

# **LAPORAN PRAKTIKUM**

## **MODUL I TIPE DATA**



**Disusun oleh:  
Nadhif Atha Zaki  
NIM: 2311102007**

**Dosen Pengampu:**  
Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
PURWOKERTO  
2023**

# **BAB I**

## **TUJUAN PRAKTIKUM**

1. Mahasiswa dapat mempelajari tipe data primitif, abstrak, dan kolektif.
2. Mahasiswa dapat memahami pengaplikasian pada tools yang digunakan.
3. Mahasiswa mengaplikasikan berbagai tipe data pada bahasa pemograman yang telah ditentukan.

## **BAB II**

### **DASAR TEORI**

Tipe data adalah adalah sebuah pengklasifikasian data berdasarkan jenis data tersebut. Tipe data dibutuhkan agar kompiler dapat mengetahui bagaimana sebuah data akan digunakan. Adapun tipe data yang akan dipelajari, sebagai berikut :

1. Tipe data Primitif
2. Tipe data Abstrak
3. Tipe data Koleksi

#### **Tipe data Primitif**

Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem, tipe data primitif ini disediakan oleh banyak bahasa pemrograman, perbedaannya terletak pada

jumlah bit yang dialokasikan untuk setiap bit pada tipe data primitif tergantung pada bahasa pemrograman, compiler dan sistem operasinya. Contoh tipe data primitif adalah :

- a. Int : adalah tipe data yang digunakan untuk menyimpan bilangan bulat seperti 12, 1, 4, dan sebagainya.
- b. Float : tipe data yang digunakan untuk menyimpan bilangan desimal seperti 1.5, 2.1, 3.14, dan sebagainya.
- c. Char : berfungsi untuk menyimpan data berupa sebuah huruf. Biasanya digunakan untuk simbol seperti A, B, C dan seterusnya
- d. Boolean : tipe data ini digunakan untuk menyimpan nilai boolean yang hanya memiliki dua nilai yaitu true dan false.

#### **Tipe Data Abstrak**

Tipe data abstrak atau yang biasa disebut Abstrak Data Tipe(ADT) merupakan tipe data yang dibentuk oleh programer itu sendiri. Pada tipe data abstrak bisa berisi banyak tipe data, jadi nilainya bisa lebih dari satu dan beragam tipe data. Fitur Class

adalah fitur Object Oriented Program(OOP) pada bahasa C++ yang mirip dengan fitur

data structures Struct pada bahasa C. Keduanya berfungsi untuk membungkus tipe data di dalamnya sebagai anggota. menurut [learn.microsoft.com](https://learn.microsoft.com/en-us/cpp/faq) perbedaan antara  
Praktikum Struktur Data dan Algoritma 2

Struct dan Class adalah pada akses defaultnya dimana Struct bersifat public dan Class

bersifat private.

Tipe Data Koleksi

Tipe data koleksi (Collection Data Type) adalah tipe data yang digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan.  
Tipe

data koleksi memungkinkan Anda menyimpan, mengelola, dan mengakses sejumlah besar data dengan cara yang terstruktur. Ada beberapa tipe data koleksi yang umum digunakan dalam pemrograman, dan di antaranya adalah:

a. Array : Array adalah struktur data statis yang menyimpan elemen-elemen dengan tipe data yang sama. Elemen-elemen tersebut dapat diakses dengan menggunakan indeks. Array memiliki ukuran tetap yang ditentukan saat deklarasi.

b. Vector : Vector adalah Standard Template Library (STL) jika di dalam C/C++ memiliki bentuk `std::vector` . Umumnya, vector mirip seperti array yang memiliki kemampuan untuk menyimpan data dalam bentuk elemen-elemen yang alokasi memorinya dilakukan otomatis dan bersebelahan. Kemampuan vector bukan hanya pada jumlah elemen yang dinamis, vector pada C/C++ juga dilengkapi dengan fitur-fitur pelengkap seperti element access, iterators, capacity, modifiers

c. Map : Map terasa mirip dengan array namun dengan index yang memungkinkan untuk berupa tipe data selain integer. Pada map, indeks tersebut diberi nama "key". Pada `std::map` digunakan Self-Balancing Tree khususnya Red-Black Tree.

