

KONSEP STRUKTUR DATA DAN IMPLEMENTASINYA DALAM BAHASA PEMROGRAMAN PYTHON, JAVA DAN C++



Fajar Ratnawati
Danuri
Jaroji



**KONSEP STRUKTUR DATA DAN
IMPLEMENTASINYA DALAM BAHASA
PEMROGRAMAN PYTHON, JAVA DAN
C++**

KONSEP STRUKTUR DATA DAN IMPLEMENTASINYA DALAM BAHASA PEMROGRAMAN PYTHON, JAVA DAN C++

Fajar Ratnawati
Danuri
Jaroji



KONSEP STRUKTUR DATA DAN IMPLEMENTASINYA DALAM BAHASA PEMROGRAMAN PYTHON, JAVA DAN C++

© Penerbit Perkumpulan Rumah Cemerlang Indonesia (PRCI)

Penulis:

**Fajar Ratnawati
Danuri
Jaroji**

Editor: Rusli

Cetakan Pertama: Desember 2022

Cover: Tim Kreatif PRCI

Tata Letak: Tim Kreatif PRCI

Hak Cipta 2022, pada Penulis. Diterbitkan pertama kali oleh:

**Perkumpulan Rumah Cemerlang Indonesia
ANGGOTA IKAPI JAWA BARAT**

Pondok Karisma Residence Jalan Raflesia VI D.151
Panglayungan, Cipedes Tasikmalaya – 085223186009

Website: www.rcipress.rcipublisher.org

E-mail: rumahcemerlangindonesia@gmail.com

Copyright © 2022 by Perkumpulan Rumah Cemerlang Indonesia
All Right Reserved

- Cet. I –: Perkumpulan Rumah Cemerlang Indonesia, 2022
Dimensi : 18,2 x 25,7 cm
ISBN: 978-623-448-423-6

Hak cipta dilindungi undang-undang
Dilarang memperbanyak buku ini dalam bentuk dan dengan
cara apapun tanpa izin tertulis dari penulis dan penerbit

Undang-undang No.19 Tahun 2002 Tentang
Hak Cipta Pasal 72

Undang-undang No.19 Tahun 2002 Tentang Hak Cipta
Pasal 72

Barang siapa dengan sengaja melanggar dan tanpa hak melakukan perbuatan sebagaimana dimaksud dalam pasal ayat (1) atau pasal 49 ayat (1) dan ayat (2) dipidana dengan pidana penjara masing-masing paling sedikit 1 (satu) bulan dan/atau denda paling sedikit Rp.1.000.000,00 (satu juta rupiah), atau pidana penjara paling lama 7 (tujuh) tahun dan/atau denda paling banyak Rp.5.000.000.000,00 (lima miliar rupiah).

Barang siapa dengan sengaja menyiarkan, memamerkan, mengedarkan, atau menjual kepada umum suatu ciptaan atau barang hasil pelanggaran hak cipta terkait sebagai dimaksud pada ayat (1) dipidana dengan pidana penjara paling lama 5 (lima) tahun dan/atau denda paling banyak Rp.500.000.000,00 (lima ratus juta rupiah).

KATA PENGANTAR

Puji syukur kehadiran Allah SWT dengan ridhoNya penulis dapat menyelesaikan buku ajar berupa buku yang digunakan untuk matakuliah Struktur Data.

Buku ini disusun dengan sebaik-baiknya, namun demikian tidak menutup kemungkinan terdapat kekurangan, sehingga masukan dibutuhkan demi kesempurnaan buku ini.

Kami mengucapkan terimakasih ke berbagai pihak yang telah membantu menyelesaikan buku ini terutama rekan-rekan dosen di Jurusan Teknik Informatika. Semoga buku ini bermanfaat untuk kita semua terutama bagi peserta didik.

