



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)
Tahun / Semester : 2020-2021 / Genap
Modul : 4 - *Structures and Dynamic Arrays*
Hari, Tanggal Praktikum : Jumat, 19 Maret 2021

Naskah Soal Praktikum

Pembuat Naskah: Nur Sekti Waskitha Jati, Danu Ihza Pamungkas

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 *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. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan mengapa bukan soal yang lainnya!
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

Tanjirou Kamado sedang belajar mengenai proyeksi ortogonal. Proyeksi ortogonal terdiri dari dua jenis, yaitu proyeksi skalar ortogonal dan proyeksi vektor ortogonal. Perhitungan proyeksi tersebut menggunakan metode-metode perhitungan vektor, seperti

- Dot product

$$\begin{bmatrix} A_{1x} & A_{1y} & A_{1z} \end{bmatrix} \cdot \begin{bmatrix} A_{2x} & A_{2y} & A_{2z} \end{bmatrix} = A_{1x}A_{2x} + A_{1y}A_{2y} + A_{1z}A_{2z}$$

- Cross product

$$\begin{bmatrix} A_{1x} & A_{1y} & A_{1z} \end{bmatrix} \times \begin{bmatrix} A_{2x} & A_{2y} & A_{2z} \end{bmatrix} = (A_{1y}A_{2z} - A_{1z}A_{2y})i + (A_{1z}A_{2x} - A_{1x}A_{2z})j + (A_{1x}A_{2y} - A_{1y}A_{2x})z$$

- Panjang vektor

$$\vec{a} = |\vec{a}| = \left| \begin{bmatrix} A_{1x} & A_{1y} & A_{1z} \end{bmatrix} \right| = \sqrt{A_{1x}^2 + A_{1y}^2 + A_{1z}^2}$$

Proyeksi skalar ortogonal vektor \vec{a} terhadap vektor \vec{b} dirumuskan sebagai $\frac{\vec{a} \cdot \vec{b}}{|\vec{b}|}$, sementara proyeksi vektor

ortogonal vektor \vec{a} terhadap vektor \vec{b} dirumuskan sebagai $\frac{\vec{a} \cdot \vec{b}}{|\vec{b}|^2} \vec{b}$. Karena Tanjirou suka iseng, dia mencoba

membuat fungsi F yang akan mencari proyeksi vektor ortogonal secara terus menerus hingga berhenti sebelum vektor terakhir. Setelah itu, dilakukan operasi cross product pada 2 vektor terakhir yang tersisa.

$$F = (A_1 \circ A_2 \circ A_3 \circ \dots \circ A_{n-1}) \times A_n$$

Asumsikan $(\vec{a} \circ \vec{b})$ memiliki arti proyeksi vektor orthogonal \vec{a} terhadap vektor \vec{b} .

Misal jika masukan vektor adalah A, B, C, dan D, maka fungsi F akan bernilai

$$F = (A \circ B \circ C) \times D$$

Karena Tanjirou tidak tahu berapa matriks yang akan dilakukan perhitungan, maka program harus menerima vektor dengan jumlah yang tidak ditentukan sebelumnya. Asumsikan Tanjirou menginput minimal 2 vektor. Vektor hasil fungsi F dicetak dengan dua angka di belakang koma saja, termasuk angka 0.

Contoh eksekusi program (garis bawah menunjukkan input)

#1

Vektor 1

Masukkan komponen i: 2

Masukkan komponen j: 1

Masukkan komponen k: 4

Vektor 1 : $2.00i + 1.00j + 4.00k$

Masukkan vektor lagi? [y/n]: y

Vektor 2

Masukkan komponen i: 6

Masukkan komponen j: 3

Masukkan komponen k: 4

Vektor 2 : $6.00i + 3.00j + 4.00k$

Masukkan vektor lagi? [y/n]: n

Menghitung fungsi F.....

Vektor hasil fungsi F = $- 8.00i + 16.00j + 0.00k$

#2

Vektor 1

Masukkan komponen i: 1

Masukkan komponen j: 1

Masukkan komponen k: 1

Vektor 1 : $1.00i + 1.00j + 1.00k$

Masukkan vektor lagi? [y/n]: y

Vektor 2

Masukkan komponen i: 2

Masukkan komponen j: 2

Masukkan komponen k: 2

Vektor 2 : $2.00i + 2.00j + 2.00k$

Masukkan vektor lagi? [y/n]: y

Vektor 3

Masukkan komponen i: 7

Masukkan komponen j: 8

Masukkan komponen k: 9

Vektor 3 : $7.00i + 8.00j + 9.00k$

Masukkan vektor lagi? [y/n]: n

Menghitung fungsi F.....

