

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN
ALGORITMA**

**MODUL I
TIPE DATA**



Disusun Oleh :

Raka Andriy Shevchenko
2311102054

Dosen

Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO**

A. Dasar Teori

Tipe Data adalah jenis data yang memiliki batasan tempat dan karakteristik sesuai dengan interpretasi data dan dapat diolah oleh komputer untuk memenuhi kebutuhan dalam pemrograman komputer. Tipe data berguna untuk mempresentasikan jenis dari suatu nilai yang terdapat dalam program.^[1]

Dalam pemrograman C++ ada beberapa tipe data, beberapa contohnya adalah integer, float, dan lain lain. Tetapi itu tadi hanyalah tipe data primitif, tipe data dibagi menjadi 3: yaitu, tipe data primitif, tipe data abstrak, dan juga tipe data koleksi.

1. Tipe data primitif

Tipe data primitif adalah sekumpulan tipe data dasar yang menjadi dasar pembuatan semua tipe data lainnya, sering kali didukung langsung oleh prosesor. Contoh tipe data primitif diantara lain:

- a. Int: tipe data ini digunakan untuk menyimpan bilangan bulat seperti 1, 2, 3, dan seterusnya;
- b. Float: tipe data ini digunakan untuk menyimpan bilangan pecahan seperti 2.5, 3.14, dan seterusnya;
- c. Char: tipe data ini digunakan untuk menyimpan karakter 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*.

2. Tipe data abstrak

Tipe data abstrak di C++ dapat didefinisikan menggunakan kata kunci "*class*" dan "*struct*". *Class* adalah tipe data yang dapat mengandung *member variables* dan *member functions*, serta dapat menghindari akses ke detail implementasi internasinya. Sedangkan *Struct* adalah tipe data yang hanya mengandung *member variables*, tidak memiliki *member functions*.^[2] Kedua konstruksi dapat mengandung data *members*, serta dapat menggunakan *inheritance*.

3. 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

Tipe data *array* digunakan untuk menyimpan beberapa elemen dengan jenis yang sama. Elemen dalam *array* dapat didefinisikan dengan menggunakan kurung siku []. Anggota atau isi dari *array* itu sendiri harus satu jenis tipe data, misalnya terdiri dari kumpulan angka bulat (integer), kumpulan karakter (char), maupun kumpulan angka pecahan (double).

b. Vector

Tipe data vector dalam bahasa pemrograman C++ adalah tipe data dinamis yang berisi kumpulan elemen dengan jenis yang sama. Vector adalah tipe data yang mirip dengan array, tetapi dengan jumlah elemen yang dinamis. Vector dapat bertambah dan berkurang sesuai kebutuhan, serta elemennya dapat diakses atau diubah.^[3]

c. Map

Tipe data map dalam bahasa pemrograman C++ adalah tipe data yang digunakan untuk menyimpan kumpulan elemen dengan *key* dan *value* yang berbeda. Map merupakan tipe data yang didefinisikan oleh pengguna sendiri. Map terasa mirip dengan *array* namun dengan indeks 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*.

B. Guided

a. Guided 1

Source Code:

```
#include <iostream>
using namespace std;
// Main program
int main()
{
    char op;
    float num1, num2;

    //Meminta pengguna untuk memasukkan operator: +, -, *, /
    cout << "Masukkan operator (+, -, *, /): ";
    // It allows user to enter operator i.e. +, -, *, /
    cin >> op;

    //Meminta pengguna untuk memasukkan operand
    cout << "Masukkan 2 angka: ";
    // It allow user to enter the operands
    cin >> num1 >> num2;

    // Switch statement begins
    switch (op)
    {
        // If user enter +
        case '+':
            cout << "Hasil penjumlahan: " << num1 + num2;
            break;
        // If user enter -
        case '-':
            cout << "Hasil pengurangan: " << num1 - num2;
            break;
        // If user enter *
        case '*':
            cout << "Hasil perkalian: " << num1 * num2;
            break;
        // If user enter /
        case '/':
            cout << "Hasil pembagian: " << num1 / num2;
            break;
        // If the operator is other than +, -, * or /,
```

```

    // error message will display
    default:
        cout << "Error! operator is not correct";
    } // switch statement ends
    return 0;
}

