

REMEDIAL NILAI UTS ALGORITMA DAN STRUKTUR DATA

Nama: Mohammad Alwi Nanda Saputra

NIM : A11.2023.15442

KEL : A11.4215

Linked list adalah struktur data linear di mana elemen (disebut **node**) saling terhubung menggunakan penunjuk. Setiap node berisi data dan referensi ke simpul berikutnya dalam daftar. Berbeda dengan array, daftar tertaut memungkinkan penyisipan atau penghapusan elemen secara efisien dari posisi mana pun dalam daftar, karena node-nodenya tidak disimpan secara berdekatan di memori. Ada beberapa jenis daftar tertaut, termasuk daftar tunggal (di mana setiap node menunjuk ke node berikutnya) dan daftar ganda (di mana setiap node menunjuk ke simpul berikutnya dan sebelumnya).

1. Insert(penambahan)

a. Insert First

Penambahan node baru yang terletak didepan head dan akan menjadi head yang baru menggantikan yang awal. Langkah-langkah menggunakan insert first:

Node awal: 1,7

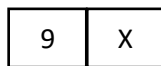
Insert : 9

Hasil : 9,7,1

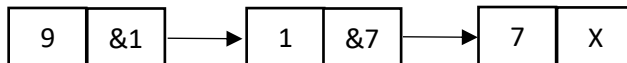
Node Awal



Insert



Hasil akhir



```
// Comment node baru dengan data sesuai dan next kosong
22 Node* newNode = new Node;
23 newNode->data = 1; // Masukkan data ke 1 (karena)
24 newNode->next = nullptr; // Mengatur next ke nullptr (kosong)
25
26 // Menambahkan di front node baru
27 cout << "Data baru pada node: " << newNode->data << endl;
28 if (newNode->next == nullptr) {
29     // Node baru tidak memiliki node selanjutnya
30     cout << "Node baru tidak memiliki node selanjutnya" << endl;
31 }
32 insertFirst(newNode, 7);
33
34 // Menambahkan data data linked list
35 insertLast(newNode, 9);
36
37 // Menampilkan data data linked list
38 cout << "Data pada linked list:" << endl;
39 while (newNode != nullptr) {
40     cout << "Data: " << newNode->data << " | Next: " << newNode->next << endl;
41     newNode = newNode->next;
42 }
```

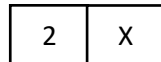
b. Insert Last

Penambahan node baru pada linked list, tetapi node baru ini terletak di akhir(tail), langkah-langkah penggunaan insert last:

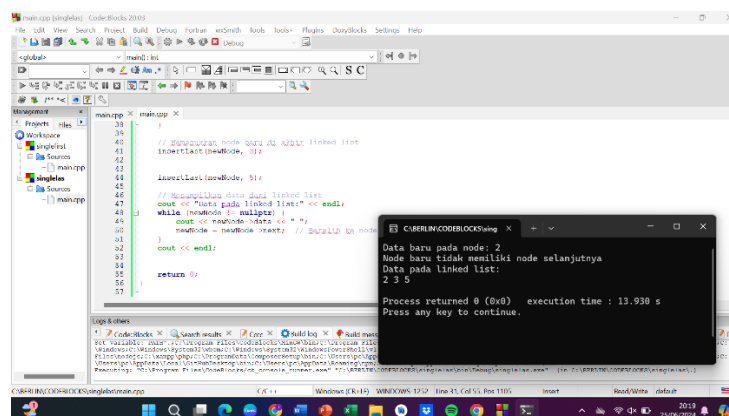
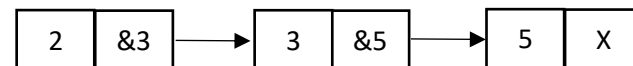
Node awal: 3,5
 Node baru: 2
 Hasil akhir: 2,3,5
 Node awal



Node baru



Hasil akhir

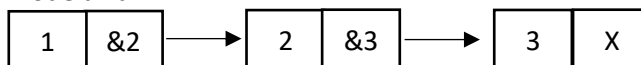


c. Insert After

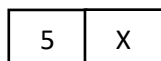
Penambahan node baru pada linked list, tetapi node baru yang ditambahkan akan berada di setelah data yang dituju, Langkah-langkah pada insert after:

Node awal: 1,2,3
 Node baru: 5
 Hasil akhir: 1,2,5,3

Node awal

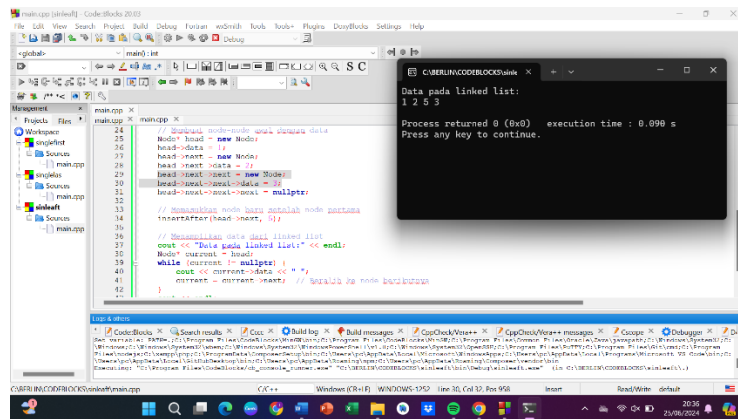


Node baru



Hasil akhir





2. Delete(penghapusan)

a. Delete first

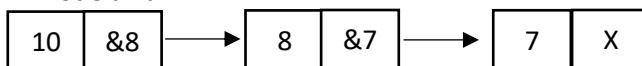
Salah satu cara penghapusan pada single linkedlist yang dilakukan dengan cara menghapus node paling depan/head, Langkah-langkah delete first:

Node awal: 10,8,7

Delet : 10

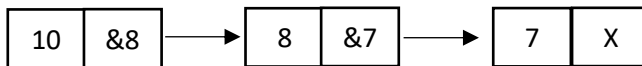
Hasil akhir: 8,7

1. Node awal



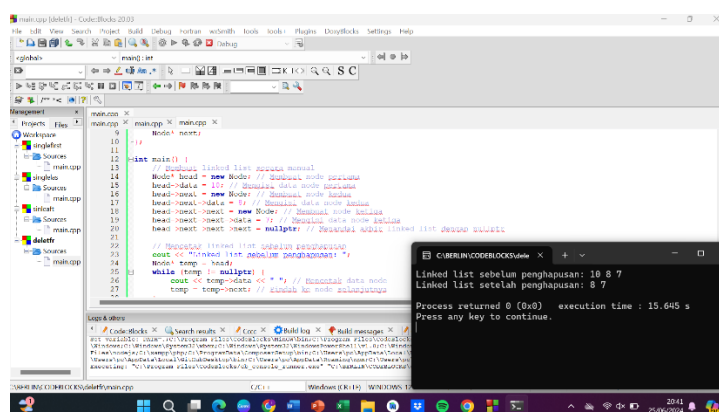
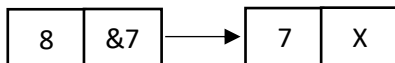
2.1 Menyimpan node yang menjadi head pertama kedalam temp

2.2 Memindah head



3. menghapus head

3.1 hasil akhir



b. Delete after

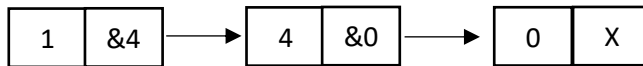
Cara untuk menghapus data yang berada setelah data/node, Langkah-langkah delete after

Node awal:1,4,0

Delete after:4

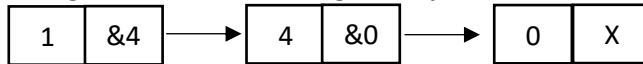
Hasil akhir:1,0

Node awal

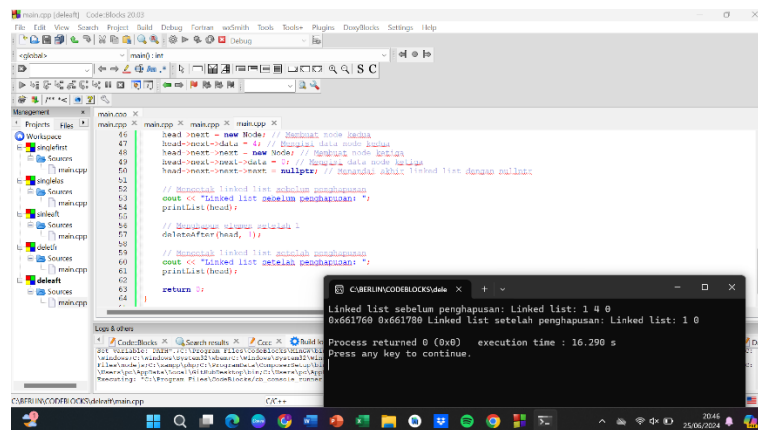
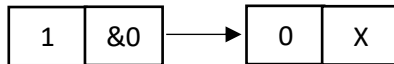


Menyimpan node target kedalam temp

Mengubah Alamat awal target menjadi Alamat milik node yang akan dihapus



Hasil akhir



c. Delet last

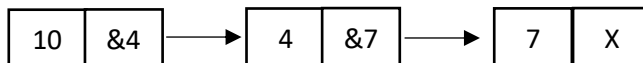
Ini adalah cara penghapusan data yang berada diposisi tail atau posisi paling belakang, berikut cara penggunaannya:

Node awal: 10,4,7

Delete :7

Hasil akhir: 10,4

1. Node awal



2. menggunakan while untuk mencari node terakhir(null) dan memasukan node yang akan dihapus tersebut ke temp

2.1 menghapus node terakhir



2.2 mengganti next milik data yang terletak sebelum data yang dihapus menjadi null

