Ujian Tengah Semester

STMIK AMIK Bandung

Pramudya Arya Wicaksana 2242805

Contents

1 Pertanyaan pertama **3**
1.1 Soal . . . 3
1.2 Jawab . . . 3

2 Pertanyaan kedua **3**
2.1 Soal . . . 3
2.2 Jawab . . . 3

3 Pertanyaan ketiga **3**
3.1 Soal . . . 3
3.2 Jawab . . . 3
3.2.1 Array . . . 3
3.2.2 Single linked list . . . 4
3.2.3 Double linked list . . . 4
3.2.4 Stack . . . 4
3.2.5 Queue . . . 5

3.2.6 Dequeue 5

4 Pertanyaan keempat **5**
4.1 Soal 5
4.2 Jawab 5

5 Pertanyaan kelima **6**
5.1 Soal 6
5.2 Jawab 6

1 Pertanyaan pertama

1.1 Soal

Jelaskan mengenai Struktur Data, Manfaat, Jenis, berikut contoh Struktur Data ?

1.2 Jawab

Struktur data merupakan sebuah format untuk mengorganisasi data, mengolah data, menerima dan menyimpan data, manfaatnya kita bisa mengelola data lebih baik sehingga bisa mencapai time complexity $O(N^2)$, jenis dari data struktur dikategorikan menjadi beberapa jenis, beberapa tipe data yang wajib diketahui oleh Software Engineer adalah

1. Heap
2. Linkedlist
3. Stack
4. Queue
5. Binary tree
6. Array

2 Pertanyaan kedua

2.1 Soal

Bagaimana hubungan antara struktur data dengan algoritma dalam pembuatan program, jelaskan?

2.2 Jawab

Korelasi hubungan antara struktur data dengan algoritma adalah setiap algoritma pasti memiliki struktur data agar membuat sebuah algoritma tersebut menjadi efektif dan efisien, serta dapat memempercepat development sebuah aplikasi, seperti contoh jika ingin membuat algoritma Fibonacci, jika menggunakan binary tree akan menjadi mudah

3 Pertanyaan ketiga

3.1 Soal

Jelaskan mengenai struktur data array, Single Linked List, Double Linked List, Stack, Queue, deQueue serta gambarkan ilustrasi berikut contohnya ?

3.2 Jawab

3.2.1 Array

Array merupakan kumpulan data dari suatu tipe data

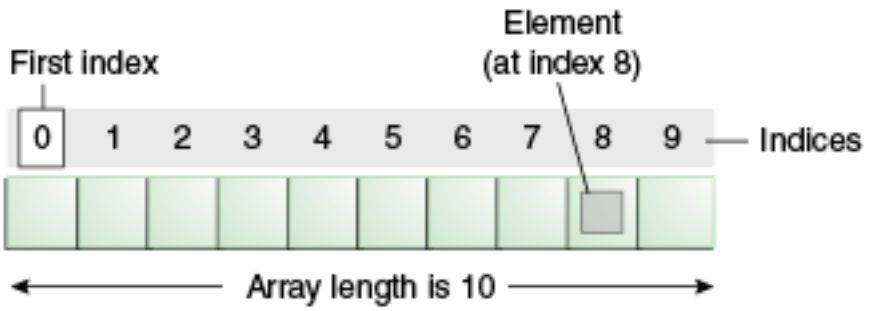

Figure 1: Array

Contoh penggunaan array adalah saat kita ingin mengelompokan suatu data

3.2.2 Single linked list

Single linked list merupakan sekumpulan data atau *Node* yang saling terhubung dengan node lain menggunakan pointer, single linked list hanya memiliki 1 arah, dari head ke tail (kiri ke kanan)

