Kuis Week 5 - Stuta

Yow Dawg, Mari Mengerjakan Kuis Agar Jadi Mahasigma Yang Alpha.
STUTUTUUTAAA

* Menunjukkan pertanyaan yang wajib diisi

Email *

- [x] Rekam **gembulnika@gmail.com** sebagai email yang disertakan dengan respons saya

Nama *

skibidi wagh

NIM *

1

Kelas *

- (•) 001
- ( ) 002
- ( ) 003

Apa Itu Searching? *

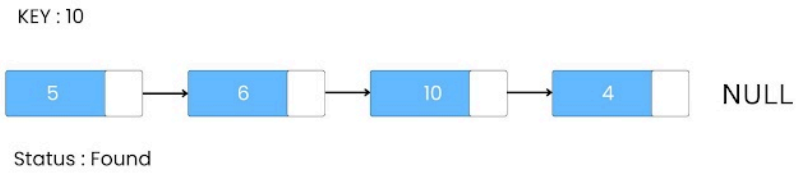
- (•) operasi dasar list dengan melakukan aktivitas pencarian terhadap node tertentu.
- ( ) operasi dasar list dengan menghapus aktivitas terhadap note tertentu.
- ( ) operasi dasar list dengan menambahkan aktivitas terhadap note tertentu.
- ( ) operasi dasar list dengan mengurang aktivitas terhadap note tertentu.

Apa proses dari Searching? *

- (●) Proses ini berjalan dengan mengunjungi setiap node dan berhenti setelah node yang dicari ketemu.
- ( ) Proses ini berjalan sendiri dengan mengunjungi node tertentu dan berhenti setelah node yang dicari ketemu.
- ( ) Proses ini berhenti dengan mengunjungi setiap node dan berhenti setelah node yang dicari ketemu.
- ( ) Proses ini stop & go dengan mengunjungi setiap node dan berhenti setelah node yang dicari ketemu.

Tujuan dari pencarian adalah menemukan node dengan nilai 10 di dalam linked list. *

Manakah pernyataan yang benar tentang proses pencarian node dengan nilai 10?



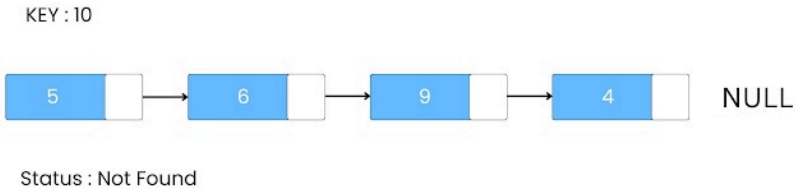
- ( ) Nilai 10 tidak ditemukan karena node terakhir memiliki nilai 4.
- ( ) Nilai 10 ditemukan pada node kedua.
- (●) Nilai 10 ditemukan pada node ketiga.
- ( ) Nilai 10 ditemukan pada node keempat.

Kemudahan memakai operasi Searching ? *

- (●) operasi-operasi seperti insert after, delete after, dan update akan lebih mudah.
- ( ) operasi-operasi seperti delete after, dan update akan lebih mudah.
- ( ) operasi-operasi seperti update akan lebih mudah.
- ( ) operasi-operasi seperti insert after akan lebih mudah.

Tujuan dari pencarian adalah menemukan node dengan nilai 10 di dalam linked list. *

Manakah pernyataan yang benar tentang proses pencarian node dengan nilai 10?



- ◯ Nilai 10 ditemukan pada node kedua.
- ◯ Nilai 10 ditemukan pada node ketiga.
- ◯ Nilai 10 ditemukan pada node keempat.
- ◉ Nilai 10 tidak ditemukan dalam linked list.

