

Linked List

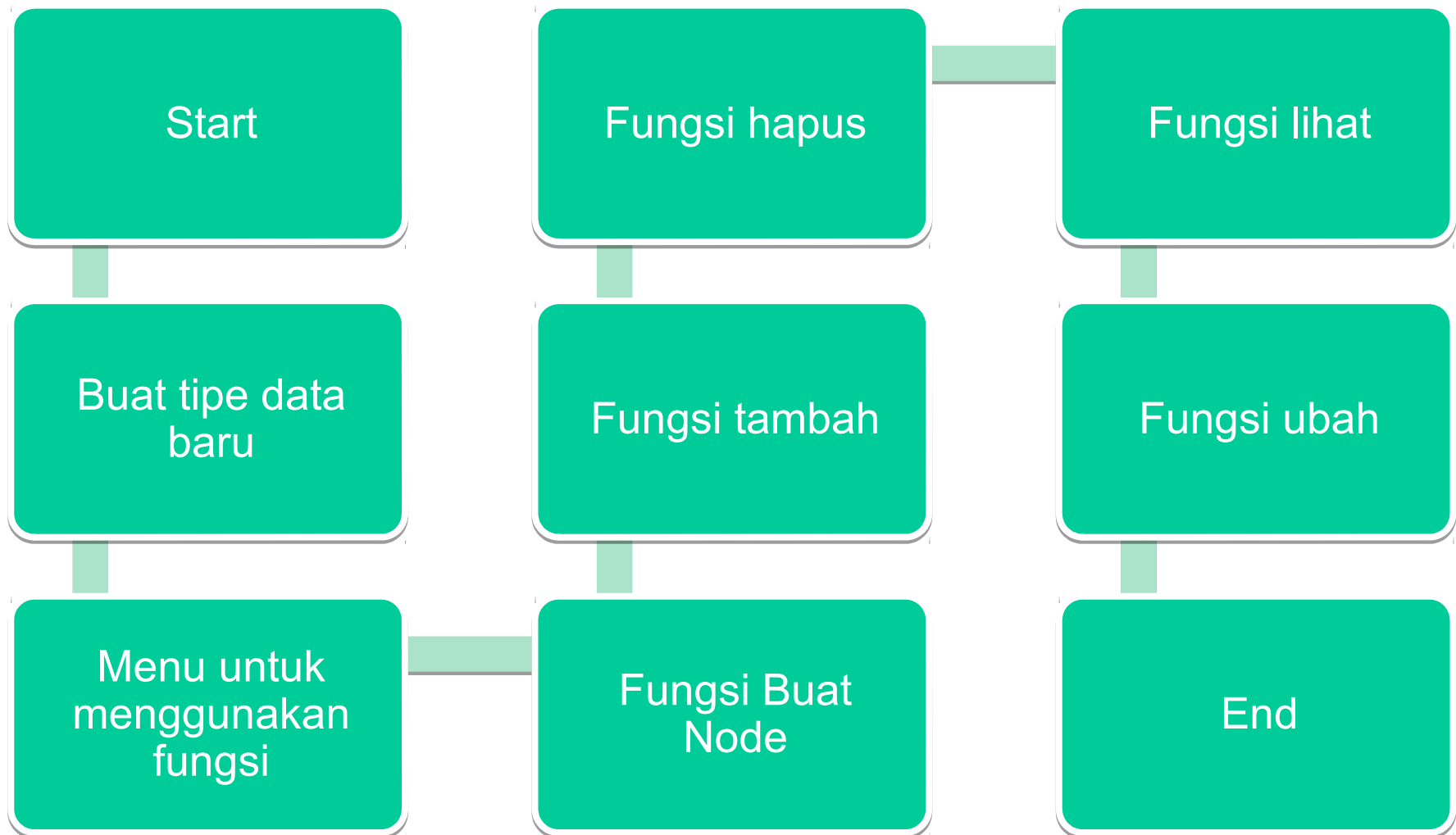


Nama

Email

Contact

Step yang dilakukan



Buat tipe data baru(Struct)

```
struct node(disarankan sebagai nama struct){  
    isi data;  
    pointer yang akan menunjuk node selanjutnya;  
}
```

