



## Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)  
Tahun / Semester : 2021-2022 / Genap  
Modul : 4 - Structures and Dynamic Arrays  
Hari, Tanggal Praktikum : Kamis, 17 Maret 2022

---

### Naskah Soal Praktikum

Pembuat Naskah: Kevin Naoko, Muhammad Morteza Mudrick

#### Ketentuan:

1. Kerjakanlah satu dari dua soal berikut pada *template repository* yang anda peroleh ketika mengambil *assignment* di GitHub Classroom praktikum!
2. *Commit* yang dilakukan setelah sesi praktikum berakhir tidak akan dipertimbangkan dalam penilaian.
3. *Header* setiap *file* harus mengikuti format yang telah disediakan pada *file template repository*. *Header* yang tidak mengikuti format tersebut tidak akan dinilai.
4. Buku catatan laboratorium yang berisi alasan pemilihan soal, *flowchart*, dan *data flow diagram* dari solusi yang anda buat dikumpulkan ke tugas.stei.itb.ac.id paling lambat pukul 11.00 WIB satu hari kerja setelah sesi praktikum.
5. Solusi soal pertama harus dapat dikompilasi dengan perintah `make soal-01` dan menghasilkan *file executable* dengan nama `soal-01`. Demikian pula, soal kedua harus dapat dikompilasi dengan perintah `make soal-02` dan menghasilkan *file executable* dengan nama `soal-02`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

## Soal 1

Anda ditugaskan untuk **menghitung jarak total yang akan ditempuh** oleh pilot helikopter, serta **menghitung apakah bahan bakar yang ada di helikopter mencukupi untuk menempuh jarak tersebut**. Anda hanya perlu mengimplementasikan fungsi `calculateFuelUsed` dan `calculateDistance` saja. Beberapa fungsi pendukung yang lain sudah diberikan. Input dari program ini berupa file `.csv`.

**Jarak yang perlu dihitung adalah jarak total dari titik  $a_0, a_1, a_2, \dots, a_{n-1}, a_n$ .** Titik  $a_0$  dan  $a_n$  adalah titik awal dan titik akhir dari helikopter. Titik  $a_0$  dan  $a_n$  adalah sama, yaitu  $\{0,0,0\}$ . Titik lainnya ( $a_1, a_2, \dots, a_{n-1}$ ) merupakan titik kunjung helikopter. Setiap titik memuat koordinat dalam bidang 3 dimensi, yang dapat dituliskan sebagai berikut

$$a_i = \{x_i, y_i, z_i\}$$

Masukan file `.csv` memuat informasi nama, jumlah bahan bakar awal (dalam liter), faktor konsumsi bahan bakar (dalam km/liter), serta beberapa titik koordinat  $\{x_n, y_n, z_n\}$  (dalam km). **Perhatikan bahwa titik yang dimuat dalam `.csv` hanya titik kunjung helikopter ( $a_1, a_2, \dots, a_{n-1}$ ). Jangan lupa untuk memasukkan titik  $a_0$  dan  $a_n$  dalam perhitungan.**

**Input program adalah sebagai berikut.** Pengguna dapat memasukkan beberapa input file berupa file `.csv` yang berisi data prajurit. Bila file tidak ditemukan, program akan mengabaikan masukan tersebut dan meminta lagi. Program juga akan terus menerima masukan input file hingga pengguna menuliskan kata 'exit'. Setelah selesai membaca file, program akan menghitung jarak total yang ditempuh, serta konsumsi bahan bakar. Bila bahan bakar mencukupi, program menampilkan sisa bahan bakar. Bila tidak mencukupi, program menampilkan output jumlah bahan bakar yang perlu diisi (**perhatikan fungsi `processOutput` pada template kode**). Lebih detailnya, output program dapat dilihat pada contoh eksekusi program dibawah

Perhitungan jarak antar titik dalam bidang 3 dimensi dapat dilakukan dengan rumus berikut

$$\text{Jarak titik } a_1 \text{ dengan } a_2 = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Misalkan pada file `data1.csv` terdapat 4 titik koordinat ( $a_1, a_2, a_3$ , dan  $a_4$ ). Jarak total yang perlu dihitung adalah jarak antara  $a_0$  ke  $a_1$ ,  $a_1$  ke  $a_2$ , dan seterusnya hingga jarak  $a_4$  ke  $a_5$ . Dalam kasus ini, titik  $a_0$  dan

$a_5$  adalah  $\{0, 0, 0\}$ . Setelah mendapatkan total jarak tempuh, perhitungan jumlah bahan bakar dapat dilakukan dengan rumus berikut

$$\text{Konsumsi bahan bakar} = \frac{\text{jarak total (km)}}{\text{faktor konsumsi bahan bakar (km/liter)}}$$

Untuk memudahkan, silahkan menggunakan *array of struct* yang setiap indeksnya menyimpan data satu prajurit.

