**Nama : Rifqy Anugerah**

**Kelas : D3 RPLA 4505**

**NIM : 6706213095**

https://github.com/rifqyanugerah/Rifqy-Anugerah-Implementasi-Struktur-Data.git

**Pre-Assesment**

**1.** Pada akhir slide searching dan sorting

# a. Searching Linear

Lakukan pencarian linear search untuk X = 9 pada suatu array angka dengan isi array: {30, 4, 7, 10, 8, 100, 33}. Lakukan pencarian langkah per Langkah! **Jawab:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | X = 9 |
| arr | 30 | 4 | 7 | 10 | 8 | 100 | 33 |

**for**(**int** i=0;i<arr.length;i++){

**if**(arr[i] == x){

System.out.println(x+" nilai yang di cari ada pada indeks ke-“: "+ i);

}

Penjelasan

1. Akan dilakukan sebuah looping for, dimana range looping 0 - Panjang array yaitu 7, jadi akan dilakukan pengulangan sebanyak 7x
2. Dalam pengulangan itu x akan di cocokan di setiap nilai arr didalamnya, dimulai dari index 0 sampai selesai
3. Jika ditemukan akan menampilkan “nilai yang di cari ada pada indeks ke-“
4. Dan pada soal diatas tidak akan menampilkan apa apa, karena nilai x tidak terdapat pada array arr.

# b. Binary Seach

Lakukan pencarian binary search untuk X = 33 pada suatu array angka dengan isi array: {4, 7, 10, 8, 9, 30, 33, 100}. Lakukan pencarian langkah per Langkah! **Jawab:**

Sebelum di sort:

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **X =33** |
| arr | 4 | 7 | 10 | 8 | 9 | 30 | 33 | 100 |
| **Array.sort(arr);** Sesudah di sort: | |  |  |  |  |  |  |  |  |
| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | **X =33** |
| arr | 4 | 7 | 8 | 9 | 10 | 30 | 33 | 100 |

int first = 0; int

last=arr.length-1; int mid = (first + last)/2; while( first <= last ){ if ( arr[mid] < x ){ first = mid + 1;

}else if ( arr[mid] == x

){

System.out.println("Element is found at index: " + mid); break; }else{

last = mid - 1;

}

mid = (first + last)/2;

}

if ( first > last ){

System.out.println("Element is not found!");

}

}

Penjelasan:

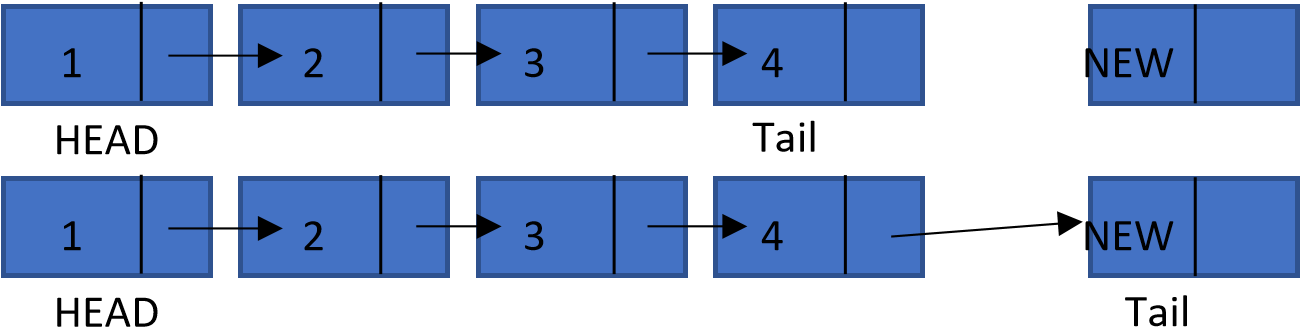
1. Pada kasus ini array diurutkan menggunakan sort terlebihdahulu, lalu saya menggunakan pengulangan while dimana dilakukan dengan kondisi Ketika first kurang dari sama dengan mid, dengan mid yang di dapat dari mencari Panjang array(length) lalu di bagi 2 dan first merupakan index awal dengan nilai 0.
2. Dalam pengulangan aka nada pengecekan berabang if, kondisi pengecekan pertama yaitu, jika nilai array pada index mid kurang dari x(nilai yang di cari), jika true maka posisi first akan berpindah menjadi selankah dari mid, jika false maka akan menjalankan kondisi if kedua
3. Pada kondisi if kedua akan ada pengecekan jika nilai array pada index mid sama dengan nilai x maka akan di lakukan menampilan "Element is found at index: " + nilai mid”
4. Jika kondisi keduanya false maka nilai last akan berubah menjadi nilai mid -1
5. Jika semua kondisi bernilai false maka akan menampilkan "Element is not found!"
6. Pada kasus ini pertama akan masuk kondisi kesatu, lalu first berubah menjadi mid+1(5), dan mid akan berubah lagi dari 4 menjadi 6, pada pengulangan kedua, akan masuk pada kondisi kedua yaitu mid sama dengan x nilai yang di cari, lalu system akan memunculkan **Element is found at index: 6**

