



## Program Studi Teknik Elektro ITB

Nama Kuliah (Kode) : Praktikum Pemecahan Masalah dengan C (EL2208)  
Tahun / Semester : 2023-2024 / Genap  
Modul : 2 – Pointer & Function  
Hari, Tanggal Praktikum : Kamis, 7 Maret 2024

---

### Naskah Soal Praktikum

Pembuat Naskah: Agape D'sky, Eunike Kristianti

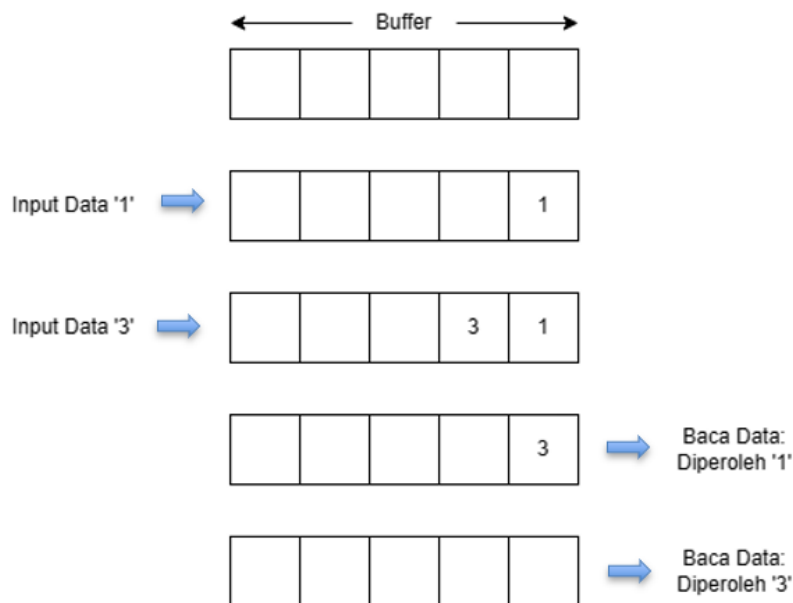
#### Ketentuan:

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 praktikum.stei.itb.ac.id paling lambat pukul 11.00 WIB dua hari kerja setelah sesi praktikum. Tulis pula alasan anda memilih mengerjakan soal yang anda kerjakan dan penjelasan/rancangan dari algoritma yang Anda gunakan!
5. Solusi soal pertama dan kedua harus dapat dikompilasi dengan perintah `make main` dan menghasilkan *file executable* dengan nama `main`.
6. Bila diperlukan, sesuaikanlah isi *Makefile* yang tersedia pada *template repository* untuk memenuhi syarat kompilasi dan *file* keluaran di atas!

## Soal 1

Dalam sistem yang kompleks, manajemen memori merupakan salah satu hal yang krusial untuk diperhatikan. Setiap alamat memori harus di-*track* supaya tidak terjadi *memory leaking*. Untuk itu, terkadang, suatu informasi disimpan pada alamat yang statis, supaya tidak mempersulit pemrograman. Dalam soal ini, akan ditunjukkan bagaimana cara membuat struktur penyimpanan *buffer* yang efisien dan mudah dilacak kondisinya.

Misalkan, suatu *buffer* didefinisikan dengan ukuran 5. *Buffer* ini ingin digunakan untuk menyimpan data secara sementara. Spesifikasi dari *buffer* data yang diinginkan adalah:

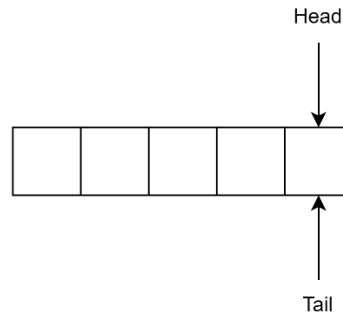


**Gambar 1 Visualisasi Buffer**

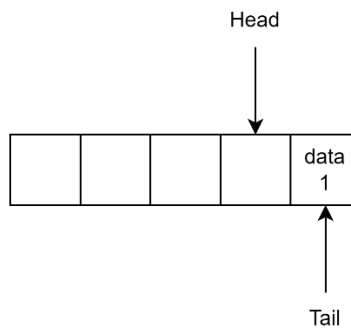
- Data yang masuk lebih dulu akan keluar lebih dulu juga ketika *buffer* dibaca (*first in, first out*),
- Apabila data dalam *buffer* sudah 5, *buffer* tidak diperbolehkan menampung lebih banyak data (penulisan data akan ditolak),
- Apabila data dalam *buffer* kosong, tidak diperbolehkan melakukan pembacaan data dari *buffer*.

Untuk meningkatkan efisiensi *buffer*, implementasi *buffer* akan dilakukan dengan menggunakan sebuah *array of integer* statis. Cara kerja dari implementasi ini adalah sebagai berikut:

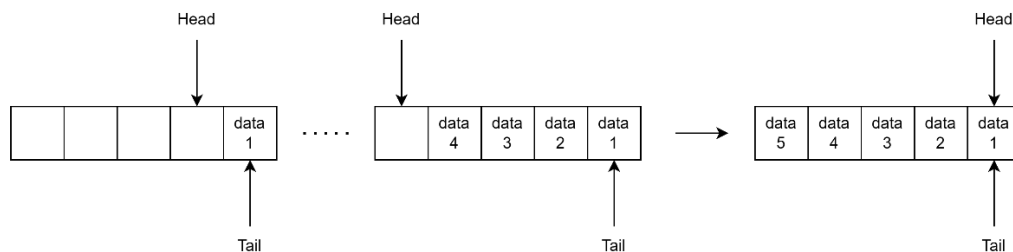
- Ketika *array* dideklarasikan (`int arr[5]`), terdapat 2 buah variabel tambahan yang diperlukan, yaitu **head** dan **tail**. **Head** adalah penunjuk alamat (*pointer*) informasi paling baru di *buffer*, **tail** adalah penunjuk alamat (*pointer*) informasi paling usang di *buffer*. Perhatikan bahwa **head** dan **tail** adalah **POINTER TO INTEGER!**



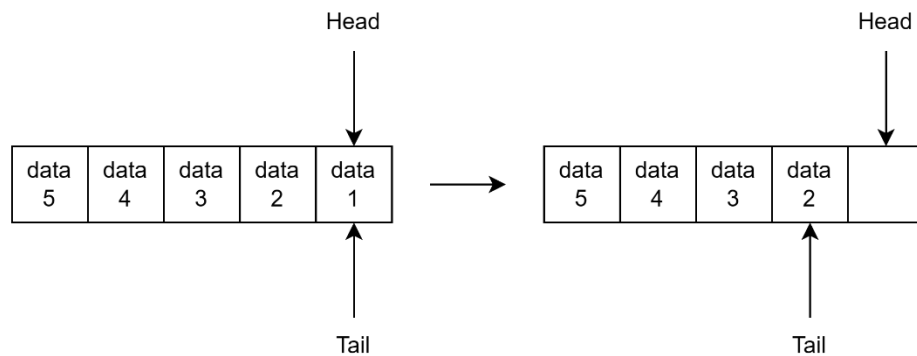
- Ketika data baru masuk ke dalam *buffer*, maka variabel **head** akan ikut bergeser.



- Misalkan data baru masuk secara terus-menerus sampai *buffer* penuh, maka **head** akan bergeser terus-menerus sampai kembali ke posisi seperti gambar di bawah. Perhatikan bahwa posisi **head** akan kembali sama dengan **tail** dalam dua kondisi, yakni ketika *buffer* penuh dan kosong.



- Setiap kali data dibaca dari *buffer*, **tail** akan bergeser supaya selalu menunjuk data yang paling usang.