Bengkalis, Desember 2022

Penulis

DAFTAR ISI

KATA PENGANTAR	i
DAFTAR ISI	ii
BAB I STRUKTUR DATA STATIS	1
A. Dasar Struktur Data.....	1
B. Array	3
Array dalam bahasa python	3
Array dalam Bahasa java	5
Array dalam Bahasa C++	8
C. Array Multidimensi.....	12
Array multidimensi dalam Bahasa python	12
Array multidimensi dalam Bahasa java.....	14
Array multidimensi dalam Bahasa C++	21
Evaluasi.....	23
BAB II STRUKTUR DATA DINAMIS	25
A. Linked List.....	27
Linked List dalam Bahasa python.....	27
Linked List dalam Bahasa java.....	37
Linked List dalam Bahasa C++	40
B. Stack.....	43
Stack dalam bahasa python	44
Stack dalam Bahasa java.....	47
Stack dalam Bahasa C++	53
C. Queue	53
Queue dalam Bahasa python.....	54
Queue dalam Bahasa java	57
Queue dalam Bahasa C++	59
D. Tree.....	60
Tree dalam Bahasa python	62
Tree dalam Bahasa java	67

Tree dalam Bahasa C++	68
Evaluasi	76
BAB III SORTING	79
A. Buble Sort.....	80
Buble Sort dengan bahasa pemrograman python.....	80
Buble Sort dengan bahasa pemrograman Java.....	81
Buble Sort dengan bahasa pemrograman C++	82
B. Selection Sort	83
Selection Sort dengan bahasa pemrograman Python.....	83
Selection Sort dengan bahasa pemrograman Java	84
Selection Sort dengan bahasa pemrograman C++	85
C. Insertion Sort	86
Insertion Sort dengan Bahasa pemrograman python.....	87
Insertion Sort dengan Bahasa pemrograman Java.....	87
Insertion Sort dengan Bahasa pemrograman C++	88
D. Merge Sort.....	89
Merge Sort dengan Bahasa pemrograman Python	90
Merge Sort dengan Bahasa pemrograman Java	91
Merge Sort dengan Bahasa pemrograman C++	94
Evaluasi	96
BAB IV SEARCHING	97
A. Sequential Search	97
Sequential search dalam Bahasa pyhton.....	98
Sequential search dalam Bahasa java.....	98
Sequential search dalam Bahasa C++.....	99
B. Binary Search	100
Binary search dalam Bahasa python	101
Binary search dalam Bahasa java	101
Binary search dalam Bahasa C++	103
C. Interpolation Search.....	104
Interpolation Search dalam Bahasa pyhton	104

Interpolation Search dalam Bahasa java	105
Interpolation Search dalam Bahasa C++	106
D. Fibonacci Search.....	107
Fibonacci search dalam Bahasa python.....	108
Fibonacci search dalam Bahasa java.....	109
Evaluasi.....	112
DAFTAR PUSTAKA.....	114
BIOGRAFI PENULIS	115

BAB I

STRUKTUR DATA STATIS

A. Dasar Struktur Data

Struktur data adalah cara menyimpan, mengorganisasikan, dan mengatur data dalam media penyimpanan komputer sehingga data tersebut dapat digunakan secara efisien baik dari segi waktu komputasi maupun memori. Data merupakan representasi dari fakta dunia nyata. Sedangkan fakta merupakan keterangan tentang kenyataan yang disimpan, direkam atau direpresentasikan dalam bentuk tulisan, suara, gambar, sinyal atau symbol.

Struktur data statis adalah organisasi atau kumpulan data dalam memori yang ukurannya tetap. Oleh sebab itu perlu didefinisikan ukuran maksimum dari data, karena memori tidak dapat dialokasikan Kembali dilain waktu. Dalam struktur data statis, memori dialokasikan pada saat compile dan pengguna tidak dapat mengubah ukuran setelahnya. Tetapi data yang disimpan didalamnya dapat diubah.