**2.** Gambarkan dan jelaskan proses insert belakang dan tengah baik untuk Singly dan Doubly

Linked List

**a.** Singly Linked List

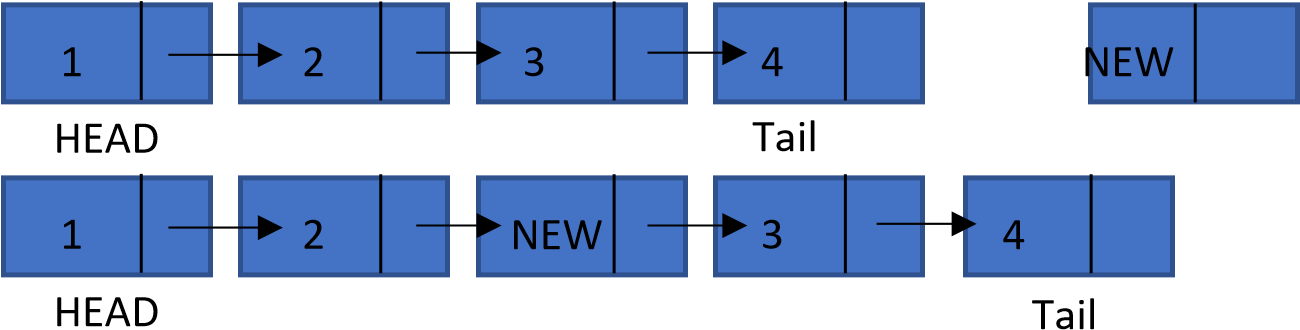
# i. Insert belakang



Penjelasan:

Node baru diinput dari blakang sehingga node baru menjadi tail atau nilai index akhir pada suatu array.

