**TUGAS KELOMPOK**

**ALGORITMA DAN PEMROGRAMAN**

**A11.4105**



**IMPLEMENTASI**

**ABSTRACT DATA TYPE**

**Ferdian Nur Fariza A11.2023.15074**

**Fernandito Ibrahim Maryana A11.2023.15071**

**UNIVERSITAS DIAN NUSWANTORO**

**DESEMBER**

**2023**

# SOAL

1. Carilah minimal 4(empat) artikel yang membahas konsep Abstract Data Type.
2. Lakukan analisis terhadap artikel tersebut meliputi:
   1. Bagaimana konsep ADT berjalan?
   2. Mengapa harus ada ADT?
   3. Apa hubungannya ADT dengan Object Oriented Programming (OOP)?
   4. Berikan contoh penerapan ADT dengan perancangan program sederhana.

# PEMBAHASAN

1. Artikel Dari Google Books:
   1. Introduction to Programming with C++ for Engineers (by Bogusław Cyganek)
   2. The Fundamentals of C++ Progrmming Language (by Alexander Aronowitz)
   3. Mastering C++ Programming Language (by Divya Sachdeva and Natalya Ustukpayeva)
   4. C++ Template Metaprogramming in Practice (by Li Wei)
2. Hasil analisis dari 4 referensi artikel:
   1. Konsep abstraksi dalam ilmu komputer sering ditemui: saat mewakili algoritma, itu hanya memberi tahu tingkat detail apa yang diperlukan (Cyganek, 2020). Di dalam buku tersebut dijelaskan bahwa abstraksi dapat di katakan bahwa program hanya akan menampilkan sesuatu yang diperlukan saja tanpa menampilkan detailnya.

Jenis-jenis ini biasanya disebut sebagai tipe data abstrak meskipun mungkin lebih baik menyebutnya *user-defined* (Aronowitz, 2020 ). Hal tersebut berkaitan dengan istilah Abtract Data Type (ADT) karena ADT adalah contoh dari implementasi dari konsep abstraksi dimana kita dapat mendefinisikan tipe data atau fungsi tertentu tanpa harus memberi tahu kepada pengguna tentang detailnya.

Array adalah kumpulan elemen dari tipe data yang sama, dan struktur adalah kumpulan elemen dari berbagai tipe data (Aronowitz, 2020 ). Dengan menggunakan *struct* dalam C++ memungkinkan kita untuk membuat tipe data baru dari berbagai tipe data.

* 1. Bahasa tidak mendukung gaya pemrograman tertentu jika dibutuhkan banyak upaya atau bahkan seni untuk menulis program dengan gaya tersebut. Namun, ini tidak berarti bahwa bahasa tersebut melarang program menulis dalam hal ini gaya (Aronowitz, 2020 ). Bukan tidak mungkin untuk menulis program seperti ADT dalam bahasa C, namun tidak efisien dan buang-buang energi. Berbeda dengan bahasa C++ yang mendukung abstraksi data dan mendukung pemrograman berorientasi objek dibanding bahasa C sehingga kita dapat menerapkan konsep ADT dengan lebih mudah dan efisien.

Dalam buku Aronowitz bahasa pemrograman C++ memiliki kelebihan yaitu lebih baik dari bahasa C, mendukung abstraksi data, dan mendukung pemrograman berorientasi objek. Bahasa dikatakan mendukung gaya pemrograman tertentu jika memiliki fitur yang membuat pemrograman dalam gaya itu nyaman (sederhana, *reliable*, dan efisien).

ADT menjadi penting karena memiliki banyak peran di dalam pembuatan atau pengembangan suatu program. Tipe data abstrak didefinisikan sebagai semacam "kotak hitam" (Cyganek, 2020). Dijelaskan bahwa ADT seperti kotak kosong yang menyembunyikan struktur dan apa yang ada didalamnya. Ini seperti yang dijelaskan oleh Pak Danang bahwa seorang developer game tidak akan mengungkapkan cara dia membuat gamenya, karena jika diungkapkan maka orang tidak akan membeli game tersebut namun orang akan membuatnya sendiri.