```

Output:

```

PS C:\kuliah\Struktur Data\pertemuan pertama> & 'c:\Users\RAKA\.vscode\extensions\m
icrosoft-MIEngine-Out-vu3c5nid.2jg' '--stderr=Mic
Masukkan operator (+, -, *, /): +
Masukkan 2 angka: 1 2
Hasil penjumlahan: 3
PS C:\kuliah\Struktur Data\pertemuan pertama>

```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko

NIM: 2311102054

Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk melaksanakan operator aritmatika berupa tambah (+), kurang (-), kali (x), bagi (/). Program ini dibuat menggunakan tipe data primitif berupa char dan float, char untuk bagian operatornya, dan float untuk bagian angka angkanya.

b. Guided 2

Source Code:

```

#include <stdio.h>

//Struct
struct Mahasiswa
{
    const char *name;
    const char *address;
    int age;
};

int main()
{
    // menggunakan struct
    struct Mahasiswa mhs1, mhs2;
    // mengisi nilai ke struct
    mhs1.name = "Dian";
}

```

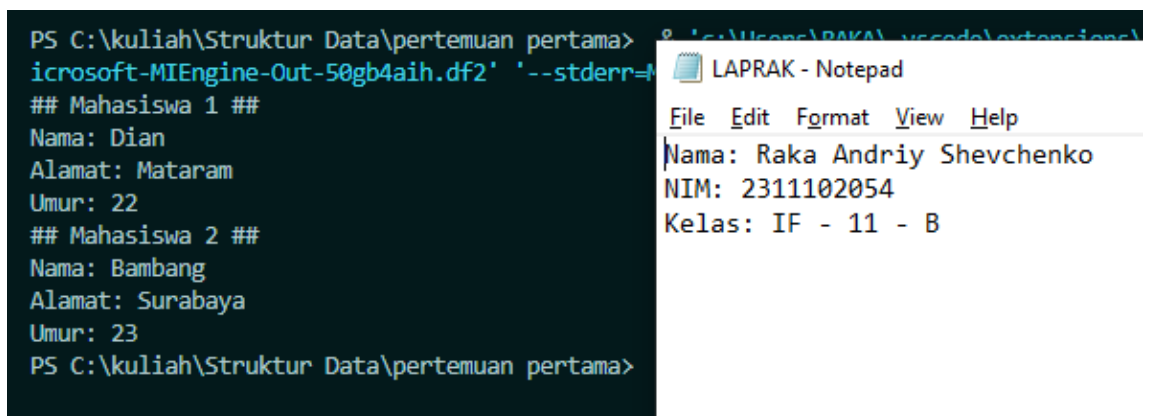
```

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

    // mencetak isi struct
    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;
}

```

Output:



```

PS C:\kuliah\Struktur Data\pertemuan pertama> g++ 'c:\Users\RAKA\vscode\extensions\
icrosoft-MIEngine-Out-50gb4aih.df2' '--stderr=1
## Mahasiswa 1 ##
Nama: Dian
Alamat: Mataram
Umur: 22
## Mahasiswa 2 ##
Nama: Bambang
Alamat: Surabaya
Umur: 23
PS C:\kuliah\Struktur Data\pertemuan pertama>

```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk mendata nama, alamat, dan umur mahasiswa, bisa hanya 1 ataupun lebih mahasiswa. Program ini dibuat menggunakan tipe data abstrak berupa *Struct*. Didalam *source code* sudah dijelaskan bagaimana cara penggunaan tipe data *struct*.

c. Guided 3

Source Code:

```

#include <iostream>
using namespace std;
int main()

