LAPORAN PRAKTIKUM
MATA KULIAH TEORI ALGORITMA STRUKTUR DATA

Dosen Pengampu : Triana Fatmawati, S.T, M.T

PERTEMUAN 11 : LINKED LIST

Nama : Yonanda Mayla Rusdiaty

NIM : 2341760184

Prodi : D-IV Sistem Informasi Bisnis

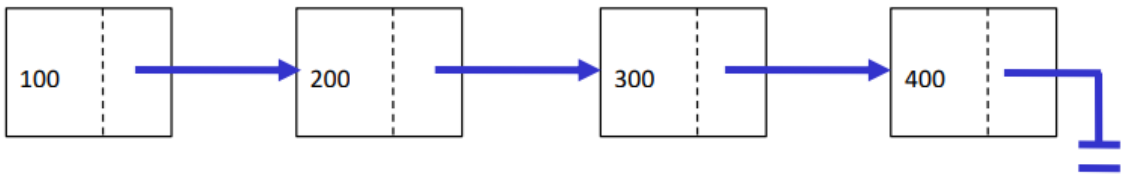
JURUSAN TEKNOLOGI INFORMASI

POLITEKNIK NEGERI MALANG

2024

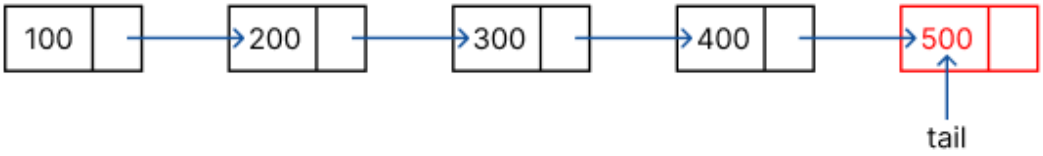
LATIHAN

1. Suatu link list berisi 4 node berikut:



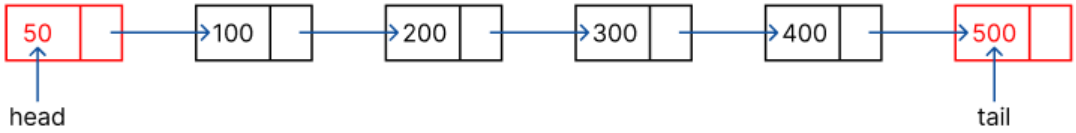
a. Tambahkan node baru dengan data 500 dari belakang.

➔ Menambahkan node baru 500 ke bagian belakang linked of list = **addLast().** Dalam kondisi ini, linked list telah berisi node, jadi akan dilakukan iterasi terlebih dahulu hingga node terakhir (tail) yaitu 400 ditemukan. Kemudian, set node baru = 500 sebagai next node dari tail sebelumnya.



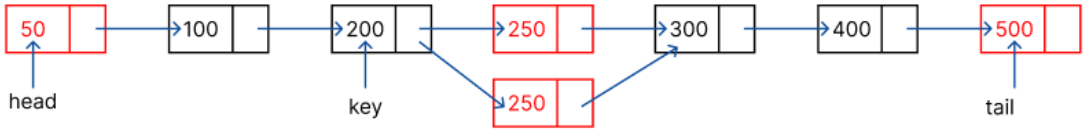
b. Tambahkan node baru dengan data 50 dari depan.

➔ Menambahkan node baru 50 dari depan linked list = **addFirst().** Dalam kondisi ini, linked list telah berisi node, maka atribut next pada node input yaitu 50, akan menunjuk ke head node sebelumnya yaitu 100, dan node input akan dijadikan sebagai head yang baru.



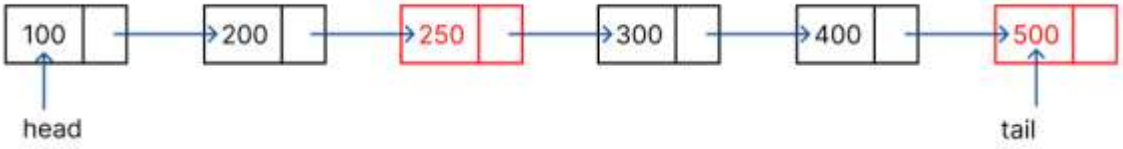
c. Tambahkan node dengan data 250 setelah node 200

➔ Untuk menambahkan node baru yaitu 250 setelah node 200, digunakan **insertAfter().** Node 200 disini digunakan sebagai key node. Dalam kondisi ini, linked list telah berisi node, maka akan dilakukan iterasi terlebih dahulu hingga menemukan node yang datanya sama dengan 250. Jika sudah ditemukan, jadikan next node dari key node sebagai next node dari node baru, dan jadikan node baru sebagai next node dari key node.