Implementasi manakah yang dengan benar mengreverse linkedlist? *

```
ListNode* reverseLinkedList(ListNode* head) {
    // A)
    if (head == nullptr || head->next == nullptr) {
        return head;
    }
    ListNode* newHead = reverseLinkedList(head->next);
    head->next->next = head;
    head->next = nullptr;
    return newHead;
```

◉ Opsi 1

```
// B)
ListNode *prev = nullptr, *current = head, *nextNode;
while (current != nullptr) {
    nextNode = current->next;
    current->next = prev;
    prev = current;
    current = nextNode;
}
return prev;
```

◯ Opsi 2

```
// C)
std::vector<ListNode*> stack;
ListNode* current = head;
while (current != nullptr) {
    stack.push_back(current);
    current = current->next;
}
ListNode* newHead = stack.back();
current = newHead;
stack.pop_back();
while (!stack.empty()) {
    current->next = stack.back();
    stack.pop_back();
    current = current->next;
}
current->next = nullptr;
return newHead;
```

◯ Opsi 3

```
// D)
if (head == nullptr || head->next == nullptr) {
    return head;
}
ListNode *left = head, *right = head->next;
left->next = nullptr;
while (right != nullptr) {
    ListNode* temp = right->next;
    right->next = left;
    left = right;
    right = temp;
}
return left;
```

◯ Opsi 4

*

Dalam linked list, jika kita ingin mencari elemen yang dimulai dari node current dan elemen tersebut tidak ditemukan, di mana current seharusnya menunjuk pada akhir pencarian?

- (●) nullptr
- ( ) Node pertama dalam list
- ( ) Node yang berisi nilai terkecil yang paling dekat dengan kunci pencarian
- ( ) Node dengan nilai lebih besar dari kunci pencarian

*

Manakah dari berikut ini yang dapat digunakan untuk menghentikan pencarian nilai dalam linked list lebih awal, jika list diurutkan dalam urutan ascending?

```
1   if (head->data > key)
2       return false;
```

(●) Opsi 1

```
4   if (head->data == key)
5       return true;
```

( ) Opsi 2

```
7   if (head->data < key)
8       return false;
```

( ) Opsi 3

```
10  if (head == nullptr)
11      return false;
```

( ) Opsi 4

Diberikan sebuah linked list: 1 -> 3 -> 5 -> 7 -> nullptr. Apa yang akan dikembalikan fungsi berikut saat mencari nilai 5 dalam linked list? *

```
struct Node {
    int data;
    Node* next;
};

bool search(Node* head, int key) {
    while (head != nullptr) {
        if (head->data == key)
            return true;
        head = head->next;
    }
    return false;
}
```

- ○ false
- ◉ true
- ○ Compilation Error
- ○ Node Error

Apa implementasi yang benar dari fungsi yang mengembalikan node ke-n dari akhir linked list? *

```
ListNode* nodeKeX(ListNode* head, int n) {
    // A)
    int length = 0;
    ListNode* current = head;
    while (current != nullptr) {
        length++;
        current = current->next;
    }
    if (n > length) {
        return nullptr;
    }
    current = head;
    for (int i = 0; i < length - n; i++) {
        current = current->next;
    }
    return current;
```

◉ Opsi 1

```
    // B)
    ListNode *fast = head, *slow = head;
    for (int i = 0; i < n; i++) {
        if (fast == nullptr) {
            return nullptr;
        }
        fast = fast->next;
    }
    while (fast != nullptr) {
        slow = slow->next;
        fast = fast->next;
    }
    return slow;
```

○ Opsi 2

```
    // C)
    std::vector<ListNode*> stack;
    ListNode* current = head;
    while (current != nullptr) {
        stack.push_back(current);
        current = current->next;
    }
    if (n > stack.size()) {
        return nullptr;
    }
    return stack[stack.size() - n];
```

○ Opsi 3

```
    // D)
    ListNode* current = head;
    for (int i = 0; i < n; i++) {
        if (current == nullptr) {
            return nullptr;
        }
        current = current->next;
    }
    return current;
}
```

○ Opsi 4

Kirim

Kosongkan formulir

Google Formulir