Ukuran yang bersifat tetap memberikan banyak manfaat sekaligus kelemahan. Dengan alokasi memori tetap, tidak perlu khawatir dengan adanya masalah overflow dan underflow saat memasukkan atau menghapus elemen dalam struktur data statis. Kekurangannya adalah struktur data statis menghabiskan banyak memori dan tidak hemat ruang. Karena memori disediakan pada waktu kompilasi, semua memori yang dialokasikan bersebelahan sehingga tidak perlu khawatir mengenai alamat memori semua elemen data. Kita hanya perlu menyimpan alamat elemen pertama dan lokasi elemen lainnya dapat dengan mudah ditemukan. Adapun salah satu contoh dari struktur data statis adalah array.

Karakteristik struktur data statis

Berikut adalah karakteristik dari struktur data statis:

1. Ukuran tipe data tetap
2. Memori statis dialokasikan pada waktu compile oleh kompiler
3. Data disimpan dalam memori stack program.

4. Kurang efisien dibandingkan struktur data dinamis.
5. Tidak ada penggunaan kembali memori.
6. Tidak akan terjadi overflow.

Kelebihan struktur data statis

Berikut adalah keunggulan atau kelebihan dari struktur data statis:

1. Struktur data statis mudah ditangani karena kompiler menangani semua proses alokasi dan dealokasi.
2. Memori dialokasikan dalam bentuk yang berdekatan sehingga tidak perlu mempertahankan struktur struktur data atau variabel eksplisit lainnya untuk menyimpan lokasi memori.
3. Karena ukuran tetap, pengguna tidak perlu khawatir tentang kondisi overflow atau underflow saat memasukkan atau menghapus elemen apa pun ke struktur data masing-masing.
4. Sangat mudah untuk diprogram dan struktur data seperti array menyediakan akses acak.

Kekurangan struktur data statis

Struktur data statis sangat berguna untuk programmer tetapi setiap hal yang baik memiliki beberapa kekurangan juga. Berikut adalah beberapa kelemahan dari struktur data statis:

1. Pengguna harus memperkirakan ruang maksimum yang diperlukan untuk struktur data statis yang mungkin lebih banyak dibandingkan dengan yang sebenarnya dibutuhkan. Hal ini dapat mengakibatkan banyak memori menjadi hilang.
2. Penyisipan elemen baru di antara dua elemen yang ada dalam struktur data statis hanya dimungkinkan jika ada ruang kosong di antara keduanya, jika tidak, penyisipan akan memakan banyak waktu.
3. Penghapusan elemen dapat menciptakan ruang kosong di antara dua elemen dan menutupi ruang itu banyak memakan waktu.

B. Array

Array merupakan tipe terstruktur yang mempunyai komponen dalam jumlah yang tetap dan dengan tipe data yang sama. Array juga merupakan variabel yang bisa menyimpan banyak data dalam satu variabel. Array menggunakan indeks untuk memudahkan akses terhadap data yang disimpannya. Indeks array selalu dimulai dari "0".

Array dalam bahasa python

Sebelum memulai percobaan, siapkan terlebih dahulu sebuah project di IDE yang digunakan dengan nama 'Project1-StrukturDataStatis' sesuai dengan langkah-langkah pembuatan project baru. Terdapat dua bagian di project 1 yaitu Array, dan Array Multidimensi.

Percobaan 1 : Mengenal array

Tulis script berikut pada file baru dengan nama 'P1-array':

```
#Array 1 Dimensi
sisi = 8
siswa = ["Budi", "Charlie", "Erlangga"]
tas = ["eiger","bodypack",5, 12.5]
print(sisi)
print(siswa[0])
print(siswa[1])
print(siswa[2])
print(tas[0])
print(tas[3])
```

Percobaan 2 : Menampilkan nilai elemen array

Masih di file yang sama ('P1-array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
#Menampilkan semua data/nilai array
#Perintah : for..in
for data in siswa:
    print(data)
```

Percobaan 3 : Menghitung panjang array

Masih di file yang sama ('P1-array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
#Menghitung panjang array
#Perintah : len
panjang = len(siswa)
print(panjang)
```