Dalam rangka menyelesaikan rancangan implementasi ini, praktikan diminta untuk melengkapi fungsi untuk memasukkan data (**writeData**), membaca data (**readData**), dan fungsi untuk menuliskan isi buffer (**printBuffer**). Praktikan cukup melengkapi rangka kode yang sudah dibuat dengan memanfaatkan **POINTER**.

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

Buffer kosong sudah dibuat.

Input test case: 1

Perlakuan: write(1), write(2), write(3)

Hasil read data: -

Isi buffer: 1 2 3

Lokasi head: + 3

Lokasi tail: + 0

Buffer kosong sudah dibuat.

Input test case: 2

Perlakuan: write(1), write(2), write(3), read(), read()

Hasil read data: 1 2

Isi buffer: 3

Lokasi head: + 3

Lokasi tail: + 2

Buffer kosong sudah dibuat.

Input test case: 3

Perlakuan: write(1), read(), write(2), write(3), write(10)

Hasil read data: 1

Isi buffer: 2 3 10

Lokasi head: + 4

Lokasi tail: + 1

Buffer kosong sudah dibuat.

Input test case: 4

Perlakuan: write(1), write(3), write(5), write(6), write(7), read()

Hasil read data: 1

Isi buffer: 3 5 6 7

Lokasi head: + 0

Lokasi tail: + 1

## Soal 2

Tommy merupakan seorang anak yang pandai dalam komputasi. Ia ingin membantu ibunya yang merupakan seorang pedagang agar tidak perlu lagi menghitung belanjaan dan juga kembalian secara manual. Program menerima input berupa jumlah barang-barang yang diinginkan pembeli dan uang yang dibawa oleh pembeli, kemudian menampilkan jumlah belanjaan dan juga kembalian yang perlu diberikan oleh Ibu.

Notes:

- Asumsi input selalu valid
- List barang dan juga harga tiap barang ada pada template *source code* program
- Bila total belanjaan lebih dari uang yang dibawa pembeli, program akan memberitahu bahwa uang tidak cukup
- Bila total belanjaan sama dengan uang yang dibawa pembeli, program akan memberitahu bahwa tidak ada kembalian.
- Program harus menggunakan pointer dan fungsi
- Kembalian yang dapat digunakan adalah:
  - Lembaran 100.000
  - Lembaran 50.000
  - Lembaran 20.000
  - Lembaran 10.000
  - Lembaran 5.000
  - Lembaran 2.000

- Lembaran 1.000
- Koin 500
- Koin 200
- Koin 100

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

**#1**

List barang:

1. Beras (Rp12.500)
2. Daging (Rp21.700)
3. Sayur (Rp3.900)
4. Bumbu (Rp4.600)
5. Sambal (Rp2.200)

Jumlah barang secara berurutan: 2 1 5 3 2

Uang yang dibawa pembeli: 200000

Total belanjaan: 84400

Kembalian:

- 1 lembar uang 100.000
- 1 lembar uang 10.000
- 1 lembar uang 5.000
- 1 lembar uang 500
- 1 lembar uang 100

**#2**

List barang:

1. Beras (Rp12.500)
2. Daging (Rp21.700)
3. Sayur (Rp3.900)
4. Bumbu (Rp4.600)
5. Sambal (Rp2.200)

Jumlah barang secara berurutan: 5 2 0 0 0

Uang yang dibawa pembeli: 12000

Total belanjaan: 105900

Uang tidak cukup!

**#3**

List barang:

1. Beras (Rp12.500)
2. Daging (Rp21.700)
3. Sayur (Rp3.900)
4. Bumbu (Rp4.600)
5. Sambal (Rp2.200)

Jumlah barang secara berurutan: 2 1 5 3 2

Uang yang dibawa pembeli: 84400

Total belanjaan: 84400

Uang pas. Tidak ada kembalian.