuatlah pembalik kalimat menggunakan stack

3. Implementasikan notasi infix ke postfix menggunakan stack
4. Implementasikan operasi notasi postfix menggunakan stack

## Modul 3 Queue

Buatlah operasi shift menggunakan queue

- Input: bilangan decimal dan jumlah shift
- Output bilangan decimal setelah shift

Contoh:

Masukan Bilangan decimal: 25

Masukan jumlah shift: 3

Bilangan decimal setelah shift: 7

Jawab:

## Modul 4 Single Linked list

1. Buatlah single linked list dengan data tipe integer yang dapat melakukan operasi sisip secara terurut dan hapus simpul tertentu dengan ketentuan sebagai berikut:
  - a. Operasi sisip, buat sebuah simpul baru
    - Jika data simpul baru < data pada head maka gunakan sisip awal list
    - Jika pencarian data simpul baru mencapai NULL (data belum ada), sisip akhir list
    - Jika data simpul baru = data pada simpul tertentu maka berikan pesan simpul sudah ada (duplikat)
    - Lainnya sisip sebelum simpul tertentu
  - b. Operasi hapus
    - Jika posisi simpul yang dihapus pada head, gunakan hapus awal list
    - Jika posisi simpul yang dihapus pada tail, gunakan hapus akhir list
    - Lainnya hapus simpul Tengah
2. Implementasikan stack dengan single linked list, buatlah menu push, pop dan tampil
3. Buatlah single linked list dengan simpul berupa data mahasiswa yang terdiri dari NRP, nama dan kelas. Buatlah operasi sisip secara terurut, Hapus data mahasiswa tertentu dan update data (nama dan kelas saja)

Jawab:

## Modul 5 Double linked list

1. Buatlah double linked list dengan data tipe integer yang dapat melakukan operasi sisip secara terurut dan hapus simpul tertentu dengan ketentuan sebagai berikut:
  - a. Operasi sisip, buat sebuah simpul baru
    - Jika data simpul baru < data pada head maka gunakan sisip awal list
    - Jika pencarian data simpul baru mencapai NULL (data belum ada), sisip akhir list
    - Jika data simpul baru = data pada simpul tertentu maka berikan pesan simpul sudah ada (duplikat)
    - Lainnya sisip sebelum simpul tertentu
  - b. Operasi hapus
    - Jika posisi simpul yang dihapus pada head, gunakan hapus awal list
    - Jika posisi simpul yang dihapus pada tail, gunakan hapus akhir list
    - Lainnya hapus simpul Tengah

2. Implementasikan Queue dengan double linked list. Buatlah menu enqueue dan dequeue dan tampil
3. Buatlah double linked list dengan simpul berupa data mahasiswa yang terdiri dari NRP, nama dan kelas. Buatlah operasi sisip secara terurut, Hapus data mahasiswa tertentu dan update data (nama dan kelas saja)

## Modul 6 Rekursi

1. Buatlah fungsi yang menulis angka dari n ke 0 dengan menggunakan proses rekursi
2. Tuliskan sebuah fungsi untuk menulis angka dari 0 ke n dengan menggunakan proses rekursi
3. Tuliskan sebuah fungsi rekursi yang melakukan pengecekan apakah sebuah elemen x merupakan anggota dari sebuah array a[n].
4. Tulis sebuah fungsi yang melakukan pengecekan apakah sebuah angka merupakan bilangan prima atau bukan(n bukan bilangan prima jika dapat dibagi dengan angka kurang dari n)

Jawab:

## Modul 7 Insertion sort dan selection sort

- a. Urutkan data dibawah ini dengan algoritma insertion dan selection sort, jelaskan pula Langkah-langkahnya
- b. 9 1 2 5 6 4

Jawab:

### Langkah-langkah Insertion Sort:

1. Pada iterasi pertama, anggap elemen pertama sudah diurutkan.
2. Ambil elemen berikutnya dan bandingkan dengan elemen-elemen yang sudah diurutkan.
3. Pindahkan elemen-elemen yang lebih besar dari elemen yang sedang diperiksa ke posisi setelahnya.
4. Tempatkan elemen yang sedang diperiksa ke posisi yang sesuai dalam urutan.
5. Ulangi langkah 2-4 untuk setiap elemen yang belum diurutkan.