Maka diperlukan ADT dalam membuat program yang memungkinkan untuk menyembunyikan detail dan struktur di dalamnya sehinga developer memiliki privasi atas programnya sendiri. Dengan adanya ADT *developer* dengan mudah melakukan tinjauan kode yang telah dibuat jika terjadi *bug* ataupun kesalahan kode dengan meninjau kode dengan lebih efisien dan fleksibel. Tentunya hal tersebut meningkatkan produktivitas *developer* tersebut.

* 1. Tujuan mendasar dari pemrograman C ++ adalah untuk mengenalkan Object Orientation ke bahasa pemrograman C, dan class, juga dikenal sebagai user-defined type, adalah elemen kunci dari C ++ yang memungkinkan object-oriented programming (Sachdeva & Ustukpayeva, 2022).

Hubungan OOP dan ADT adalah keduanya menerapkan konsep menyembunyikan detail dan hanya menampilkan yang dibutuhkan. ADT dan OOP merupakan *user-defined type* jadi hal ini menjadi salah satu alasan bahwa keduanya saling berhubungan. Bedanya jika ADT bekerja untuk tipe data dan operasi di dalam tipe data, jika dalam OOP kita menggunakan representasi dari suatu objek dan bagaimana objek tersebut berinteraksi dengan objek yang lain.

Kata kunci *class* digunakan untuk memulai definisi *class*, diikuti oleh *class name*, dan *class body* kita dapat menentukan bentuk objek dengan menggabungkan representasi data dan metode untuk mengubah data itu menjadi satu paket. Anggota kelas adalah data dan *method* yang membentuk kelas (Sachdeva & Ustukpayeva, 2022).

* 1. Program sederhana “Program Input 3 Data Mahasiswa” merupakan penerapan ADT dengan perancangan :

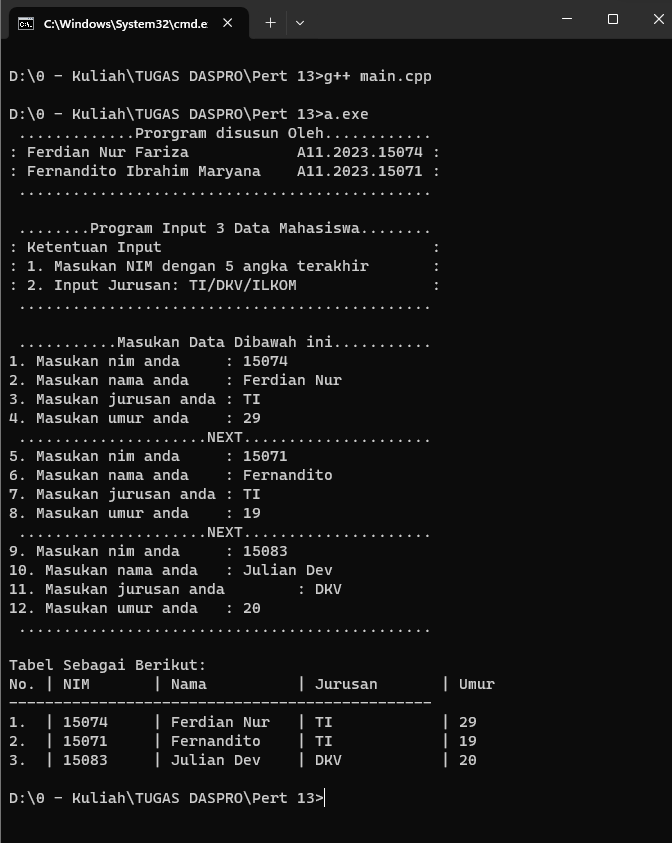
ADT,

Fungsi (Void), dan

Array 2D.

File program bernama main.cpp di folder dengan spesifikasi program yang dapat menampilkan tabel dari nilai yang di masukan oleh pengguna namun dengan syarat tertentu.

Berikut merupakan *screenshot* hasil dari program ketika selesai di jalankan :



Gambar 1. Screenshot hasil program.

# REFERENSI

Aronowitz, A. (2020 ). *The Fundamentals of C++ Progrmming Language.* Amazon Digital Services LLC.

Cyganek, B. (2020). *Introduction to Programming with C++ for Engineers.* Poland: Wiley.

Sachdeva, D., & Ustukpayeva, N. (2022). *Mastering C++ Programming Language.* CRC Press.

Wei, L. (2020). *C++ Template Metaprogramming in Practice.* CRC Press.