



Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)

Tahun / Semester : 2021-2022 / Genap

Modul : 7 - Stacks and Queues

Hari, Tanggal Praktikum : Rabu, 6 April 2022

Naskah Soal Praktikum

Pembuat Naskah: Gala Ramadhan Ilham Warno Putra, Hudzaifah Afif Al Fatih Nasution

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

Pak Justin adalah seorang guru komputer SD. Dan ingin memberikan muridnya sebuah permainan sederhana. Dimana muridnya sebagai pengguna dapat memilih untuk memasukan angka atau menghapus angka. Anak-anak sangat antusias dengan hal tersebut. Akan tetapi karena Pak Justin mengalami kelelahan setelah menjalani *tour* ke berbagai tempat, Pak Justin terlupa hingga baru tersadar bahwa besok Pak Justin harus membawakan permainan tersebut ke anak-anak. Karena terburu-buru Pak Justin menjadi mengalami kesulitan dalam membuat program tersebut. Pak Justin berencana akan menggunakan *queue*.

Tugas Anda :

Bantu Pak Justin untuk dapat membuat program tersebut. Yaitu suatu program dengan menggunakan *queue*. Dengan spesifikasi sebagai berikut.

Hint : Anda dapat menggunakan linked list jika anda rasa diperlukan.

Hal yang perlu Anda perhatikan :

- Terdapat **3 pilihan** menu yang dapat dipilih. Pilihan 1 untuk menghapus data, pilihan 2 untuk menghapus data terdepan atau paling awal dimasukan, pilihan 99 untuk mengakhiri program.
- **Asumsi**kan pilihan menu yang dimasukan oleh pengguna selalu benar. Yaitu hanya antara 1 atau 2 atau 99 saja.
- **Angka paling depan** adalah angka yang dimasukan pertama kali. Sedangkan **angka paling belakang** adalah angka yang dimasukan terakhir kali.
- Proses **hapus** akan menghapus angka paling depan, sehingga akan membuat angka terdepan menjadi digantikan dengan angka setelahnya, sedangkan angka paling belakang adalah tetap.
- Pengguna dapat memasukan **angka berapapun**. Dari bilangan **positif, nol, hingga negatif**.
- **Asumsi**kan pengguna memasukan data minimal 1 kali atau menyisakan data minimal 1 buah. Artinya asumsikan saja tidak mungkin tidak ada data yang tersisa atau dengan kata lain minimal data yang tersisa adalah 1 buah.
- Jika hanya terdapat 1 data masukan atau hanya bersisa 1 data masukan (masukan oleh pengguna sebanyak n , namun dihapus sebanyak $n-1$), maka nilai data paling depan dan paling belakang adalah sama.
- **Asumsi**kan pengguna selalu memasukan pilihan 2, yaitu menambahkan data, pada pilihan pertama.
- Untuk memberikan gambaran perhatikan ilustrasi berikut

Struktur :

Depan	...	Belakang
-------	-----	----------

Pengguna : tambah nilai 1

1

Output :

Angka Paling Depan : 1

Angka Paling Belakang : 1

Pengguna : tambah nilai 2

1	2
---	---

Output :

Angka Paling Depan : 1

Angka Paling Belakang : 2

Pengguna : tambah nilai 3

1	2	3
---	---	---

Output :

Angka Paling Depan : 1

Angka Paling Belakang : 3

Pengguna : hapus

2	3
---	---

Output :

Angka Paling Depan : 2

Angka Paling Belakang : 1

Pengguna : hapus

3

Output :

Angka Paling Depan : 3

Angka Paling Belakang : 3

Contoh Eksekusi Program (garis bawah menandakan input)

#1

Pilihan Menu :

- 1. Hapus
- 2. Tambah
- 99. Akhiri

Masukan Pilihan: 2

Masukan angka : 1

Masukan Pilihan: 2

Masukan angka : 2

Masukan Pilihan: 2

Masukan angka : 3

Masukan Pilihan: 99

Angka Paling Depan : 1

Angka Paling Belakang : 3

#2

Pilihan Menu :

- 1. Hapus
- 2. Tambah
- 99. Akhiri

Masukan Pilihan: 2

Masukan angka : 1

Masukan Pilihan: 2

Masukan angka : 2

Masukan Pilihan: 2

Masukan angka : 3

Masukan Pilihan: 1

1 Data terhapus

Masukan Pilihan: 99

Angka Paling Depan : 2

Angka Paling Belakang : 3

#3

Pilihan Menu :

- 1. Hapus
- 2. Tambah
- 99. Akhiri

Masukan Pilihan: 2

Masukan angka : 101

Masukan Pilihan: 2

Masukan angka : 102

Masukan Pilihan: 2

Masukan angka : 103

Masukan Pilihan: 1

1 Data terhapus

Masukan Pilihan: 1

1 Data terhapus

Masukan Pilihan: 99
Angka Paling Depan : 103
Angka Paling Belakang : 103

Soal 2

Instruction register adalah tempat komputer menyimpan instruksi dari sebuah program. Dalam soal ini akan dibuat *instruction register* yang akan membaca instruksi dari sebuah file.txt. Apabila nama file tidak ditemukan maka program akan mengembalikan *error message* “File tidak bisa dibuka\n” kemudian program menerima *input* nama *file* kembali sampai *valid*.

Program akan menerima nomor *line* dari pengguna dan mengembalikan perintah dari *line* tersebut. Nomor *line* pada program dimulai dari 1. Apabila nomor *line* yang diberikan tidak *valid* (nomor *line* lebih besar dari jumlah *line* pada program atau nomor *line* kurang dari 1) maka program akan mengembalikan *error message* “Line tidak valid\n” kemudian program menerima *input* nomor *line* kembali sampai *valid*.

Program kemudian akan mengembalikan *line* selanjutnya yang akan dieksekusi setelah *line* tersebut. Defaultnya *line* selanjutnya adalah *line* setelah nomor *line input* pengguna ditambah 1 kecuali apabila terdapat perintah `jump` pada *line* tersebut, maka program akan berpindah ke *line* sesuai dengan angka yang mengikuti perintah `jump`. Apabila *line* nomor *line* adalah *line* terakhir dari *file* maka program akan mengembalikan angka 0

Instruction register harus diimplementasikan dengan *stack* atau *queue*.

Contoh file kode yang dibaca

code.txt

```
terima input integer 1
cetak integer 1
jump 1
terima input integer 2
jumlahkan integer 1 ke 2
cetak integer 2
jump 3
keluar program
```

Contoh Eksekusi Program (garis bawah menandakan input)

```
#1
Nama file yang akan dibaca: code.txt
Line yang ingin dibaca: 6
```

Line 6: cetak integer 2
Line selanjutnya: 7

#2

Nama file yang akan dibaca: code.txt
Line yang ingin dibaca: 7
Line 7: jump 3
Line selanjutnya: 3

#3

Nama file yang akan dibaca: codong
File tidak bisa dibuka
Nama file yang akan dibaca: code.txt
Line yang ingin dibaca: 10
Line tidak valid
Line yang ingin dibaca: 2
Line 2: cetak integer 1
Line selanjutnya: 3

#4

Nama file yang akan dibaca: code.txt
Line yang ingin dibaca: 13
Line tidak valid
Line yang ingin dibaca: 8
Line 8: keluar program
Line selanjutnya: 0