### Urutan Data:

Langkah 1: [1] 9 2 5 6 4 (Urutan: 1)

Langkah 2: [1 2] 9 5 6 4 (Urutan: 1 2)

Langkah 3: [1 2 5] 9 6 4 (Urutan: 1 2 5)

Langkah 4: [1 2 5 6] 9 4 (Urutan: 1 2 5 6)

Langkah 5: [1 2 4 5 6] 9 (Urutan: 1 2 4 5 6 9)

#### Langkah-langkah Selection Sort:

1. Pada iterasi pertama, temukan elemen minimum dalam seluruh array.
2. Tukar elemen minimum dengan elemen pertama.
3. Ulangi proses untuk sisa array, kecuali elemen pertama yang sudah terurut.
4. Teruskan proses ini hingga seluruh array terurut.

#### Urutan Data:

Langkah 1: [1] 9 2 5 6 4 (Urutan: 1)

Langkah 2: [1 2] 9 5 6 4 (Urutan: 1 2)

Langkah 3: [1 2 4] 5 6 9 (Urutan: 1 2 4)

Langkah 4: [1 2 4 5] 6 9 (Urutan: 1 2 4 5)

Langkah 5: [1 2 4 5 6] 9 (Urutan: 1 2 4 5 6 9)

## Modul 8 Bubble sort dan shell sort

- a. Urutkan data dibawah ini dengan algoritma bubble sort dan shell sort, jelaskan pula Langkah-langkahnya
- b. 9 1 2 5 6 4

Jawab:

#### Langkah-langkah Bubble Sort:

1. Bandingkan dua elemen sebelah adjacent, jika elemen di sebelah kanan lebih kecil, tukar posisi keduanya.
2. Lakukan langkah 1 untuk setiap pasangan elemen dalam array.
3. Setelah satu iterasi, elemen terbesar sudah berada di ujung kanan.
4. Ulangi langkah 1-3 untuk seluruh array, kecuali elemen yang sudah berada di ujung kanan.
5. Teruskan proses ini hingga seluruh array terurut.

#### Urutan Data:

Langkah 1: [1 9] 2 5 6 4 (Urutan: 1 9)

Langkah 2: [1 2] 9 5 6 4 (Urutan: 1 2)

Langkah 3: [1 2 5] 9 6 4 (Urutan: 1 2 5)

Langkah 4: [1 2 5 6] 9 4 (Urutan: 1 2 5 6)

Langkah 5: [1 2 5 6 4] 9 (Urutan: 1 2 5 6 4)

Langkah 6: [1 2 5 6 4 9] (Urutan: 1 2 5 6 4 9)

Langkah-langkah Shell Sort:

1. Tentukan nilai gap yang akan digunakan (misalnya,  $n/2$ ,  $n/4$ , dst.).
2. Lakukan pengurutan dengan algoritma insertion sort menggunakan gap tersebut.
3. Kurangi nilai gap dan ulangi langkah 2.
4. Teruskan proses ini hingga gap mencapai 1.

Urutan Data:

Langkah 1 (gap = 3): [5 1 2] 9 6 4 (Urutan: 2 1 5 9 6 4)

Langkah 2 (gap = 1): [1 2 4 5 6 9] (Urutan: 1 2 4 5 6 9)

## Modul 9 merge sort dan quick sort

- a. Urutkan data dibawah ini dengan algoritma merge sort dan quick sort jelaskan pula Langkah-langkahnya
- b. 9 1 2 5 6 4

Jawab:

Langkah-langkah Merge Sort:

1. Jika panjang array  $\leq 1$ , anggap sudah terurut.
2. Bagi array menjadi dua bagian.
3. Lakukan rekursi untuk kedua bagian tersebut.
4. Gabungkan dua bagian yang sudah diurutkan menjadi satu array terurut.

Urutan Data:

Langkah 1: [9 1] [2 5] [6 4] (Pembagian)

Langkah 2: 1 9 | 2 5 | 4 6 (Rekursi)

Langkah 3: 1 2 4 5 6 9 (Gabungan)

Langkah-langkah Quick Sort:

1. Pilih elemen pivot dari array.
2. Partisi array sehingga elemen-elemen yang lebih kecil dari pivot berada di sebelah kiri, dan elemen-elemen yang lebih besar berada di sebelah kanan.
3. Lakukan rekursi untuk kedua partisi tersebut.
4. Gabungkan hasil rekursi.