# ii. Insert tengah

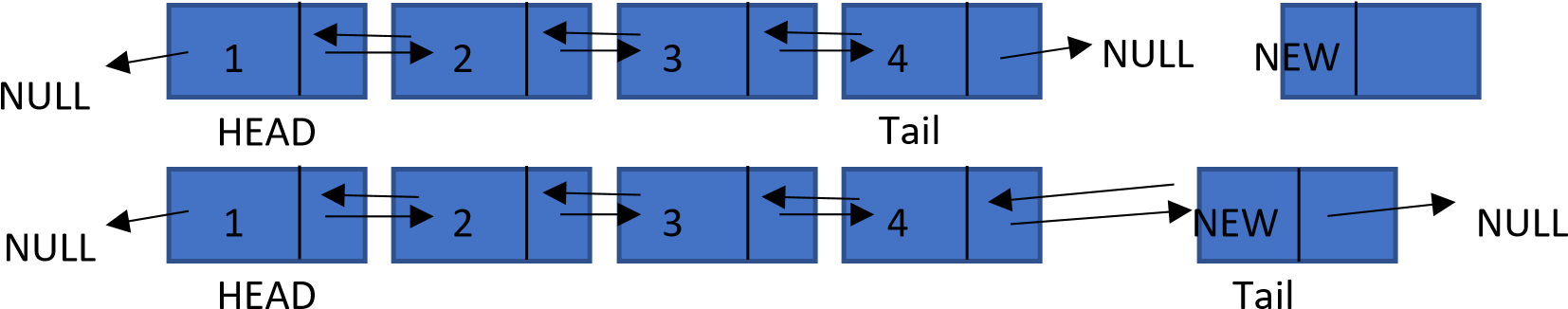


Penjelasan:

Untuk menambahkan data baru di tengah kita harus mendapkan nilai mid disebuah array terdahulu dengan cara arr.lenght/2, lalu node baru akan di insert pada tengah dengan cara bergeser dari 1, ke 2, lalu ketengah.

**b.** Double Linked List

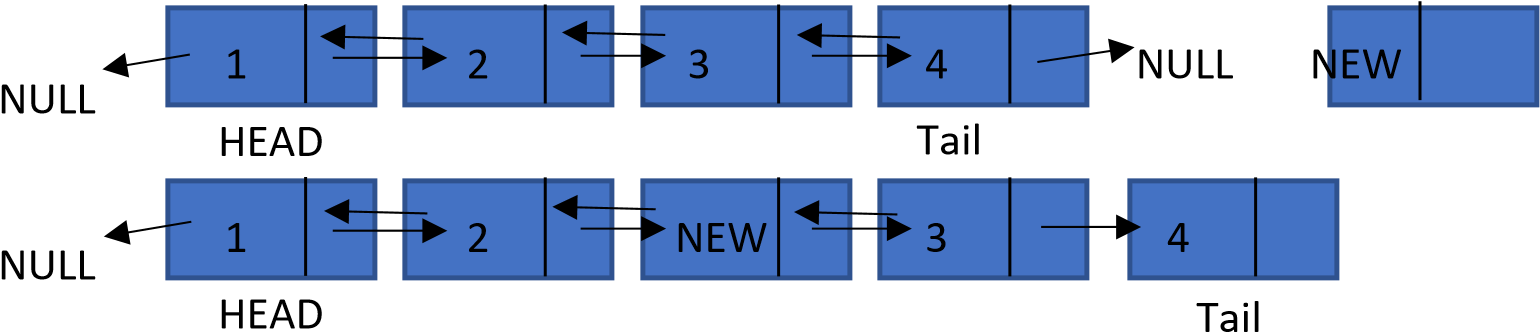
# i. Insert belakang



Penjelasan:

Menambahkan node baru di akhir akan membuat node baru menjadi tail dalam array tersebut, lalu niali yang no 4 akan merujuk pada nilai baru yang merupakan tailnya.

# ii. Insert tengah



Penjelasan:

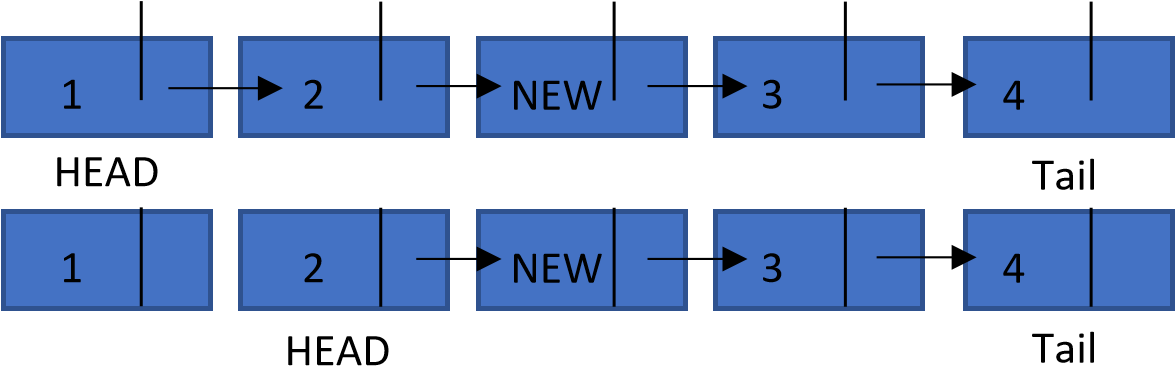
Untuk menambahkan data baru di tengah kita harus mendapkan nilai mid disebuah array terdahulu dengan cara arr.lenght/2, lalu node baru akan di insert pada tengah dengan cara bergeser dari 1, ke 2, lalu ketengah.