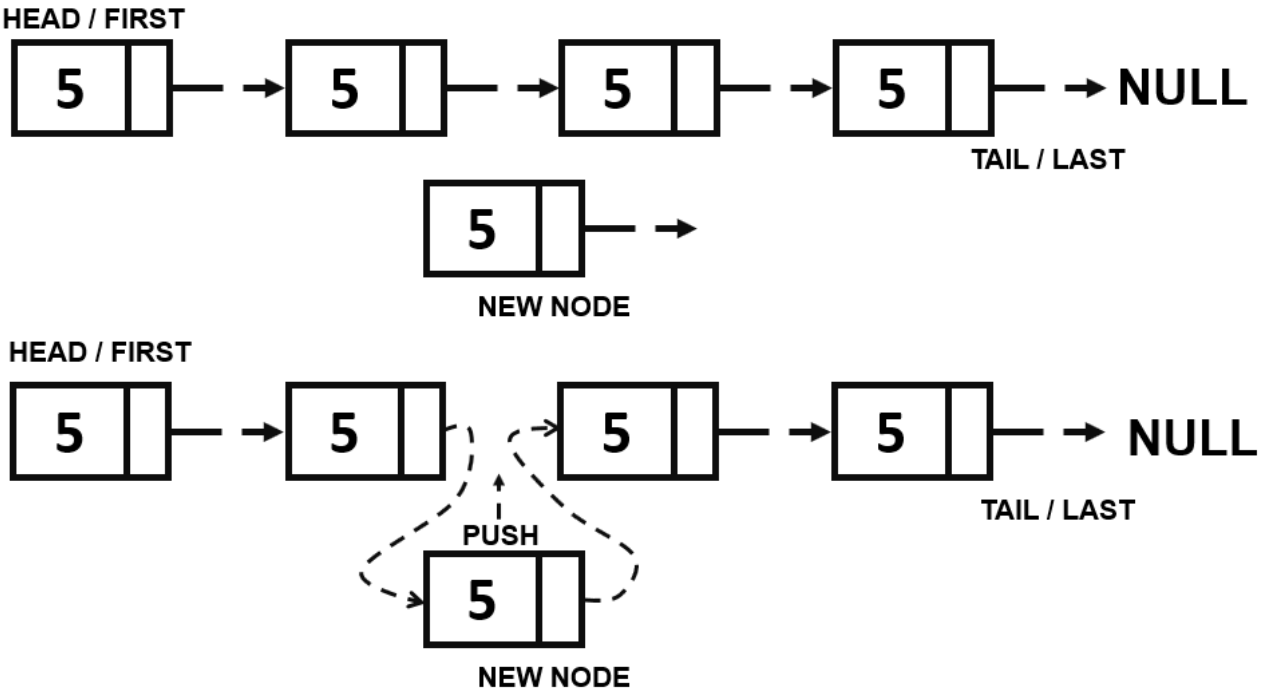

Figure 2: Single linked list

Contoh penggunaan single linked list adalah saat kita ingin implementasi Stack dan Queue

3.2.3 Double linked list

Double Linked List adalah suatu linked list yang mempunyai 2 penunjuk yaitu penunjuk ke simpul sebelumnya dan ke simpul berikutnya

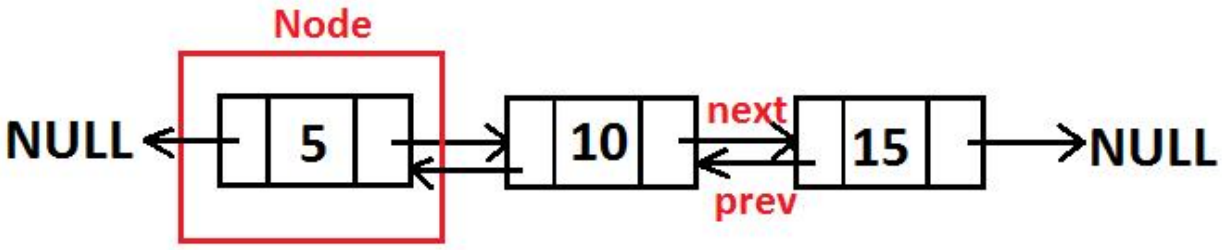

Figure 3: Double linked list

Contoh penggunaan double linked list adalah saat kita ingin implementasi Music player

3.2.4 Stack

Stack merupakan data struktur yang memiliki sifat **LIFO** atau Last In First Out, data pertama dari Stack akan masuk paling bawah lalu dilanjutkan dengan data berikutnya diatas data yang sudah dimasukan, jika ingin mengeluarkan data maka data yang terakhir dimasukan atau paling atas akan keluar pertama

Contoh penggunaan stack adalah saat kita menyusun sebuah Undo History

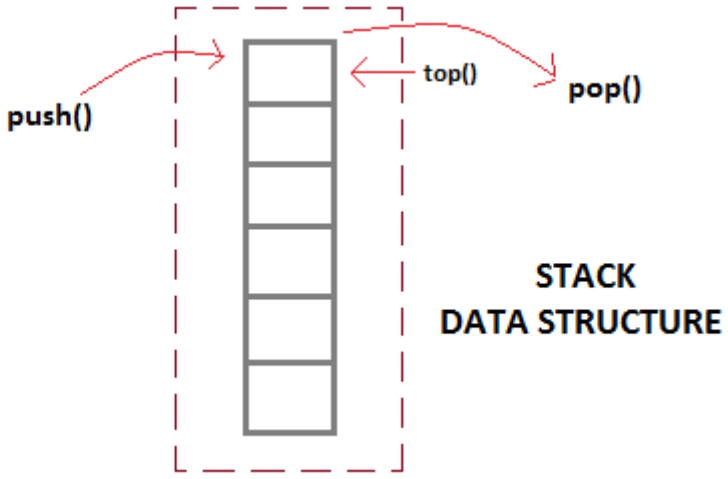

Figure 4: Stack

3.2.5 Queue

Stack merupakan data struktur yang memiliki sifat **FIFO** atau First In First Out, data pertama dari Stack akan masuk paling depan lalu dilanjutkan dengan data berikutnya dibelakang data yang sudah dimasukan, jika ingin mengeluarkan data maka data yang pertama dimasukan atau paling depan akan keluar pertama