Urutan Data:

Langkah 1: Pivot = 4. [1 2] 4 [9 5 6] (Partisi)

Langkah 2: Pivot = 2. 1 [2] 4 [9 5 6] (Partisi)

Langkah 3: Pivot = 9. 1 2 4 | 5 [6 9] (Partisi)

Langkah 4: Pivot = 6. 1 2 4 | 5 [6] 9 (Partisi)

Langkah 5: 1 2 4 5 6 9 (Gabungan)

## Modul 10 Algoritma pencarian

1. Dengan pencarian sekuensial, carilah data 9 dari data berikut

10 1 2 11 8 7 5 4 15 16

2. Dengan pencarian biner, carilah data 9 dari data berikut

2 5 9 10 12 15 18 20 22

Jawab:

Langkah-langkah pencarian sekuensial:

1. Mulai dari elemen pertama.
2. Bandingkan setiap elemen dengan nilai yang dicari (9).
3. Jika nilai ditemukan, kembalikan indeksnya.
4. Jika mencapai akhir array tanpa menemukan nilai, kembalikan -1.

Urutan Data:

10 1 2 11 8 7 5 4 15 16

Proses Pencarian:

1. Bandingkan 10  $\neq$  9
2. Bandingkan 1  $\neq$  9
3. Bandingkan 2  $\neq$  9
4. Bandingkan 11  $\neq$  9
5. Bandingkan 8  $\neq$  9
6. Bandingkan 7  $\neq$  9
7. Bandingkan 5  $\neq$  9
8. Bandingkan 4  $\neq$  9
9. Bandingkan 15  $\neq$  9
10. Bandingkan 16  $\neq$  9

Nilai 9 tidak ditemukan. Hasil pencarian sekuensial: -1.



#### Langkah-langkah pencarian biner:

1. Tentukan batas kiri dan batas kanan array.
2. Hitung indeks tengah.
3. Bandingkan nilai tengah dengan nilai yang dicari.
4. Sesuaikan batas kiri atau kanan berdasarkan hasil perbandingan.
5. Ulangi langkah 2-4 hingga nilai ditemukan atau batas kiri lebih besar dari batas kanan.

#### Urutan Data:

2 5 9 10 12 15 18 20 22

#### Proses Pencarian:

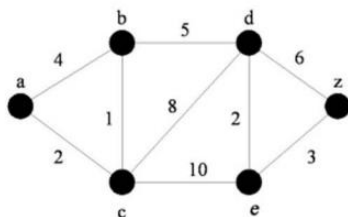
1. Batas kiri = 1, Batas kanan = 9
2. Indeks tengah =  $(1 + 9) / 2 = 5$
3. Bandingkan 15  $\neq$  9 (karena  $15 > 9$ )
4. Batas kiri = 1, Batas kanan = 4
5. Indeks tengah =  $(1 + 4) / 2 = 2$
6. Bandingkan 5  $\neq$  9 (karena  $5 < 9$ )
7. Batas kiri = 3, Batas kanan = 4
8. Indeks tengah =  $(3 + 4) / 2 = 3$
9. Bandingkan 10  $\neq$  9 (karena  $10 > 9$ )
10. Batas kiri = 3, Batas kanan = 2

Hasil Pencarian: Nilai 9 ditemukan pada indeks 3. Hasil pencarian biner: 3.

## Modul 11 Graph

Berdasarkan *graph* di bawah ini, representasikan matriks, gunakan algoritma warshall untuk mencari rute terpendek dan rute seperti pada Latihan 1.

Pada *graph* di bawah ini, selesaikan dengan algoritma dijkstra. Jalankan program dan analisa hasilnya.



## Modul 12 tree

1. Buatlah binary search tree, dengan data

7 5 12 3 6 1 4 9

Lakukan metode Traversal dengan algoritma:

- InOrder
- PreOrder
- PostOrder

Jawab:

InOrder Traversal:

1. Kunjungi simpul kiri.
2. Kunjungi akar.
3. Kunjungi simpul kanan.

Hasil InOrder Traversal: 1 3 4 5 6 7 9 12

PreOrder Traversal:

1. Kunjungi akar.
2. Kunjungi simpul kiri.
3. Kunjungi simpul kanan.

Hasil PreOrder Traversal: 7 5 3 1 6 4 12 9

PostOrder Traversal:

1. Kunjungi simpul kiri.
2. Kunjungi simpul kanan.
3. Kunjungi akar.

Hasil PostOrder Traversal: 4 1 3 6 5 9 12 7