Percobaan 4 : Mengubah nilai elemen array

Masih di file yang sama ('P1-array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
#Mengubah nilai elemen array
siswa[0] = "Suci Sekar Sari"
print(siswa)
siswa[2] = "Cahyo Prawiro"
print(siswa)
```

Percobaan 5 : Menambah elemen array

Masih di file yang sama ('P1-array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
#Menambah elemen array
#Perintah : append
siswa.append("Deski Ari Sandi")
print(siswa)
siswa.append("Ikmal")
print(siswa)
siswa.append("Charlie")
print(siswa)
siswa[5]="Charles"
print(siswa)
```

Percobaan 5 : Menyisip elemen array

Masih di file yang sama ('P1-array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
#Menyisip elemen array  
#perintah : insert  
siswa.insert(1, "Rukhi")  
print(siswa)
```

Percobaan 6 : Menghapus elemen array

Masih di file yang sama ('P1-array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
#Menghapus elemen array  
#Perintah 1: pop, menghapus berdasarkan index  
siswa.pop(5)  
print(siswa)  
#Perintah 2: remove, menghapus berdasarkan nilai/value  
siswa.remove("Charlie")  
print(siswa)
```

Array dalam Bahasa java

Siapkan sebuah class baru dengan nama 'P1Array' pada 'Project1- StrukturDataStatis'. Kemudian lakukan beberapa percobaan berikut ini.

Percobaan 1 : Mengenal array

Tulis script berikut pada file (P1-array):

```
//Array 1 Dimensi
package Project1-StrukturDataStatis;

public class P1Array {
    public static void main(String[] args) {

        String[]siswa={"Budi","Charlie","Erlangggga"};
        int[]sisi={8};
        String[]tas={"eiger","bodypack","5","12.5"};
        System.out.println(sisi[0]);
        System.out.println(siswa[0]);
        System.out.println(siswa[1]);
        System.out.println(siswa[2]);
        System.out.println(tas[0]);
        System.out.println(tas[3]);
    }
}
```

Percobaan 2 : Menampilkan nilai elemen array menggunakan "for"

Masih di class yang sama ('P1Array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
//Menampilkan semua data/nilai array
//Menggunakan Perulangan : for

package Project1-StrukturDataStatis;

public class P1Array {
    public static void main(String[] args) {

        String[]siswa={"Budi","Charlie","Erlangggga"};

        for(int i=0;i<siswa.length;i++){

            System.out.println("indeks ke-"+i+":
            "+siswa[i]);
        }
    }
}
```

Percobaan 3 : Menampilkan nilai elemen array menggunakan “for-each”

Masih di class yang sama ('P1Array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
//Menampilkan semua data/nilai array
//Menggunakan Perulangan : for-each

package Project1-StrukturDataStatis;

public class P1Array {
    public static void main(String[] args) {

        String[]siswa={"Budi","Charlie","Erlangga"};

        for(String i:siswa){

            System.out.println(i);

        }

    }
}
```

Percobaan 4 : Menghitung panjang array

Masih di class yang sama ('P1Array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
//Menghitung Panjang array
//Menggunakan Perintah : length

package Project1-StrukturDataStatis;

public class P1Array {
    public static void main(String[] args) {

        String[]siswa={"Budi","Charlie","Erlangga"};
        int[]sisi={8};
        String[]tas={"eiger","bodypack","5","12.5"};

        System.out.println(siswa.length);
        System.out.println(sisi.length);
        System.out.println(tas.length);

    }
}
```

Percobaan 5 : Mengubah nilai elemen array

Masih di class yang sama ('P1Array') pada 'Project1- StrukturDataStatis', kemudian tambahkan script berikut:

```
//Mengubah nilai array

package Project1-StrukturDataStatis;

public class P1Array {
    public static void main(String[] args) {

        String[]siswa={"Budi","Charlie","Erlangga"};
        Siswa[1]="Andi";
        System.out.println(siswa[1]);