Vektor hasil fungsi F = $1.00i - 2.00j + 1.00k$

#3

Vektor 1

Masukkan komponen i: 3

Masukkan komponen j: 4
Masukkan komponen k: 5
Vektor 1 : $3.00i + 4.00j + 5.00k$
Masukkan vektor lagi? [y/n]: y

Vektor 2
Masukkan komponen i: -3
Masukkan komponen j: 4
Masukkan komponen k: -5
Vektor 2 : $- 3.00i + 4.00j - 5.00k$
Masukkan vektor lagi? [y/n]: y

Vektor 3
Masukkan komponen i: 4
Masukkan komponen j: 3
Masukkan komponen k: -5
Vektor 3 : $4.00i + 3.00j - 5.00k$
Masukkan vektor lagi? [y/n]: y

Vektor 4
Masukkan komponen i: 1
Masukkan komponen j: 1
Masukkan komponen k: 0
Vektor 4 : $1.00i + 1.00j + 0.00k$
Masukkan vektor lagi? [y/n]: y

Vektor 5
Masukkan komponen i: 5
Masukkan komponen j: 5
Masukkan komponen k: 5
Vektor 5 : $5.00i + 5.00j + 5.00k$
Masukkan vektor lagi? [y/n]: n

Menghitung fungsi F.....
Vektor hasil fungsi F = $- 3.15i + 3.15j + 0.00k$

Soal 2

Pak Raif adalah dosen mata kuliah PAR sekaligus Kaprodi. Pak Raif tidak memiliki cukup waktu untuk menginput nilai dan mengurutkannya karena sibuk bermain Dota 2. Pak Raif berinisiatif untuk membuat program untuk mengurutkan data nilai mahasiswa. Data yang disimpan dalam program tersebut yaitu:

- Nama (maksimal 20 karakter)
- NIM (8 digit)
- Nilai (integer 0 - 100)

Setelah program menerima input jumlah mahasiswa, program melakukan *dynamic memory allocation* untuk membuat besar array sesuai dengan jumlah mahasiswa.

Asumsi jumlah minimal mahasiswa adalah 2 orang. Pak Arif juga dapat mencari nilai mahasiswa dengan keyword nama ataupun NIM secara bersamaan. Hasil pencarian bisa lebih dari 1 hasil. Data tersebut juga dapat diurutkan berdasarkan nama, NIM, dan nilai. Pengurutan nilai dimulai dari yang terbesar.

Struct yang digunakan dapat dideklarasikan sebagai berikut:

```
struct student {  
    char nama[20];  
    int NIM;  
    int nilai;  
};
```

Contoh eksekusi program (garis bawah menandakan input)

```
#1  
Masukkan jumlah mahasiswa: 3  
Masukkan nama, NIM, dan nilai:  
danu 13218022 100  
andika 13218030 99  
elang 13218041 98  
  
Masukkan Nama atau NIM yang ingin dicari:  
dan  
  
Hasil pencarian:  
danu      13218022      100  
  
Hasil sorting berdasarkan nama:  
andika 13218030      99  
danu    13218022      100  
elang   13218041      98
```

Hasil sorting berdasarkan NIM:

danu	13218022	100
andika	13218030	99
elang	13218041	98

Hasil sorting berdasarkan nilai:

danu	13218022	100
andika	13218030	99
elang	13218041	98

#2

Masukkan jumlah mahasiswa: 3

Masukkan nama, NIM, dan nilai:

danu 13218022 98

elang 13218041 100

andika 13218030 99

Masukkan Nama atau NIM yang ingin dicari:

022

Hasil pencarian:

danu	13218022	98
------	----------	----

Hasil sorting berdasarkan nama:

andika	13218030	99
danu	13218022	98
elang	13218041	100

Hasil sorting berdasarkan NIM:

danu	13218022	98
andika	13218030	99
elang	13218041	100

Hasil sorting berdasarkan nilai:

elang	13218041	100
andika	13218030	99
danu	13218022	98

#3

Masukkan jumlah mahasiswa: 3

Masukkan nama, NIM, dan nilai:

danu 13218022 99

elang 13218041 99

andika 13218030 100

Masukkan Nama atau NIM yang ingin dicari:

13218

Hasil pencarian:

danu	13218022	99
elang	13218041	99
andika	13218030	100

Hasil sorting berdasarkan nama:

andika	13218030	100
danu	13218022	99
elang	13218041	99

Hasil sorting berdasarkan NIM:

danu	13218022	99
andika	13218030	100
elang	13218041	99

Hasil sorting berdasarkan nilai:

andika	13218030	100
danu	13218022	99
elang	13218041	99