

LAPORAN PROGRESS TUGAS BESAR ALGORITMA PEMROGRAMAN

Topik: Aplikasi Pelacak Gaya Hidup Ramah Lingkungan



KELOMPOK 10 (Ngantuk)

Brama Hartoyo 103052400030
Zaky Muhammad Fauzi 103052400064

**PROGRAM STUDI S-1 SAINS DATA
FAKULTAS INFORMATIKA
UNIVERSITAS TELKOM
2025**

1. Pendahuluan

Di era digital dan krisis iklim masa ini, kesadaran betapa pentingnya dalam menjaga lingkungan sangatlah penting. Sebagai mahasiswa Data Sains, kami menyadari bahwa teknologi dapat digunakan sebagai alat bantu dalam perubahan yang positif terhadap gaya hidup manusia, salah satunya adalah dengan aplikasi berbasis algoritma yang dapat membantu masyarakat melacak dan mengelola aktivitas sehari-hari yang berdampak pada lingkungan.

Tugas Besar ini adalah bagian dari mata kuliah Algoritma Pemrograman yang bertujuan untuk mengembangkan kemampuan mahasiswa dalam membuat sebuah aplikasi sederhana berbahasa go, yaitu **Aplikasi Pelacak Gaya Hidup Ramah Lingkungan** yang dapat membantu pengguna dalam mengelola dan memantau aktivitas sehari-hari mereka terkait dengan dampak karbon yang dihasilkan dalam suatu rentang waktu.

Tujuan utama dari aplikasi ini adalah merancang dan mengimplementasikan sebuah aplikasi alat bantu interaktif yang dapat menilai, mencatat, memberikan rekomendasi, dan mendorong masyarakat agar lebih sadar akan dampak aktivitas mereka untuk tetap melindungi dan menjaga lingkungan, khususnya mengurangi dan meminimalisir emisi karbon.

2. Deskripsi Tugas Besar

Masalah yang diangkat adalah kurangnya kesadaran masyarakat terhadap aktivitas sehari-hari yang berdampak pada lingkungan. Aplikasi ini bertujuan untuk membantu pengguna memantau dan mengevaluasi aktivitasnya agar lebih ramah lingkungan dan berkelanjutan. Setiap aktivitas akan dikonversi menjadi estimasi jejak karbon (*carbon footprint*) dengan penggunaan formula sederhana yang diterapkan melalui logika pemrograman aplikasi ini. Hasil dari perhitungan ini akan ditampilkan dalam bentuk laporan bulanan sehingga pengguna dapat mengevaluasi dan memperbaiki gaya hidupnya.

3. Fitur-Fitur Program

a. Fitur utama program :

1. Input aktivitas harian pengguna sebanyak 100 data maksimal (jika data melebihi kapasitas, maka mencetak “aktivitas penuh”).
2. Pencarian aktivitas berdasarkan kategori yang ingin dicari (pencarian data dapat secara *sequential search* maupun *binary search*).

3. Pengeditan aktivitas berdasarkan ID aktivitas yang ingin diubah (mengubah isi aktivitas maupun menghapus aktivitas).
 4. Pengurutan berdasarkan dampak karbon yang ditimbulkan (*selection sort*) atau frekuensi (*insertion sort*) dari waktu aktivitas secara *ascending* (terkecil ke terbesar) atau *descending* (terbesar ke terkecil).
 5. Laporan bulanan emisi karbon pengguna (memberikan rekomendasi pengurangan emisi dalam menjalani gaya hidup yang lebih berkelanjutan dan ramah lingkungan).
 6. Keluar program.
- b. Konstanta variabel global **MAX_AKTIVITAS** (berjumlah 100).
 - c. Variabel global array **daftarAktivitas** dengan batas maksimal sebanyak **MAX_AKTIVITAS** dengan tipe data alias Aktivitas.
 - d. Variabel global **jumlahAktivitas** sebagai jumlah array yang terisi.
 - e. Tipe bentukan **Aktivitas** merepresentasikan aktivitas yang berdampak pada lingkungan :
 1. ID: identifikasi unik untuk atribut aktivitas (tipe data integer).
 2. Kategori: jenis aktivitas, seperti "Transportasi" atau "Energi" (tipe data string).
 3. Deskripsi: penjelasan singkat seperti "Mobil" atau "Sepeda" (tipe data string).
 4. DampakKarbon: jumlah CO₂ (dalam kg) yang dihasilkan per aktivitas (tipe data real)
 5. Frekuensi: seberapa sering aktivitas tersebut dilakukan per bulan (tipe data integer).
 - f. Function **toLower** membuat inputan string yang kapital (A sampai Z) menjadi seragam berupa string huruf kecil (a sampai z).
 - g. Function **findAktivitasIndexByID** mencari indeks aktivitas berdasarkan id yang dimasukan pengguna untuk peneditan aktivitas (mengubah aktivitas atau menghapus aktivitas).
 - h. Procedure **tambahAktivitas** untuk menambahkan data aktivitas baru ke array daftarAktivitas selama belum mencapai kapasitas maksimum dengan menambahkan entri dan meng-update jumlahAktivitas.
 - i. Function **cariSequential** untuk mencari data secara linear dengan mengembalikan nilai indeks aktivitas yang kategorinya sesuai dengan input atau -1 jika tidak ditemukan.
 - j. Procedure **selectionSortKategori** mengurutkan daftarAktivitas berdasarkan nilai terkecil array kategori dan menukar posisinya.
 - k. Function **cariBinary** mencari data dalam array yang sudah terurut berdasarkan kategori (syarat penggunaan fungsi ini data harus terurut)