Beberapa ketentuan dalam pembuatan program:

- Praktikan harus menggunakan struktur data, namun isi variabelnya boleh diubah.
- Untuk menyimpan data banyak prajurit, praktikan diwajibkan menggunakan *dynamic array of struct*.
- Untuk menyimpan sekumpulan koordinat x, y, dan z, praktikan diwajibkan menggunakan *dynamic array of float*.
- Variabel untuk menyimpan input nama file **tidak perlu** menggunakan *dynamic array of characters*. Nama file diasumsikan tidak lebih dari 30 karakter.
- Variabel buffer untuk membaca tiap line pada file juga **tidak perlu** menggunakan *dynamic array*. Asumsikan panjang buffer sebesar 512.
- Bila file input user tidak ditemukan, program akan meminta user menginput nama file kembali
- Program akan terus meminta masukkan pengguna hingga diberi masukkan 'exit' tidak perlu dibuat lagi (**sudah dihandle oleh template kode**)

### Tips

- Hitung jumlah koordinat yang dimuat dalam di file.csv saat proses membaca file, agar pemrosesan selanjutnya lebih mudah
- Gunakan fungsi `malloc` untuk inisialisasi array, serta fungsi `realloc` untuk menambahkan alokasi data.
- Karena jumlah elemen dari array tidak diberikan terlebih dahulu, sebaiknya gunakan fungsi `realloc` setiap kali ingin menambahkan indeks ke suatu array.
- Gunakan fungsi `processOutput` untuk print output agar dapat dinilai oleh *autograder*.
- Untuk keperluan operasi pangkat dan akar, gunakan fungsi `power` dan `squareRoot` yang terdapat pada template kode, agar autograder dapat melakukan kompilasi.

## Contoh Format File Eksternal

### Format

```
<nama prajurit>
<jumlah bahan bakar awal>
<faktor konsumsi bahan bakar>
<koordinat  $x_1$ , koordinat  $y_1$ , koordinat  $z_1$ >
<koordinat  $x_2$ , koordinat  $y_2$ , koordinat  $z_2$ >
...
...
<koordinat  $x_{n-1}$ , koordinat  $y_{n-1}$ , koordinat  $z_{n-1}$ >
```

### data1.csv

```
Eren Iyenga
37.5
2.4
1.5,2.5,3.5
4.5,5.5,4.5
7.5,8.5,1.5
10.4,11.5,2.5
```

## Contoh Eksekusi Program (garis bawah menandakan input)

### #1

```
Masukkan nama file: data1.csv
Masukkan nama file: dataPalsu.csv
File tidak tersedia, silahkan coba lagi
Masukkan nama file: data2.csv
Masukkan nama file: exit
```

```
Prajurit Eren Iyenga
Bahan bakar awal: 37.50
Konsumsi bahan bakar: 2.40 km/liter
Jarak tempuh: 34.11 km
Sisa bahan bakar: 23.29 liter
```

```
Prajurit Oleksandr Kostyliev
Bahan bakar awal: 11.20
Konsumsi bahan bakar: 1.65 km/liter
Jarak tempuh: 87.74 km
Bahan bakar tidak cukup, tambah sebanyak 41.98 liter
```

## #2

Masukkan nama file: data2.csv

Masukkan nama file: data4.csv

Masukkan nama file: data5.csv

Masukkan nama file: exit

Prajurit Oleksandr Kostyliev

Bahan bakar awal: 11.20

Konsumsi bahan bakar: 1.65 km/liter

Jarak tempuh: 87.74 km

Bahan bakar tidak cukup, tambah sebanyak 41.98 liter

Prajurit Valerii Vakhovskyi

Bahan bakar awal: 25.24

Konsumsi bahan bakar: 3.55 km/liter

Jarak tempuh: 121.17 km

Bahan bakar tidak cukup, tambah sebanyak 8.89 liter

Prajurit Zelenskyy

Bahan bakar awal: 11.10

Konsumsi bahan bakar: 3.11 km/liter

Jarak tempuh: 57.34 km

Bahan bakar tidak cukup, tambah sebanyak 7.34 liter

## Soal 2

Diberikan suatu matriks kompleks, anda harus menyimpan input tersebut pada sebuah matriks dari *struct* pada C (sudah diberikan template *struct* pada program). Kemudian anda diminta untuk mengecek apakah matriks tersebut merupakan matriks hermitian atau bukan.