## BAB III

### GUIDED

#### 1. Guided 1

##### Source code

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

int main(){
    char op;
    float num1, num2;

    cin >> op;

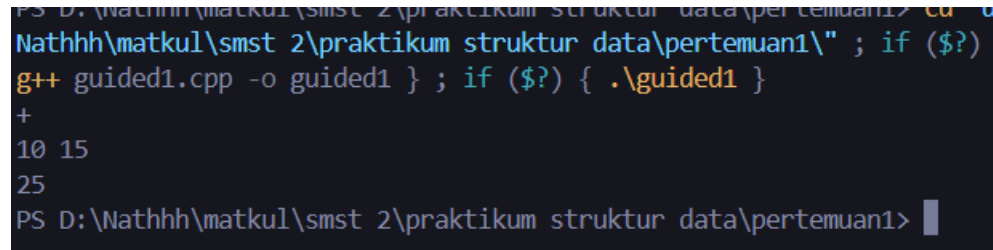
    cin >> num1 >> num2;

    switch (op)
    {
        case '+':
            cout << num1 + num2;
            break;
        case '-':
            cout << num1 - num2;
            break;
        case '*':
            cout << num1 * num2;
            break;
        case '/':
            cout << num1 / num2;
            break;

        default:
            cout << "error! operator is not correct";
    }
}
```

```
    return 0;
}
```

### Screenshoot program:



```
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> g++ guided1.cpp -o guided1 ; if ($?) { .\guided1 }
+
10 15
25
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>
```

### Deskripsi program:

Program ini adalah kalkulator sederhana yang meminta pengguna untuk memasukkan operator matematika dan dua bilangan desimal. Setelah menerima input, program menggunakan struktur kontrol switch untuk memilih operasi yang sesuai berdasarkan operator yang dimasukkan. Operasi yang didukung melibatkan penjumlahan (+), pengurangan (-), perkalian (\*), dan pembagian (/). Hasil perhitungan kemudian ditampilkan ke layar, atau jika operator tidak valid, program memberikan pesan kesalahan. Program ini memberikan penggunaan dasar dari kontrol alur dan operasi matematika dalam bahasa C++, cocok untuk kebutuhan kalkulator sederhana.

## 2. Guided 2

### Source code

```
#include <stdio.h>

struct Mahasiswa
{
    const char *name;
    const char *address;
```

```
    int age;
};

int main()
{
    struct Mahasiswa mhs1, mhs2;

    mhs1.name = "Dian";
    mhs1.address = "Mataram";
    mhs1.age = 22;
    mhs2.name = "Bambang";
    mhs2.address = "Surabaya";
    mhs2.age = 23;

    printf("## Mahasiswa 1 ##\n");
    printf("nama: %s\n", mhs1.name);
    printf("alamat: %s\n", mhs1.address);
    printf("umur: %d\n", mhs1.age);
    printf("## Mahasiswa 2 ##\n");
    printf("nama: %s\n", mhs2.name);
    printf("alamat: %s\n", mhs2.address);
    printf("umur: %d\n", mhs2.age);
    return 0;
}
```

**Screenshoot program:**