- l. Procedure **editAktivitas** dilakukan edit data pada ID aktivitas yang dipilih atau melakukan hapus data pada array dengan dilakukan shifting/pindah elemen ke kiri dan mengurangi jumlahAktivitas sebanyak 1 (menimpa data agar array tetap berurut).
- m. Procedure **selectionSortDampak** mengurutkan aktivitas berdasarkan nilai DampakKarbon, tergantung parameter *ascending* (true/false).
- n. Procedure **insertionSortFrekuensi** mengurutkan aktivitas berdasarkan frekuensi per bulan.
- o. Function **tampilkanDaftar** mencetak seluruh isi array daftarAktivitas yang sudah diisi data aktivitas.
- p. Procedure **tampilkanHasilPencarian** untuk mencetak seluruh aktivitas yang dicari berdasarkan kategori aktivitas yang dimasukkan pengguna.
- q. Function **laporanBulanan** menampilkan laporan akumulasi bulanan dengan total emisi CO₂, aktivitas dampak karbon terbanyak dan terendah, skor keberlanjutan (dengan rumus skor = $100 - (\text{totalDampak} / 100)$), jika skor < 0, maka skor = 0, jika skor > 100, maka skor = 100), rekomendasi alternatif yang lebih ramah lingkungan untuk kategori yang memiliki kontribusi lebih dari 500 atau 200 total jejak karbon.

4. Output Program

```

tubesrevisi.go X
C:\Users\soeka> OneDrive\Documents> Tugas Semester 2> Algoritma Pemrograman> Minggu 15> tubesrevisi.go
359 func main() {
360     // 1. Tambah Aktivitas
361     fmt.Scan(&dampak)
362     fmt.Print("Masukkan Frekuensi (kali per bulan, contoh: 20): ")
363     fmt.Scan(&frekuensi)
364     tambahAktivitas(id, kategori, deskripsi, dampak, frekuensi)
365 } else if pilihan == 2 {
366     fmt.Print("Masukkan Kategori yang dicari (satu kata, contoh: Transportasi): ")
367     fmt.Scan(&kategori)
368     fmt.Println("Pilih metode pencarian:")
369     fmt.Println("1. Sequential Search")
370     fmt.Println("2. Binary Search")
371     fmt.Print("Pilih metode (berupa angka, contoh: 1): ")
372     fmt.Scan(&subPilihan)
373     if subPilihan == 1 {
374         hasilIndeks := cariSequential(kategori)
375         tampilkanHasilPencarian(hasilIndeks)
376     } else if subPilihan == 2 {
377         selectionSortKategori() // Urutkan dulu sebelum binary search
378         hasilIndeks := cariBinary(kategori)
379         tampilkanHasilPencarian(hasilIndeks)
380     } else {
381         fmt.Println("Pilihan tidak valid!")
382     }
383 }
384 }
385
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\soeka\OneDrive\Documents> Tugas Semester 2\Algoritma Pemrograman\Minggu 15> go run tubesrevisi.go
=== Aplikasi Pelacak Gaya Hidup Ramah Lingkungan ===
1. Tambah Aktivitas
2. Cari Aktivitas
3. Edit Aktivitas
4. Urutkan
5. Tampilkan Daftar Aktivitas
6. Laporan Bulanan
0. Keluar
Pilih menu (berupa angka, contoh: 1):

```

5. Tantangan dan Solusi

1. Validasi masukan pengguna berupa data yang tidak valid, seperti angka negatif atau huruf pada input angka. (Solusi dengan memberi penjelasan pemisalan dalam input data untuk memastikan bahwa input yang dimasukkan sesuai dengan tipe data yang diharapkan juga dengan penambahan fitur `toLowerCase` untuk membantu *autocorrect* input pengguna).

6. Kesimpulan dan Rekomendasi

1. Implementasi aplikasi berbasis web atau desktop supaya aplikasi lebih menarik dan mudah dijangkau oleh masyarakat luas.
2. Penambahan fitur penyimpanan data secara permanen ke dalam file (misalnya `.json` atau `.csv`) atau database agar data tetap tersimpan jika aplikasi ditutup/mengalami shutdown.
3. Integrasi dengan waktu sistem secara *real time* dan kalkulasi otomatis untuk peningkatan keakuratan evaluasi dan laporan dari dampak emisi karbon.

7. Kode Program Tautan GitHub

<https://github.com/FtBdra/TubesNgantuk/blob/main/BahasaGo.go>

8. Referensi

- Prasti E. Y., Bunyamin, Said A. F. 2023. Pengenalan Algoritma Pemrograman dalam Bahasa Go dan Python (Tel-U Press).
- Z.K. Abdurahman Baizal dan Yuliant Sibaroni, 2023. Buku Ajar Algoritma dan Pemrograman Menggunakan Bahasa Pemrograman Go (Golang): Diandra Creative. Yogyakarta.