**3.** Gambarkan dan jelaskan proses delete depan dan tengah untuk Singly dan Doubly Linked

List

**a.** Single Linked List

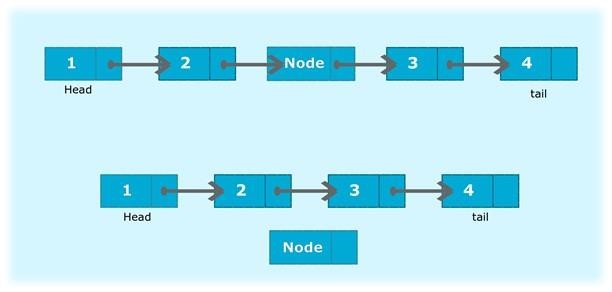
1. Delete depan

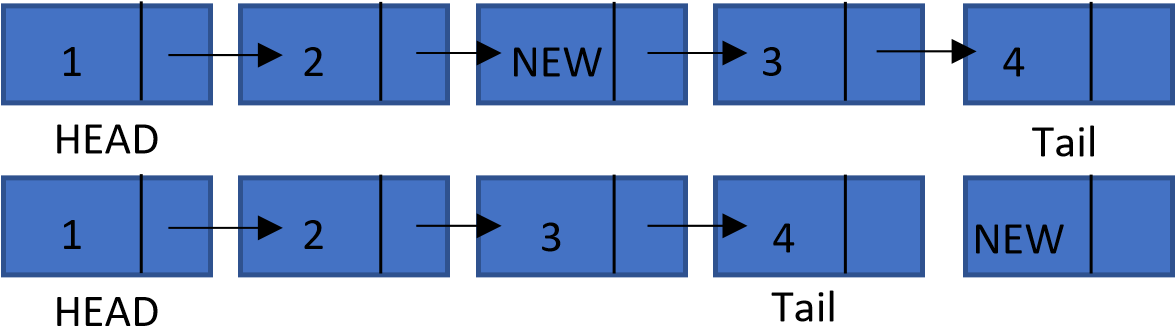


Penjelasan:

Untuk mendelete depan akan merujuk pada head array, setelah di delete maka head akan berpindah ke index setelah head, dalam kasus diatas head berpindah ke index 2, dan index satu telah keluar dari list.