```

PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> cd
Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1" ; if ($?
g++ guided2.cpp -o guided2 } ; if ($?) { .\guided2 }
## Mahasiswa 1 ##
nama: Dian
alamat: Mataram
umur: 22
## Mahasiswa 2 ##
nama: Bambang
alamat: Surabaya
umur: 23
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>

```

### Deskripsi program:

Program ini menggunakan bahasa pemrograman C dan mendemonstrasikan penggunaan struktur (struct) untuk merepresentasikan data mahasiswa. Dua variabel struct, mhs1 dan mhs2, dideklarasikan untuk menyimpan informasi dua mahasiswa. Setiap mahasiswa memiliki atribut seperti nama, alamat, dan umur. Nilai-nilai atribut tersebut diisi secara manual untuk setiap mahasiswa. Program kemudian mencetak informasi masing-masing mahasiswa ke layar menggunakan perintah printf. Hasilnya adalah tampilan yang memuat detail dari kedua mahasiswa, termasuk nama, alamat, dan umur mereka. Program ini memberikan contoh penggunaan struktur dalam bahasa C untuk merepresentasikan data terstruktur dengan cara yang lebih terorganisir.

### 3. Guided 3

Source Code:

```

#include <iostream>
using namespace std;
int main(){
    //deklarasi dan inisialisasi array

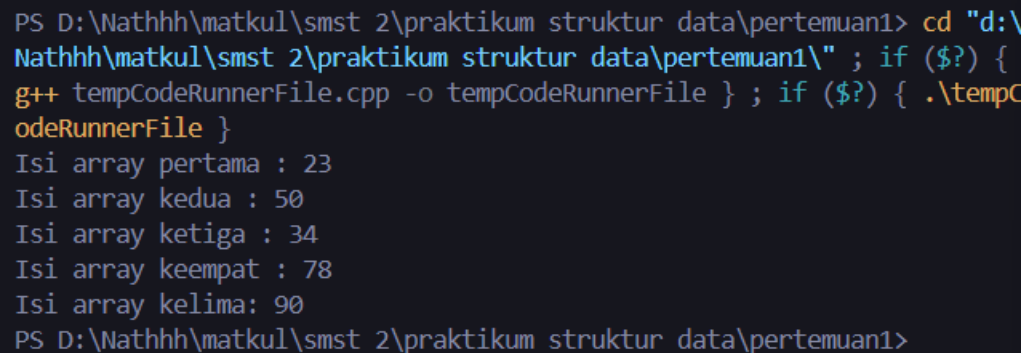
```



```
int nilai[5];
nilai[0] = 23;
nilai[1] = 50;
nilai[2] = 34;
nilai[3] = 78;
nilai[4] = 90;

cout << "Isi array pertama : "<< nilai[0] << endl;
cout << "Isi array kedua : "<< nilai[1] << endl;
cout << "Isi array ketiga : "<< nilai[2] << endl;
cout << "Isi array keempat : "<< nilai[3] << endl;
cout << "Isi array kelima: "<< nilai[4] << endl;
return 0;
}
```

#### Screenshoot program:



```
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> cd "d:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Isi array pertama : 23
Isi array kedua : 50
Isi array ketiga : 34
Isi array keempat : 78
Isi array kelima: 90
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>
```

#### Deskripsi program:

Program ini adalah contoh penggunaan array dalam bahasa C++. Pertama, dilakukan deklarasi dan inisialisasi array dengan nama nilai yang dapat menyimpan lima elemen bertipe integer. Kemudian, setiap elemen array diisi dengan nilai yang berbeda. Selanjutnya, program menampilkan isi masing-masing elemen array ke layar menggunakan perintah cout. Hasilnya adalah mencetak nilai-nilai yang telah diisi ke dalam array ke layar dengan pesan yang menyertakan indeks dan nilai masing-masing

elemen. Program ini memberikan pemahaman tentang cara menggunakan dan mengakses elemen array dalam bahasa C++.

## LATIHAN KELAS - UNGUIDED

1.

Buatlah program menggunakan tipe data primitif minimal dua fungsi dan bebas. Menampilkan program, jelaskan program tersebut dan ambil kesimpulan dari materi tipe data primitif!

### Source code

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

int tambah_2311102007(int a, int b) {
    return a + b;
}

int kali(int a, int b) {
    return a * b;
}

int main() {

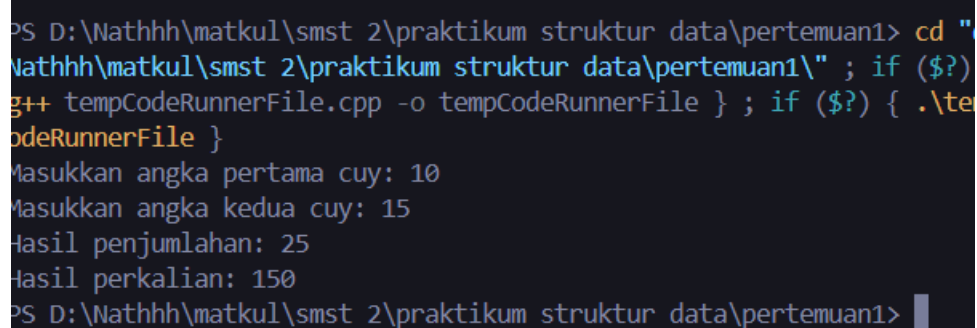
    int angka1, angka2;
    cout << "Masukkan angka pertama cuy: ";
    cin >> angka1;
    cout << "Masukkan angka kedua cuy: ";
    cin >> angka2;

    int hasil_tambah = tambah_2311102007(angka1, angka2);
    int hasil_kali = kali(angka1, angka2);

    cout << "Hasil penjumlahan: " << hasil_tambah << endl;
    cout << "Hasil perkalian: " << hasil_kali << endl;
```

```
    return 0;
}
```

### Screenshoot program



```
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> cd "D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan angka pertama cuy: 10
Masukkan angka kedua cuy: 15
Hasil penjumlahan: 25
Hasil perkalian: 150
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>
```

### Deskripsi program:

Program c++ dengan menggunakan 2 fungsi untuk menjumlahkan dan mengalikan 2 bilangan yang diinputkan

### Penjelasan Program:

1. Program ini menggunakan tipe data primitif int untuk merepresentasikan bilangan bulat.
2. Dua fungsi didefinisikan, yaitu tambah dan kali, yang masing-masing menerima dua parameter bertipe int dan mengembalikan hasil penjumlahan dan perkalian.
3. Pengguna diminta untuk memasukkan dua angka menggunakan cin.
4. Fungsi tambah dipanggil dengan dua angka yang dimasukkan, dan hasilnya disimpan dalam variabel hasil\_tambah.
5. Fungsi kali dipanggil dengan dua angka yang dimasukkan, dan hasilnya disimpan dalam variabel hasil\_kali.
6. Hasil penjumlahan dan perkalian kemudian ditampilkan ke layar menggunakan cout.

Kesimpulan dari Materi Tipe Data Primitif: Program ini menggunakan tipe data primitif int untuk merepresentasikan bilangan bulat. Tipe data primitif seperti int digunakan untuk menyimpan nilai dengan ukuran yang tetap, sehingga memungkinkan efisiensi dalam penggunaan memori. Kesimpulannya, pemahaman dan penggunaan tipe data primitif sangat penting dalam pengembangan program karena memungkinkan programmer untuk secara tepat dan efisien menangani berbagai jenis data sesuai kebutuhan program.

## 2. Unguided 2

### 2. Jelaskan fungsi dari class dan struct secara detail dan berikan contoh programnya

class adalah konstruksi yang digunakan untuk mendefinisikan objek dan perilaku mereka. Class merupakan blueprint atau cetak biru untuk objek, yang mendefinisikan atribut (data) dan metode (fungsi) yang dimiliki oleh objek tersebut. Dengan menggunakan class, programmer dapat membuat objek baru berdasarkan blueprint yang sama dengan cara yang efisien.