```

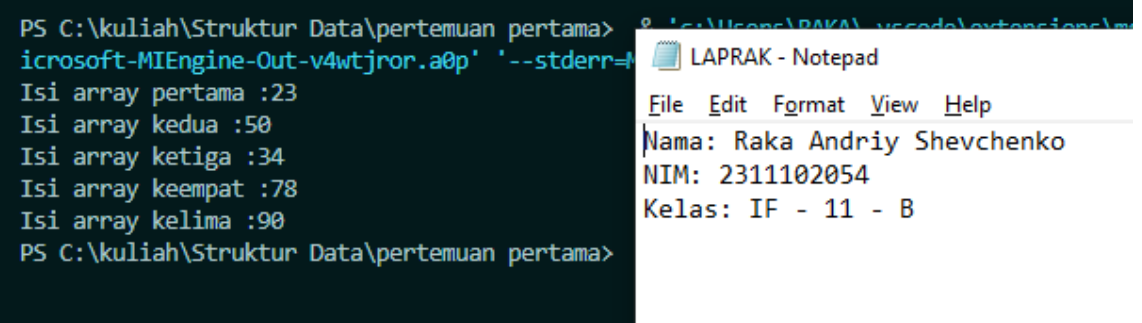
```

{
    //deklarasi dan inisialisasi array
    int nilai[5];
    nilai[0] = 23;
    nilai[1] = 50;
    nilai[2] = 34;
    nilai[3] = 78;
    nilai[4] = 90;

    //mencetak array
    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;
}

```

Output:



```

PS C:\kuliah\Struktur Data\pertemuan pertama> g++ *.cpp && .\program.exe
Microsoft-MIEngine-Out-v4wtjror.a0p' '--stderr=
Isi array pertama :23
Isi array kedua :50
Isi array ketiga :34
Isi array keempat :78
Isi array kelima :90
PS C:\kuliah\Struktur Data\pertemuan pertama>

```

LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko
 NIM: 2311102054
 Kelas: IF - 11 - B

Deskripsi:

Program ini dibuat untuk pengenalan *array*, *array* yang digunakan di program ini adalah *array* statis di mana kita memberi nilai pada setiap memori *array* secara manual. Program ini dibuat menggunakan tipe data koleksi berupa *array*, di dalam *source code* sudah dijelaskan bagaimana penggunaan tipe data *array*.

C. Unguided

a. Unguided 1

Source Code:

```
#include <cstdlib>
#include <iostream>

using namespace std;

//RANDOMIZER NO ABSEN
void absen() {
    int btm = 0;
    int top = 41;

    for (int i = 0; i < 5; i++){
        cout << (rand() % top - btm + 1) + btm << " ";
    }
}

//MENGHITUNG LUAS LINGKARAN
void ll() {
    int r;
    cout << "Masukkan jari-jari: ";
    cin >> r;

    cout << "Luas lingkaran anda adalah : "<< 3.14 * (r * r)
    << " satuan luas";
}

int main(int argc, char const *argv[])
{
    char pilih;

    cout << "Apakah menu yang ingin anda gunakan?\na.
Randomizer nomor absen\nb. Menghitung luas lingkaran\n";
    cin >> pilih;

    if (pilih == 'a' || pilih == 'A'){
        absen();
    } else if (pilih == 'b' || pilih == 'B') {
        ll();
    }
}
```



```
    return 0;
}
```

Output:

The screenshot shows a Windows command prompt window with the following text:

```
PS C:\kuliah\Struktur Data\TUGAS 1> & 'c:\Users\RAKA\.vs
IEngine-Out-3qz1ccyj.ryx' '--stderr=Microsoft-MIEngine-Er
Apakah menu yang ingin anda gunakan?
a. Randomizer nomor absen
b. Menghitung luas lingkaran
a
1 18 21 15 23
PS C:\kuliah\Struktur Data\TUGAS 1> & 'c:\Users\RAKA\.vs
IEngine-Out-tlissbiy.py0' '--stderr=Microsoft-MIEngine-Er
Apakah menu yang ingin anda gunakan?
a. Randomizer nomor absen
b. Menghitung luas lingkaran
b
Masukkan jari-jari: 15
Luas lingkaran anda adalah : 706.5 satuan luas
PS C:\kuliah\Struktur Data\TUGAS 1> 
```

