

ISI LAPORAN TUBES ALGORITMA DAN PEMROGRAMAN 2

TUBES PROBLEM / TOPIC - 29

Nama Anggota Kelompok :

- Mikael Eureka Anakotta - 103012400420

- Dillah Emeylia Putri - 103012430061

1. Pendahuluan

Tugas Besar ini dibuat dengan tujuan untuk bikin sebuah aplikasi yang bisa bantu para seniman digital dan kolektor NFT dalam mengatur koleksi mereka. Jadi, lewat aplikasi ini, mereka bisa lebih gampang nyimpen, ngelola, dan ngecek semua NFT atau karya digital yang mereka punya. Karena belakangan ini tren NFT lagi naik daun banget, apalagi sejak teknologi blockchain makin populer, makin banyak orang yang mulai tertarik buat ngoleksi karya digital dalam bentuk NFT.

Masalahnya, sampai sekarang masih banyak dari mereka yang nyimpen datanya secara manual, nyebar di banyak tempat, atau bahkan cuma catatan biasa aja. Nah, hal kayak gitu bisa bikin ribet dan susah buat dipantau, apalagi kalau koleksinya udah banyak. Makanya, dibutuhkan satu solusi yang simpel tapi tetap berguna buat bantu mereka ngatur semuanya dalam satu aplikasi aja.

Lewat aplikasi ini, pengguna bisa nyatat semua data NFT-nya, mantau harga-harga yang berubah-ubah, dan ngelihat gimana performa portofolionya. Jadi, selain buat nyimpen data, aplikasi ini juga bisa bantu pengguna ngambil keputusan yang lebih baik soal investasi NFT mereka. Intinya sih, kita pengen bikin aplikasi yang sederhana, gampang dipakai, tapi tetap punya fungsi-fungsi penting buat kolektor dan seniman digital.

2. Deskripsi Tugas Besar

Aplikasi yang dikembangkan dalam tugas besar ini adalah sebuah **aplikasi manajemen portofolio NFT** berbasis terminal menggunakan bahasa pemrograman **Go (Golang)**. Aplikasi ini dirancang untuk membantu pengguna dalam mencatat, mengelola, serta menilai NFT (Non-Fungible Token) yang mereka miliki. Meskipun sederhana, aplikasi ini mencakup berbagai fitur fungsional yang mendukung aktivitas dasar dalam pengelolaan aset digital.

Secara umum, aplikasi ini menggunakan pendekatan **pemrograman prosedural** dengan memanfaatkan array statik dan struktur data struct untuk menyimpan data NFT. Data NFT disimpan dalam array berukuran tetap, yang memungkinkan pengguna untuk menyimpan hingga 1000 NFT dalam satu sesi.

ISI LAPORAN TUBES ALGORITMA DAN PEMROGRAMAN 2

Setiap NFT terdiri dari tiga atribut utama, yaitu:

- **ID** (bertipe string) sebagai identifikasi unik dari setiap NFT,
- **Nama** karya NFT,
- **Harga** NFT dalam bentuk desimal (float64).

Fitur-fitur utama yang disediakan oleh aplikasi ini antara lain:

1. **Menambahkan NFT baru**, di mana pengguna dapat menginput data NFT untuk kemudian disimpan ke dalam array utama.
2. **Mengedit NFT**, memungkinkan pengguna memperbarui nama dan harga NFT yang sudah ada berdasarkan ID-nya.
3. **Menghapus NFT**, yaitu menghapus NFT tertentu dan menggeser data dalam array untuk mencegah kekosongan data di tengah array.
4. **Mencari data NFT**, tersedia dua metode pencarian:
 - **Sequential Search**, digunakan untuk mencari NFT berdasarkan nama dan harga.
 - **Binary Search**, digunakan untuk mencari NFT berdasarkan ID, namun mensyaratkan data ID sudah diurutkan terlebih dahulu.
5. **Mengurutkan data NFT**, terdapat dua metode pengurutan:
 - **Selection Sort** untuk mengurutkan NFT berdasarkan ID,
 - **Insertion Sort** untuk mengurutkan berdasarkan harga.
6. **Menampilkan riwayat aktivitas**, yang mencatat setiap proses penambahan, pengeditan, atau penghapusan NFT selama aplikasi berjalan.
7. **Menghitung total nilai portofolio NFT**, yakni menjumlahkan seluruh harga NFT untuk menampilkan nilai total aset yang dimiliki.
8. **Mencetak daftar NFT**, menampilkan data NFT yang tersimpan dalam array secara rapi ke layar terminal.