Contoh class:

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

class Mahasiswa {
public:

    string nama;
    int umur;

    void displayInfo() {
        cout << "Nama: " << nama << endl;
        cout << "Umur: " << umur << " tahun" << endl;
    }
}
```

```
};

int main() {

    Mahasiswa mahasiswa1;

    mahasiswa1.nama = "Nadhif";
    mahasiswa1.umur = 19;

    mahasiswa1.displayInfo();

    return 0;
}
```

### Output:

```
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> c
Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1\" ; if (
g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .
odeRunnerFile }
Nama: Nadhif
Umur: 19 tahun
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>
```

Sedangkan struct mirip dengan class dalam hal mendefinisikan struktur data, tetapi memiliki perbedaan utama dalam sifat default visibilitasnya. Secara default, semua anggota struct bersifat public, sedangkan pada class, anggota bersifat private secara default. Struct biasanya digunakan untuk menyimpan data tanpa memiliki metode (fungsi) terkait.

### Contoh struct:

```
#include<iostream>

using namespace std;

struct Point {
```

```

    int x_2311102007;
    int y;
};

int main() {
    Point titik;

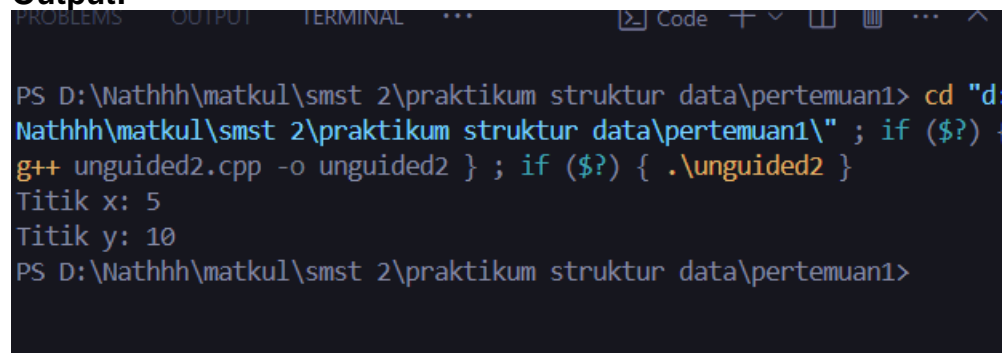
    titik.x_2311102007 = 5;
    titik.y = 10;

    cout << "Titik x: " << titik.x_2311102007 << endl;
    cout << "Titik y: " << titik.y << endl;

    return 0;
}

```

### Output:



```

PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> cd "d:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1"
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> g++ unguided2.cpp -o unguided2
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> .\unguided2
Titik x: 5
Titik y: 10
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>

```

### 3. Unguided 3

Buat dan jelaskan program menggunakan fungsi map dan jelaskan perbedaan dari array dengan map.

#### Source code:

```

#include <iostream>
#include <vector>
#include <map>

```

```

#include <algorithm>

using namespace std;

int main() {
    vector<int> array = {1, 2, 3, 4, 5};

    map<int, int> transformedMap;
    transform(array.begin(), array.end(),
inserter(transformedMap, transformedMap.begin()),
    [](int element) { return make_pair(element, element
* element); });

    cout << "Hasil transformasi dalam map:\n";
    for (const auto& pair : transformedMap) {
        cout << pair.first << " -> " << pair.second << "\n";
    }

    return 0;
}

```

### Output:

```

PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1> cd
Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1\ ; if ($?) {
g++ unguided3.cpp -o unguided3 } ; if ($?) { .\unguided3 }
Hasil transformasi dalam map:
1 -> 1
2 -> 4
3 -> 9
4 -> 16
5 -> 25
PS D:\Nathhh\matkul\smst 2\praktikum struktur data\pertemuan1>

```

### Deskripsi Program:



Program menggunakan C++ dan memanfaatkan library standar seperti iostream, vector, map, dan algorithm. Program ini mendefinisikan sebuah vektor array yang berisi lima bilangan bulat. Menggunakan fungsi transform dari library algorithm, setiap elemen vektor diubah menjadi pasangan kunci-nilai dengan kunci sebagai elemen asli dan nilai sebagai elemen dipangkatkan dua, dan hasilnya disimpan dalam map bernama transformedMap. Akhirnya, program menampilkan hasil transformasi tersebut dengan mencetak setiap pasangan kunci dan nilai dalam map. Program ini membuktikan kemampuan C++ dalam mengolah data menggunakan konsep vektor, map, dan transformasi.

### **Penjelasan Program:**

#### **1. Deklarasi Library dan Penggunaan namespace std:**

- Program dimulai dengan mendeklarasikan beberapa library standar C++, seperti iostream, vector, map, dan algorithm.
- Menggunakan using namespace std; untuk memudahkan penggunaan elemen-elemen dari namespace std tanpa menuliskan std:: sebelumnya.

#### **2. Inisialisasi dan Transformasi Array:**

- Membuat sebuah vektor bernama array yang berisi lima bilangan bulat (1, 2, 3, 4, dan 5).
- Menggunakan fungsi transform dari library algorithm untuk mentransformasi setiap elemen vektor array menjadi pasangan kunci-nilai dalam map yang disebut transformedMap.

#### **3. Fungsi Lambda Sebagai Operasi Transformasi:**

- Menggunakan fungsi lambda sebagai operasi transformasi yang mengubah setiap elemen vektor menjadi pasangan kunci-nilai.
- Setiap kunci adalah elemen asli, dan nilai adalah hasil dari elemen tersebut dipangkatkan dua.

#### **4. Menampilkan Hasil Transformasi:**

- Menggunakan loop for dan iterator untuk menelusuri setiap pasangan kunci-nilai dalam transformedMap.
- Mencetak setiap pasangan ke layar dengan format "kunci -> nilai".

## **BAB IV**

### **KESIMPULAN**

#### **1. Tipe Data Primitif:**

- Tipe data primitif adalah tipe data yang sudah ditentukan oleh sistem dan disediakan oleh bahasa pemrograman.
- Contoh tipe data primitif melibatkan bilangan bulat (**int**), bilangan desimal (**float**), karakter (**char**), dan boolean.
- Allokasi bit untuk setiap tipe data primitif dapat bervariasi tergantung pada bahasa pemrograman, compiler, dan sistem operasi.

#### **2. Tipe Data Abstrak (ADT):**

- Tipe data abstrak atau Abstrak Data Type (ADT) dibentuk oleh programmer dan dapat berisi berbagai tipe data dengan nilai yang lebih dari satu.
- Penggunaan kelas dalam Object-Oriented Programming (OOP), seperti dalam C++, mirip dengan struktur data struct pada bahasa C.
- Perbedaan utama antara struct dan class terletak pada akses default, dimana struct bersifat public dan class bersifat private.

#### **3. Tipe Data Koleksi:**

- Tipe data koleksi digunakan untuk mengelompokkan dan menyimpan beberapa nilai atau objek secara bersamaan.
- Contoh tipe data koleksi melibatkan array (struktur data statis), vector (dalam STL C++ untuk dinamis array), dan map (dengan indeks berupa "key" pada std::map).

#### **4. Array, Vector, dan Map:**

- Array adalah struktur data statis yang menyimpan elemen dengan tipe data yang sama dan memiliki ukuran tetap.

- Vector adalah struktur data dinamis yang menyimpan elemen dengan alokasi memorinya otomatis dan dilengkapi dengan berbagai fitur seperti iterators dan modifiers.
- Map adalah struktur data yang mirip dengan array, namun menggunakan indeks yang dapat berupa tipe data selain integer, dengan `std::map` menggunakan struktur data Self-Balancing Tree (Red-Black Tree).

Keseluruhan, pemahaman tipe data primitif, abstrak, dan koleksi memberikan dasar yang kuat dalam memahami cara data diorganisir dan diakses dalam pengembangan perangkat lunak.