To the right of the command prompt is a Notepad window titled '*Untitled - Notepad' with the following text:

```
File Edit Format View Help
Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B|
```

Deskripsi:

Program ini dibuat untuk 2 tujuan, yaitu randomizer nomor absen, dan juga menghitung luas lingkaran. Cara penggunaan randomizer adalah seperti berikut: Pertama, saya mendeklarasikan batas atas dan bawah dari nomor yang akan dipilih sistem; Kedua, saya menggunakan perulangan untuk seberapa banyak nomor yang akan sistem ambil, ini sebenarnya tidak perlu jika anda hanya ingin mengambil 1 nomor saja; Terakhir, saya memanggil fungsi rand() lalu menambahkan batas atas dan bawah pada fungsi tersebut.

Tipe data primitif adalah tipe data dasar yang sudah terdefinisi secara bawaan dalam bahasa pemrograman. Tipe data primitif biasanya digunakan untuk menyimpan nilai yang sederhana dan tidak kompleks, seperti bilangan bulat, bilangan pecahan, karakter, dan nilai boolean.

b. Unguided 2

1. Class

Class adalah *building block* yang mengarah pada pemrograman Berorientasi Objek. Ini adalah tipe data yang ditentukan pengguna, yang menyimpan anggota data dan fungsi anggotanya sendiri, yang dapat

diakses dan digunakan dengan membuat *instance class* tersebut. *Class* pada C++ seperti cetak biru untuk suatu objek. Misalnya: Perhatikan *Class* Mobil. Mungkin ada banyak mobil dengan nama dan merek berbeda tetapi semuanya memiliki beberapa kesamaan seperti semuanya memiliki 4 roda, Batas Kecepatan, Jarak tempuh, dll. Jadi di sini, Mobil adalah *Class*nya, dan roda, batas kecepatan, dan jarak tempuh adalah propertinya.

2. Struct

Struct adalah cara untuk mengelompokkan beberapa variabel terkait ke dalam satu tempat. Setiap variabel dalam *struct* dikenal sebagai anggota *structure*.

Berbeda dengan *array* yang hanya dapat berisi satu tipe data, *struct* dapat berisi banyak tipe data berbeda (int, string, float, dll.).

3. Contoh Penggunaan Class dan Struct

Source Code:

```
#include <iostream>
#include <string>

using namespace std;

class Class {
public:
    int absen;
    string nama;
    string kelas;
};

struct
{
    string tipe;
}Tp;

int main(int argc, char const *argv[])
{
    Class Obj;
```

```

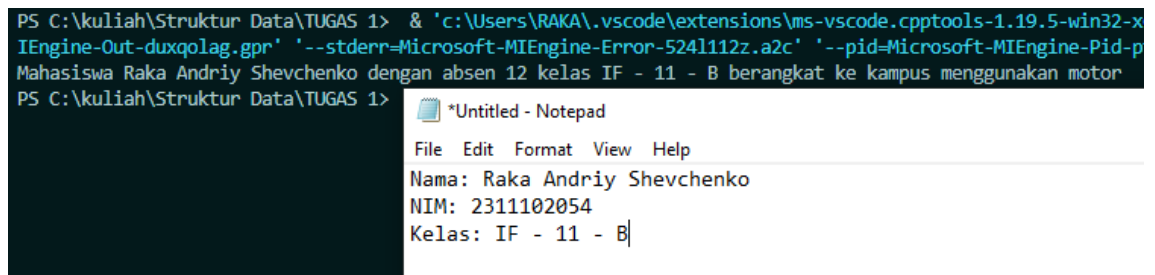
Obj.absen = 12;
Obj.nama = "Raka Andriy Shevchenko";
Obj.kelas = "IF - 11 - B";
Tp.tipe = "motor";

cout << "Mahasiswa " << Obj.nama << " dengan absen " <<
Obj.absen << " kelas " << Obj.kelas << " berangkat ke
kampus menggunakan " << Tp.tipe << endl;

return 0;
}