1. Delete Tengah



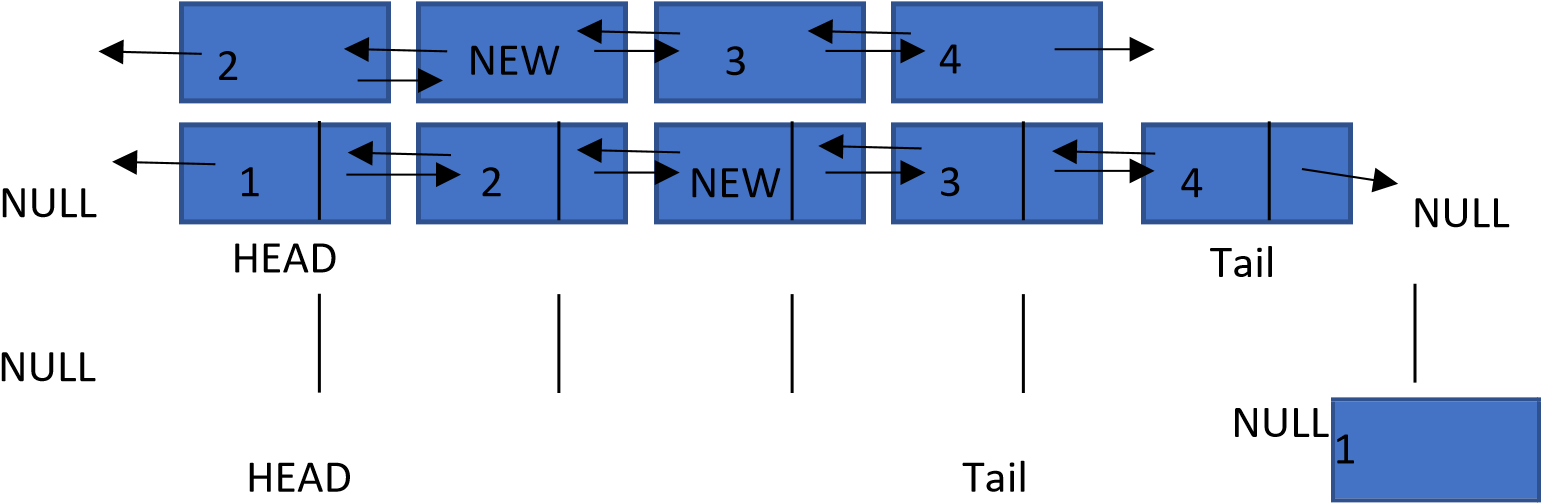


Penjelasan:

Membuat variable mid, yaitu nilai tengah, jika telah di dapat maka nilai mid akan merujuk pada node tengah dan menghupusnya.

**b.** Double Linked List

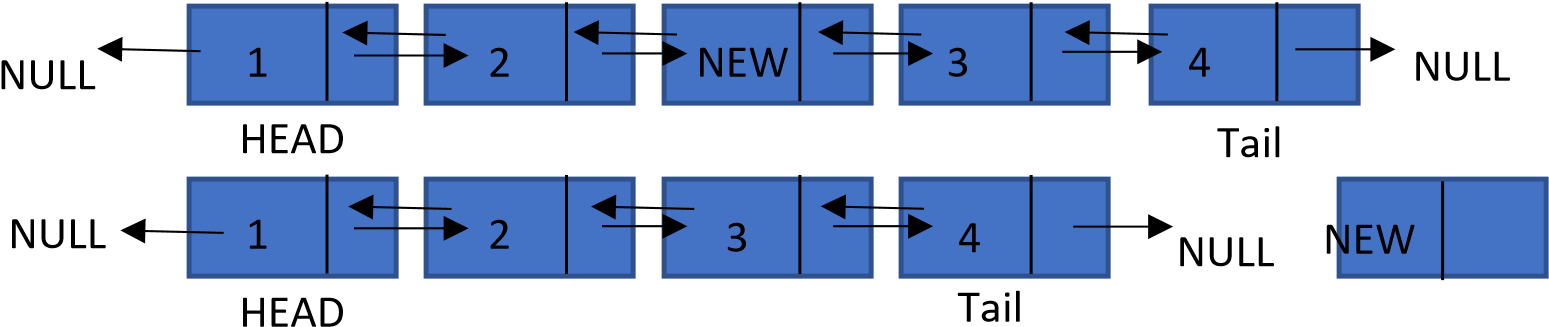
1. Delete depan



Penjelasan:

Untuk mendelete depan akan merujuk pada head array, setelah di delete maka head akan berpindah ke index setelah head, dalam kasus diatas head berpindah ke index 2, dan index satu telah keluar dari list.

1. Delete Tengah



Penjelasan:

Membuat variable mid, yaitu nilai tengah, jika telah di dapat maka nilai mid akan merujuk pada node tengah dan menghupusnya.