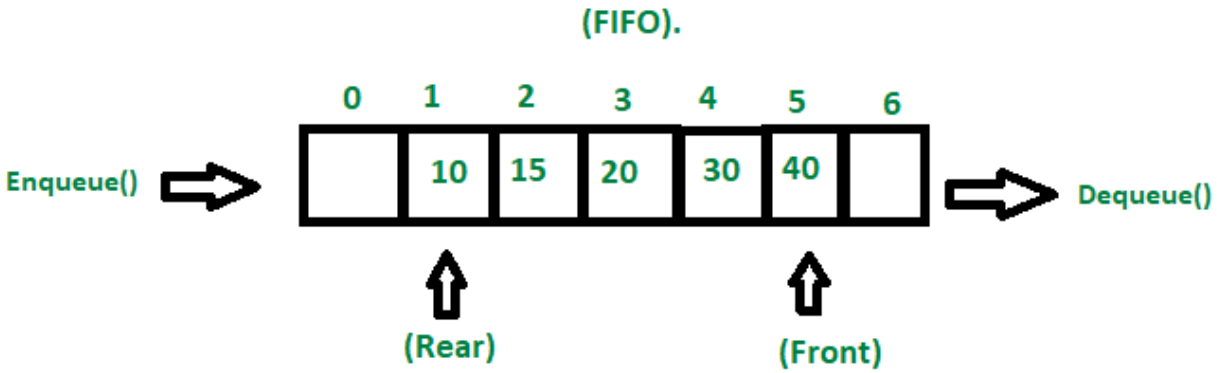

Figure 5: Queue

Contoh penggunakan queue adalah saat kita akan pub sub ke message broker

3.2.6 Dequeue

Dequeue merupakan data struktur linear yang memiliki 2 ujung, bedanya dengan Queue adalah jika queue hanya bisa memasukan dari satu arah maka dequeue bisa memasukan dan mengeluarkan data dari arah manapun depan maupun belakang

4 Pertanyaan keempat

4.1 Soal

Diketahui ekspresi infix seperti berikut : (05*((10-5)(2-1)(2+5))/2) Ubahlah ke dalam bentuk postfix dan hitunglah nilainya?

4.2 Jawab

Postfix := 05 10 5 - 2 1 - 2 5 + * 2 / Nilai := (((2-1)*(2+5))/2) = 3.5

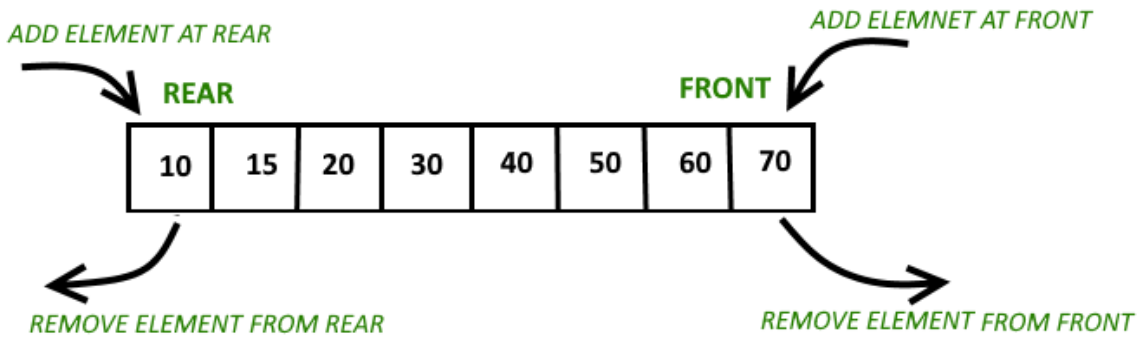

Figure 6: Dequeue

5 Pertanyaan kelima

5.1 Soal

Diketahui deret angka : 05 , 90, 67, 80, 101 Susun angka tersebut menjadi Priority Queue Ascending dan Lakukan operasi berikut : poll(), add(102), poll(), add(82)

5.2 Jawab

```
import java.util.*;

class QueuePriority {

    public static void main(String args[])
    {
        PriorityQueue<Integer> queue = new PriorityQueue<Integer>();

        queue.add(05);
        queue.add(90);
        queue.add(67);
        queue.add(80);
        queue.add(101);

        Integer val = null;
        while( (val = queue.poll()) != null) {
            System.out.println(val);
        }
        queue.poll();
        queue.add(102);
        queue.poll();
        queue.add(82);
        while( (val = queue.poll()) != null) {
            System.out.println(val);
        }
    }
}
```

```
~ > cd "/home/aya/Sandbox/Java-Projects/" && javac QueuePriority.java && java QueuePriority
5
67
80
90
101
82
~ >
```