```

Output:



The screenshot shows a terminal window with the following command and output:

```

PS C:\kuliah\Struktur Data\TUGAS 1> & 'c:\Users\RAKA\.vscode\extensions\ms-vscode.cpptools-1.19.5-win32-x
IEngine-Out-duxqolag.gpr' '--stderr=Microsoft-MIEngine-Error-524112z.a2c' '--pid=Microsoft-MIEngine-Pid-p
Mahasiswa Raka Andriy Shevchenko dengan absen 12 kelas IF - 11 - B berangkat ke kampus menggunakan motor
PS C:\kuliah\Struktur Data\TUGAS 1>

```

Next to the terminal is a Notepad window titled '*Untitled - Notepad' with the following text:

```

File Edit Format View Help
Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B|

```

Deskripsi:

Program ini dibuat untuk mendata nama, absen, kelas, dan kendaraan yang dipakai mahasiswa untuk berangkat menuju kampus. Program ini dibuat menggunakan tipe data abstrak berupa *class* dan *struct*. Disini saya merubah *class*nya menjadi *public* agar dapat diakses saat ingin dioutputkan. Lalu saya membuat *struct* bernama Tp, pada int main saya memanggil semua tipe data yang saya punya. Pertama saya mendeklarasikan ulang *class* saya menjadi Obj, setelah itu mendeklarasikan nilai setiap isi *class* dan *struct* yang saya punya sesuai dengan nama dan fungsinya. Lalu terakhir saya *output*kan semuanya kedalam bentuk kalimat.

c. Unguided 3

Source Code:

```

#include <iostream>
#include <map>
#include <string>

```

```

using namespace std;

int main(int argc, char const *argv[])
{
    map<int, string> absen;

    absen[1] = "Raka";
    absen[2] = "Andriy";
    absen[3] = "Shevchenko";

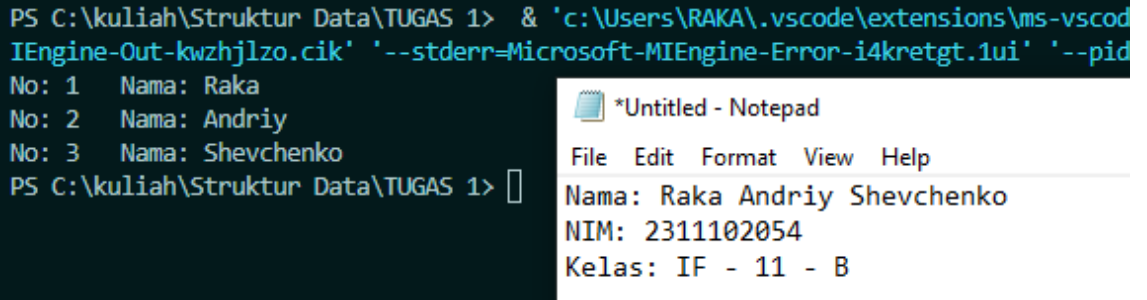
    map<int, string>::iterator itr;

    for (itr = absen.begin(); itr != absen.end(); ++itr) {
        cout << "No: " << itr->first << "\tNama: " << itr-
>second
            << '\n';
    }

    return 0;
}

```

Output:



The screenshot shows a terminal window with the following output:

```

PS C:\kuliah\Struktur Data\TUGAS 1> & 'c:\Users\RAKA\.vscode\extensions\ms-vscode
IEngine-Out-kwzhjlzo.cik' '--stderr=Microsoft-MIEngine-Error-i4kretgt.1ui' '--pid
No: 1    Nama: Raka
No: 2    Nama: Andriy
No: 3    Nama: Shevchenko
PS C:\kuliah\Struktur Data\TUGAS 1>

```

Overlaid on the terminal is a Notepad window titled '*Untitled - Notepad' with the following text:

```

File Edit Format View Help
Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B

```

Deskripsi:

Program ini dibuat untuk mendata absen dan nama mahasiswa, program ini dibuat menggunakan tipe data koleksi berupa tipe data *map*. Disini saya menggunakan *map* dengan *key* berupa integer yang berfungsi untuk mengisi absen mahasiswa, dan *value* berupa string yang berfungsi untuk mengisi nama mahasiswa. Setelah saya isi semua *key* dan *value*-nya, selanjutnya saya membuat iterator bernama *itr* untuk *map* tersebut. Setelah itu saya membuat perulangan *for* dengan mendeklarasikan *itr* untuk berada di awal *map*, lalu jika *itr* masih tidak sama dengan akhir dari *map*, maka perulangan akan dilanjutkan

dengan *itr* yang bertambah. Setelah itu saya memanggil *key* dan *value* dari *map* yang saya punya, lalu memisahkan kelanjutannya (*key-value* lain) dengan *newline* (line baru).

Perbedaan paling jelas antara *array* dengan *map* adalah fungsinya, *array* digunakan untuk menyimpan beberapa nilai (*value*) dalam satu variabel, sedangkan *map* digunakan untuk menyimpan pasangan *Key-Value items*. Jelasnya adalah pada bagaimana elemennya didefinisikan dan diakses. *Array* memiliki indeks yang berurutan dan semua elemennya harus sama jenis, sedangkan *map* menggunakan kunci (*key*) yang berbeda untuk setiap elemen dan tidak terbatas ke jenis elemen.

D. Kesimpulan

Tipe data adalah bagian penting dalam pemrograman, yang digunakan untuk menyimpan dan mengelola data. Dalam bahasa pemrograman C++, ada tipe data primitif, tipe data abstrak, dan tipe data koleksi. Tipe data primitif adalah tipe data yang sudah terdefinisi secara bawaan dalam bahasa pemrograman, seperti integer, float, boolean, dan char. Tipe data abstrak didefinisikan oleh pengguna sendiri, seperti *class* atau *struct*. Tipe data koleksi, seperti *array*, *vector*, dan *map*, digunakan untuk menyimpan kumpulan elemen dengan jenis yang sama (*array* dan *vector*). Sedangkan tipe data *map* digunakan untuk menyimpan kumpulan elemen dengan kunci dan nilai (*key-value*) yang berbeda. Setiap tipe data memiliki ukuran memori dan jangkauan yang berbeda, sehingga penggunaan tipe data yang tepat menjadi penting untuk membuat program yang efektif dan tidak boros memori.

E. Referensi

[1] MateriDosen. Pengertian dan Macam-macam Tipe Data C++, dengan Contoh Program. Diakses dari

<https://www.materidosen.com/2017/01/pengertian-dan-macam-macam-tipe-data-c.html?m=1>

[2] Amir Kamil. Abstract Data Types in C++. Diakses dari

https://eecs280staff.github.io/notes/08_ADTs_in_C%2B%2B.html

[3] Noval Agung Prayogo. A.16. Tipe Data → Vector. Diakses dari

<https://dasarpemrogramanrust.novalagung.com/basic/vector>

[4] Andre. Tutorial Belajar C++ Part 11: Jenis-jenis Tipe Data dalam Bahasa C++.

Diakses dari

<https://www.duniailkom.com/tutorial-belajar-c-plus-plus-jenis-jenis-tipe-data-dalam-bahasa-c-plus-plus/>