



Ahmad Nizar Sauki - 2306152046 Nizar ▾



Home > My courses > PROG. S1 FAK. REGULER > REG - Gasal 2024/2025 > [Reg] Struktur Data & Algoritma (A,B,C,D,E,F) ... > Pekan 5-6: Linked List > CP05 Linked List

Started on Friday, 4 October 2024, 2:58 PM

State Finished

Completed on Saturday, 5 October 2024, 10:18 AM

Time taken 19 hours 20 mins


Grade 10.00 out of 10.00 (100%)

Question 1

Correct

Mark 1.00 out of 1.00

Operasi apa yang memiliki kompleksitas **O(1)** pada Linked List tanpa reference variable last/tail (pointer ke elemen terakhir)?

- ☐ a. Menyisipkan elemen di tengah
- ☐ b. Menyisipkan elemen di akhir
- ☐ c. Mengakses elemen secara acak
- ☒ d. Menyisipkan  Menyisipkan atau menghapus elemen di awal hanya memerlukan perubahan pada referensi node pertama sehingga O(1).

Your answer is correct.

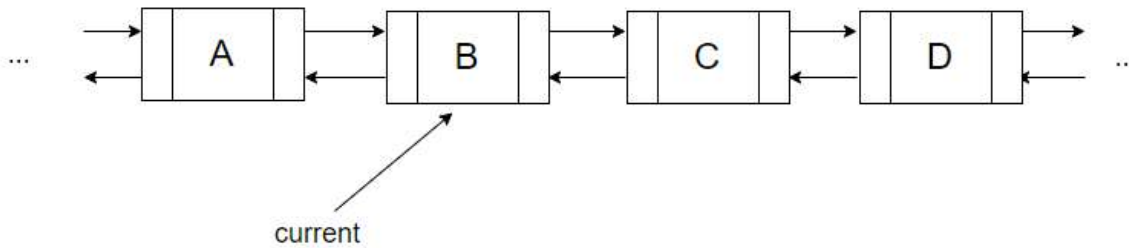
The correct answer is: Menyisipkan elemen di awal

Question 2

Correct

Mark 1.00 out of 1.00

Diketahui sebuah doubly linked list dengan sebuah pointer current yang sedang menunjuk salah satu node di dalamnya, seperti ilustrasi berikut:



Apa yang dilakukan oleh code berikut ini?

```
current.next = current.next.next;  
current.next.prev = current;
```

- ☐ a. Menghapus node berisi D dari dalam linked list
- ☐ b. Menghapus node berisi A dari dalam linked list
- ☐ c. Menghapus node berisi B dari dalam linked list
- ☒ d. Menghapus node berisi C dari dalam linked list ✓

Your answer is correct.


The correct answer is: Menghapus node berisi C dari dalam linked list

Question 3

Correct

Mark 1.00 out of 1.00

Metode mana yang **benar** untuk menambahkan elemen di akhir Linked List?

- ☒ a. `last.next = new ListNode(x);`  Menambahkan elemen di akhir Linked List melibatkan penunjukan referensi dari node terakhir ke node baru.
- ☐ b. `current = new ListNode(x);`
- ☐ c. `newNode.next = null;`
- ☐ d. `first.next = current.next;`

Your answer is correct.

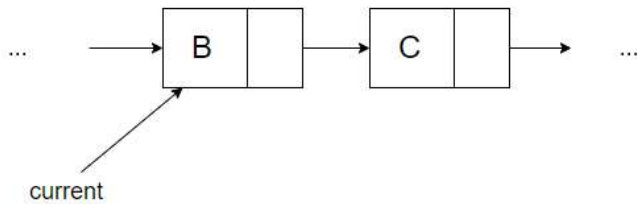
The correct answer is: `last.next = new ListNode(x);`

Question 4

Correct

Mark 1.00 out of 1.00

Diketahui sebuah singly linked list dengan sebuah pointer current yang sedang menunjuk salah satu node di dalamnya, seperti ilustrasi berikut:



Apa yang dilakukan oleh code berikut ini?

```
ListNode temp = current.next;  
current.next = new ListNode("Y", temp);  
current = current.next.next;
```

- ☐ a. Menyisipkan node berisi "Y" di posisi sebelum node berisi "B", lalu menghapus node berisi "B"
- ☐ b. Menyisipkan node berisi "Y" di posisi setelah node berisi "B", lalu menghapus node berisi "B"
- ☒ c. Menyisipkan node berisi "Y" di posisi setelah node berisi "B", lalu menggeser current ke posisi node berisi "C" ✓
- ☐ d. Menyisipkan node berisi "Y" di posisi sebelum node berisi "B", lalu menggeser current ke posisi node berisi "C"

Your answer is correct.