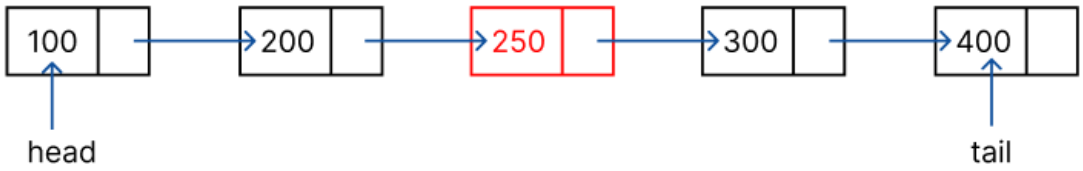
d. Hapus node depan

➔ Menghapus node depan (head) berarti **removeFisrt().** removeFirst() digunakan untuk menghapus node pertama pada linked list. Dalam kondisi ini, kita akan menjadikan next node dari head sebagai head yang baru.



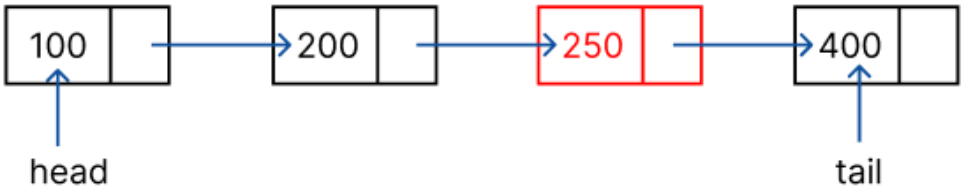
e. Hapus node belakang

➔ Menghapus node terakhir pada linked list berarti **removeLast().** Dalam kondisi ini, akan dilakukan iterasi terlebih dahulu untuk menemukan node terakhir kedua, kemudian set next nodenya dengan null.



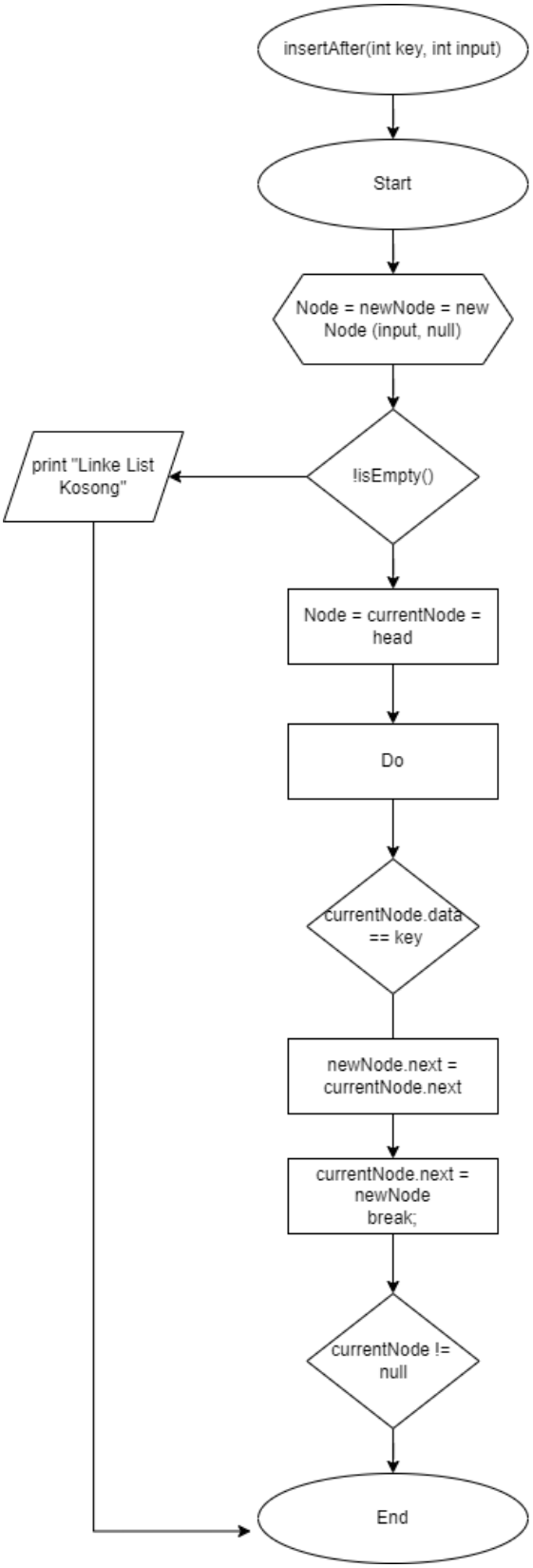
f. Hapus node yg memiliki data 300

→ Untuk menghapus data 300, berarti menggunakan **remove().** Fungsi remove (int key), digunakan untuk menghapus node dengan data sesuai key. Dalam kondisi ini, akan dilakukan iterasi untuk menemukan node yang next node nya berisi key. Arahkan next node ke node setelah key.



2. Buat pseudocode/flowchart untuk 2 fungsi berikut :
   a. Fungsi insertAt(int index, int key) untuk menambahkan data baru pada index tertentu

      ➔ Berikut adalah flowchart dari insertAfter()



b. Fungsi removeAt(index) untuk menghapus data pada index tertentu

➔ Berikut adalah flowchart dari remove()