Semua proses di atas dilakukan secara interaktif melalui menu pilihan di terminal. Aplikasi ini juga menjaga kepraktisan dan kemudahan penggunaan dengan menampilkan pesan-pesan yang membimbing pengguna dalam setiap proses.

Penggunaan algoritma sorting dan searching pada aplikasi ini bukan hanya berfungsi sebagai fitur, namun juga menjadi bentuk implementasi konsep dasar algoritma dan struktur data dalam pengembangan perangkat lunak. Dengan begitu, aplikasi ini tidak hanya fungsional tetapi juga memiliki nilai edukatif dalam konteks pembelajaran algoritma dan pemrograman dengan Go.

ISI LAPORAN TUBES ALGORITMA DAN PEMROGRAMAN 2

2. Penjelasan Tugas Besar

A. APA YANG SUDAH DIKERJAKAN

Fitur-Fitur Dasar:

1. Struktur Data

- a. NFT menyimpan informasi: nama, id, dan harga
- b. arrNFT adalah array statik untuk menyimpan data NFT (maksimal 1000)
- c. historyArr menyimpan catatan aktivitas pengguna

2. Fitur CRUD

- a. **Tambah NFT (tambahNFT)**
 - Menyimpan data baru ke array dataNFT
 - Mencatat aktivitas ke history
- b. **Edit NFT (editNFT)**
 - Mencari NFT berdasarkan ID, lalu mengubah nama dan harga
- c. **Hapus NFT (hapusNFT)**
 - Mencari NFT berdasarkan ID, lalu menghapus data tersebut dan menggeser elemen array
- d. **Cetak Semua NFT (cetakSemuaData)**
 - Menampilkan semua data NFT yang telah dimasukkan dalam bentuk tabel

3. Pencarian NFT

- a. Sequential Search by Nama (sequentialSearch)
- b. Binary Search by ID (binarySearch)
Mengurutkan dulu dengan selection sort, baru cari
- c. Sequential Search by Harga (sequentialSearchHarga)

4. Pengurutan Data

- a. **By ID**
 - selectionSortByID()
 - insertionSortByID()
- b. **By Harga**
 - selectionSortByHarga()
 - insertionSortByHarga()

5. Utility Tambahan

- a. Menampilkan History Aktivitas (tampilkanHistory)
- b. Menghitung Total Nilai Portofolio (tampilkanTotal)

ISI LAPORAN TUBES ALGORITMA DAN PEMROGRAMAN 2

6. Navigasi Menu

- a. Sudah ada sistem menu berbasis input angka dengan opsi 1–8

Yang Sedang di Kerjakan

1. **Penyempurnaan sistem navigasi dan alur menu** agar lebih nyaman digunakan
2. **Tampilan output**, seperti tabel data NFT dan format harga, agar lebih rapi
3. **Stabilisasi dan debugging**, memastikan semua fitur bekerja dengan benar
4. **Penanganan input**, seperti validasi input agar tidak error jika user salah mengetik

B. Tantangan dan masalah yang dihadapi

1. Duplikasi ID Tidak Dicek

- **Masalah:** Saat tambah NFT, kamu tidak cek apakah ID yang dimasukkan sudah ada.
- **Risiko:** Duplikasi ID membuat hasil pencarian dan penghapusan menjadi tidak akurat.

1. `fmt.Scan()` Tidak Aman untuk Input Nama Lebih dari 1 Kata

- **Masalah:** `fmt.Scan(&nama)` hanya membaca 1 kata.
- **Dampak:** Nama NFT seperti "Crypto Punk" jadi hanya "Crypto".