The correct answer is: Menyisipkan node berisi "Y" di posisi setelah node berisi "B", lalu menggeser current ke posisi node berisi "C"

Question 5

Correct

Mark 1.00 out of 1.00

Dari operasi berikut, mana saja yang memiliki kompleksitas **$O(1)$** pada Linked List yang memiliki pointer first dan tail?

- ☒ I. Insert pada akhir ✓
- ☐ II. Search elemen secara acak
- ☒ III. Insert pada awal ✓
- ☒ IV. Delete pada awal ✓

Your answer is correct.

The correct answers are: Insert pada awal, Insert pada akhir, Delete pada awal

Question 6

Correct

Mark 1.00 out of 1.00

Apa fungsi dari header node dalam Linked List?

- ☐ a. Mengatur panjang Linked List
- ☐ b. Mempercepat akses ke elemen terakhir
- ☐ c. Menyimpan data elemen pertama
- ☒ d. Menghindari kasus khusus saat menghapus elemen pertama ✓
Header node memudahkan penghapusan elemen pertama tanpa menangani kasus khusus.

Your answer is correct.




The correct answer is: Menghindari kasus khusus saat menghapus elemen pertama

Question 7

Correct

Mark 1.00 out of 1.00

Manakah dari implementasi berikut yang mendukung operasi **$O(1)$** untuk stack?

- ☒ a. Stack dengan array doubling 
- ☐ b. Queue-based stack
- ☒ c. Implementasi berbasis array  Operasi stack berbasis array dan linked list umumnya memiliki waktu $O(1)$ untuk push dan pop.
- ☒ d. Implementasi berbasis Linked List  Operasi stack berbasis array dan linked list umumnya memiliki waktu $O(1)$ untuk push dan pop.

Your answer is correct.

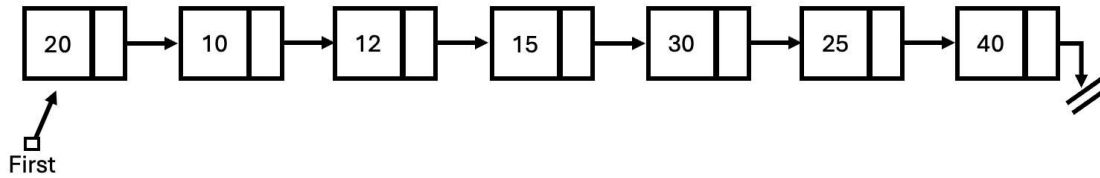
The correct answers are: Implementasi berbasis array, Implementasi berbasis Linked List

Question 8

Correct

Mark 1.00 out of 1.00

Diketahui sebuah singly linked list yang masing-masing node-nya berisi integer, seperti pada ilustrasi berikut: (pointer first menunjuk node pertama, dan pointer current masih null)



Jika potongan code berikut ini dijalankan pada linked list tersebut, apa outputnya?

```
current = first;
while(current.next.data != 15)
    current = current.next;
while(current != null){
    System.out.print(current.data + ", ");
    current = current.next;
}
```

- ☐ a. 10, 12, 15, 30, 25, 40,
- ☐ b. 15, 30, 25, 40,
- ☒ c. 12, 15, 30, 25, 40, ✓
- ☐ d. 30, 25, 40,

Your answer is correct.

The correct answer is: 12, 15, 30, 25, 40,

Question 9

Correct

Mark 1.00 out of 1.00

Apa yang terjadi jika sebuah node **dihapus** dari Linked List dalam lingkungan pemrograman bahasa Java?

- ☐ a. Node tersebut akan dihapus secara manual oleh programmer
- ☐ b. Node tersebut tetap berada di memori, dan referensi lain akan menunjuk padanya.
- ☒ c. Node tersebut dilewati oleh referensi, dan memori akan dibersihkan oleh garbage collector. ✓
- ☐ d. Node tersebut akan langsung dihapus dari memori.

Your answer is correct.

The correct answer is: Node tersebut dilewati oleh referensi, dan memori akan dibersihkan oleh garbage collector.

Question 10

Correct

Mark 1.00 out of 1.00

Apa **perbedaan** utama antara Linked List dan Array?

- ☐ a. Linked List menyimpan elemen secara terurut dalam memori, sementara array menyimpannya secara acak.
- ☒ b. Array menyimpan elemen secara berurutan di memori, sedangkan Linked List menyimpan elemen di lokasi memori yang tidak berdekatan. ✓
Array menyimpan elemen secara contiguous di memori, sementara Linked List menyimpan elemen secara non-contiguous.
- ☐ c. Linked List memiliki ukuran tetap, sedangkan array dapat berkembang sesuai kebutuhan.
- ☐ d. Linked List tidak bisa diubah ukurannya setelah dibuat, sedangkan array bisa.

Your answer is correct.

The correct answer is: Array menyimpan elemen secara berurutan di memori, sedangkan Linked List menyimpan elemen di lokasi memori yang tidak berdekatan.

◀ 06. Linked List

Jump to...



07. Tree ▶