### Format input:

- Baris pertama input merupakan bilangan  $b$  dan  $k$  dimana  $b$  adalah jumlah baris dan  $k$  adalah jumlah kolom.
- Kemudian terdapat  $b$  baris input, masing-masing baris terdapat  $2k$  bilangan, menunjukkan komponen real dan imajiner isi matriks pada suatu baris

### Format output:

- Cetak "Matriks merupakan matriks hermitian" jika matriks input merupakan matriks hermitian
- Cetak "Matriks merupakan matriks skew-hermitian" jika matriks input merupakan matriks skew-hermitian
- Cetak "Matriks bukan merupakan matriks hermitian maupun matriks skew-hermitian" jika matriks bukan merupakan keduanya
- Cetak "Matriks merupakan matriks hermitian dan matriks skew-hermitian" jika matriks merupakan keduanya.

**Matriks Hermitian** (atau self-adjoint matrix) adalah matriks kompleks  $A$  yang bernilai sama dengan konjugat transposenya (disimbolkan dengan  $A^H$ ). Konjugat transpose dari sebuah matriks  $A$  yang berukuran  $m * n$  adalah matriks berukuran  $n * m$  yang diperoleh dengan mengambil konjugat kompleks dari transpose matriks  $A$ . Secara matematis, matriks hermitian adalah matriks yang memenuhi  $A = A^H$ .

Disisi lain, **matriks Skew Hermitian** adalah matriks yang nilai konjugat transposenya merupakan negatif dari matriks tersebut. Secara matematis, matriks skew hermitian adalah matriks yang memenuhi  $A = -A^H$ .

**Sebagai contoh**, diberikan matriks  $2 * 3$

$$A = \begin{pmatrix} 1 & -2-i & 5 \\ 1+i & i & 4-2i \end{pmatrix}$$

Transpose dari matriks di atas adalah

$$A^T = \begin{pmatrix} 1 & 1+i \\ -2-i & i \\ 5 & 4-2i \end{pmatrix}$$

Konjugat transpose matriks A diperoleh dengan melakukan konjugat kompleks pada matriks di atas

$$A^H = \begin{pmatrix} 1 & 1-i \\ -2+i & -i \\ 5 & 4+2i \end{pmatrix}$$

Karena  $A \neq A^H$  ataupun  $A \neq -A^H$ , matriks A bukan merupakan matriks hermitian ataupun skew hermitian.

**Contoh lain :** Diketahui matriks B

$$B = \begin{pmatrix} 3 & 3-2i \\ 3+2i & 2 \end{pmatrix}$$

Transpose dari matriks di atas adalah

$$B^T = \begin{pmatrix} 3 & 3+2i \\ 3-2i & 2 \end{pmatrix}$$

Konjugat kompleks dari matriks  $B^T$  adalah

$$B^H = \begin{pmatrix} 3 & 3-2i \\ 3+2i & 2 \end{pmatrix}$$

Karena  $B = B^H$ , matriks B disebut matriks hermitian.

**Contoh ketiga :**

Diketahui matriks C

$$C = \begin{pmatrix} -i & 2+i \\ -2+i & 0 \end{pmatrix}$$

Transpose dari matriks di atas adalah

$$C^T = \begin{pmatrix} -i & -2+i \\ 2+i & 0 \end{pmatrix}$$

Konjugat kompleks dari matriks  $C^T$  adalah

$$C^H = \begin{pmatrix} i & -2-i \\ 2-i & 0 \end{pmatrix}$$

Karena  $C = -C^H$ , matriks C disebut matriks hermitian-skew.

### Contoh Eksekusi Program (garis bawah menandakan input)

**#1**

2 3

1 0 -2 -1 5 0

1 1 0 1 4 -2

Matriks bukan merupakan matriks hermitian maupun matriks skew-hermitian

**#2**

2 2

3 0 3 -2

3 2 2 0

Matriks merupakan matriks hermitian

**#3**

2 2

0 1 -2 -1

2 -1 0 0

Matriks merupakan matriks skew-hermitian