2. Fungsi Binary Search Tidak Akurat Tanpa Data Terurut

- **Masalah:** Binary search hanya bekerja jika data sudah **terurut**. Kamu urutkan pakai `selectionSortByID()`, tapi:
 - Itu hanya dipanggil di dalam binary search, bukan saat tambah data.
 - Tidak semua pengguna sadar bahwa binary search butuh data yang urut.

3. History Tidak Ada Informasi Detail

- **Masalah:** History hanya mencatat "Menambahkan NFT", tanpa ID atau nama.
- **Kurang Informatif** untuk user.

4. Validasi Input Pengguna Masih Kurang

- **Masalah:** Pengguna bisa saja menginput:
 - Harga negatif, Nama kosong, ID duplikat
- **Tantangan:** Ini bisa merusak data dalam array.

5. Tidak Ada Fungsi untuk Menyimpan dan Membaca Data dari File

- **Masalah:** Data hanya ada di memori, dan **hilang saat aplikasi ditutup**.
- **Tantangan:** Pengguna harus input ulang setiap kali buka program.

6. Tidak Ada Batas Maksimal Jumlah History

- **Masalah :** jika user menambah/ edit terus menerus, array `historyArr` bisa melebihi batas max
- **Tantangan :** bisa menyebabkan array out of bounds error

ISI LAPORAN TUBES ALGORITMA DAN PEMROGRAMAN 2

C. Apa yang harus ditambahkan atau direncanakan

1. Validasi Input User

Yang bisa ditambahkan:

- Cek **ID duplikat** saat tambah NFT.
- Pastikan **harga tidak boleh negatif**.
- Pastikan **nama tidak kosong** dan bisa lebih dari 1 kata (bufio.NewReader).
- Cek input tidak valid (angka/huruf) agar tidak menyebabkan crash.

2. Simpan & Baca Data dari File (Persistent Storage)

Masalah sekarang: Data hilang kalau aplikasi ditutup.

Solusi:

- Tambah fungsi `simpanKeFile()` dan `bacaDariFile()`.
- Bisa simpan data ke file .txt atau .json sederhana.

3. Sim Statistik Portofolio

Tujuan: Bikin aplikasi lebih informatif dan menarik.

Fitur tambahan yang bisa direncanakan:

- Tampilkan **NFT termahal dan termurah**.
- Tampilkan **rata-rata harga NFT**.
- Hitung **jumlah NFT berdasarkan rentang harga** (misalnya murah, sedang, mahal).

4. History Lebih Detail

Masalah sekarang: History terlalu umum, contoh: "Menambahkan NFT".

Yang bisa ditingkatkan:

- History ditulis seperti:
"Menambahkan NFT: ID=12, Nama=CryptoCat, Harga=5000"
- Tampilkan waktu (kalau mau lebih lengkap, pakai `time.Now()`).

5. Perbaiki Struktur Data dan Logika

Contoh yang perlu dirapikan:

ISI LAPORAN TUBES ALGORITMA DAN PEMROGRAMAN 2

- Ganti jumlahNFT := -1 jadi jumlahNFT := 0 (lebih natural).
- Bersihkan slot terakhir saat data dihapus (dataNFT[jumlahNFT] = NFT{}).
- Gunakan slice ([]NFT) agar lebih fleksibel dari pada array fixed.

6. Peningkatan UX (User Experience)

Misalnya:

- Tampilan lebih rapi (gunakan garis, tab, atau warna kalau di terminal).
- Tambahkan loading palsu biar terasa interaktif (misal: `fmt.Print("Menyimpan...")` lalu `time.Sleep(1 * time.Second)`).

yang harus direncanakan

Pada fitur edit NFT, penulis merancang agar pengguna memiliki keleluasaan dan memilih data yang ingin diubah, baik nama maupun harga nft. sebelum perubahan disimpan, sistem akan menampilkan perbandingan antara data sebelum dan sesudah dilakukan pengeditan titik setelah itu, pengguna diberikan dua opsi yaitu:

1. Setuju- untuk menyiapkan perubahan atau
2. batalkan-untuk membatalkan proses pengeditan

Tujuan dari rencana ini adalah untuk meningkatkan fleksibilitas dan akurasi input data sekaligus memberikan kesempatan bagi pengguna untuk meninjau kembali perubahan sebelum data benar-benar diperbarui dalam sistem.