    }
}
```

Array dalam Bahasa C++

Percobaan 1 : Mengenal array

```
// Array 1 dimensi
#include <iostream>
#include <string>
using namespace std;

int main() {
    //deklarasi array
    int sisi[]={8};
    string siswa[] = {"Budi", "Charlie", "Erlangga"};
    string tas[]{"eiger","bodypack","5","12.5"};
    //akses nilai pada array
    cout << sisi[0] <<endl;
    cout << siswa[0] <<endl;
    cout << siswa[1] <<endl;
    cout << siswa[2] <<endl;
    cout << tas[0] <<endl;
    cout << tas[3] <<endl;
    return 0;
}
```

Percobaan 2 : Menampilkan nilai elemen array

```
// Menampilkan semua data/nilai array
// Menggunakan perulangan for:each (untuk C++ vers 11)
#include <iostream>
#include <string>
using namespace std;

int main() {
    //deklarasi array
    string siswa[] = {"Budi", "Charlie", "Erlangga"};
    for(string s : siswa){
        cout << s <<"\n";
    }
    return 0;
}
```

Percobaan 3 : Menghitung panjang array

```
//Menghitung Panjang array
//Menggunakan Perintah : sizeof()
```

```

#include <iostream>
#include <string>
using namespace std;

int main() {
    //deklarasi dan inisialisasi array
    int sisi[]={8};
    string siswa[] = {"Budi", "Charlie", "Erlangga"};
    string tas={"eiger","bodypack","5","12.5"};
    //mengetahui ukuran/panjang array
    //sizeof() memberikan nilai balik byte
    //sehingga harus dibagi dengan ukuran type datanya
    //untuk mendapatkan panjang elemen array
    cout << sizeof(sisi)/sizeof(int) <<endl;
    cout << sizeof(siswa)/sizeof(string) <<endl;
    cout << sizeof(tas)/sizeof(string) <<endl;
    return 0;
}

```

Percobaan 4 : Mengubah nilai elemen array

```

//Mengubah nilai array
#include <iostream>
#include <string>
using namespace std;

int main() {
    //deklarasi dan inisialisasi array
    string siswa[] = {"Budi", "Charlie", "Erlangga"};
    //mengubah nilai array index ke 0
    siswa[0] = "Hamzah Al Fatih";
    //mengubah nilai array index ke 2
    siswa[2] = "Andi";
    for(string s : siswa){
        cout<<s<<"\n";
    }
    return 0;
}

```

Percobaan 5 : Menambah elemen array

Penambahan array di C++ tidak tersedia, namun sebagai alternatif dapat menggunakan vector, berikut penerapan codenya.

```
//menambahkan elemen pada vector (alternatif array)
#include <iostream>
#include <vector>
using namespace std;

int main() {
    //deklarasi dan inisialisasi array
    vector<string> siswa = {"Budi", "Charlie", "Erlangga"};
    //menambahkan elemen data ke vector
    siswa.push_back("Hamzah Al Fatih");
    siswa.push_back("Andi");
    for(string s : siswa){
        cout<<s<<"\n";
    }
    return 0;
}
```

Percobaan 5 : Menyisip elemen array

Pada saat buku ini disusun, fungsi insert untuk menambahkan elemen di array di C belum tersedia, sebagai alternatif dapat menggunakan fungsi insert di vector. Penerapan codenya sebagai berikut:

```
//menyisipkan elemen pada vector (alternatif array)
#include <iostream>
#include <vector>
using namespace std;

int main() {
    //deklarasi dan inisialisasi array
    vector<string> siswa = {"Budi", "Charlie", "Erlangga"};
    //menyisipkan elemen ke index 2
    siswa.insert(siswa.begin()+2,"Hamzah Al Fatih");
    for(string s : siswa){
        cout<<s<<"\n";
    }
    return 0;
}
```