Pointer penanda jika node itu di urutan awal;

Pointer penanda jika node itu di urutan akhir;

Pointer penanda jika node itu baru;

Pointer penanda untuk membantu saat melakukan perpindahan;(Bisa lebih dari 1 pointer)

PIKIRKAN SENDIRI (CreateReadUpdateDelete)

- Tambah(Depan, Belakang, Diantara)
- Hapus(Depan, Belakang, Diantara, Semua)
- Lihat(Depan, Belakang, Diantara, Semua)
 - Ubah(Depan, Belakang, Diantara)

Buat Node

1. Buat fungsi dengan parameter data yang akan dimasukkan kedalam node
2. Buat node baru dan tandai dengan pointer sebagai baru
3. Pointer baru yang variabelnya data diisikan sesuai parameter
4. Pointer baru yang variabelnya alamat diisikan sesuai kosong(karena belum ada node berikutnya)

Tambah Depan

1. Berikan inputan data yang akan dimasukkan
2. Panggil fungsi untuk buat node
3. Jika linkedlist masih kosong lakukan step 4 – 5, jika tidak lakukan step 6 - 7
4. Pointer awal & pointer akhir = Pointer baru
5. Pointer akhir dengan variable alamat selanjutnya = null
6. Pointer baru dengan variabel alamat selanjutnya = pointer awal
7. Pointer awal = pointer baru
8. Tampilkan informasi data sudah tersimpan

Tambah Tengah

1. Jika pointer awal != kosong & pointer akhir lakukan step 2 - 11, jika tidak lakukan step 12
2. Pointer bantu = pointer awal
3. Masukkan input posisi yang ingin dimasukkan
4. Untuk $i = 1$ dimana $i < \text{posisi} - 1$ dan i selalu ditambah 1, lakukan step 5
5. Pointer bantu = pointer baru dengan variable alamat selanjutnya
6. Berikan inputan data yang akan dimasukkan
7. Panggil fungsi untuk buat node
8. Pointer bantu1 = pointer bantu dengan variabel alamat selanjutnya
9. Pointer baru dengan variabel alamat selanjutnya = pointer bantu1
10. Pointer bantu dengan variabel alamat selanjutnya = pointer baru
11. Tampilkan informasi data berhasil ditambahkan
12. Tampilkan informasi bahawa operasi tidak bisa dilakukan

Tambah Belakang

1. Berikan inputan data yang akan dimasukkan
2. Panggil fungsi untuk buat node
3. Jika linkedlist masih kosong lakukan step 4 – 5, jika tidak lakukan step 6 - 7
4. Pointer awal & pointer akhir = Pointer baru
5. Pointer akhir dengan variable alamat selanjutnya = null
6. Pointer akhir dengan variable alamat selanjutnya = pointer baru
7. Pointer akhir = pointer baru
8. Tampilkan informasi data berhasil disimpan

1. Jika linkedlist kosong tampilkan informasi bahawa linkedlist kosong
2. Jika pointer awal dengan variable alamat selanjutnya kosong panggil fungsi lihat depan
3. Jika bukan step 1 – 2, lakukan step 4 - 7
4. Pointer bantu = pointer awal
5. Selama pointer bantu != kosong, lakukan step 6 - 7
6. Tampilkan variable data pada pointer bantu
7. Pointer bantu = pointer bantu dengan variable alamat selanjutnya

1. Jika linkedlist kosong tampilkan informasi bahwa linkedlist kosong
2. Jika tidak, tampilkan variable data pada pointer head

1. Jika pointer awal \neq kosong dan pointer akhir, lakukan step 2 - 6
2. Pointer bantu = pointer awal
3. Masukkan input posisi yang ingin dimasukkan
4. Untuk $i = 1$ dimana $i < \text{posisi} - 1$ dan i selalu ditambah 1, lakukan step 5
5. Pointer bantu = pointer baru dengan variable alamat selanjutnya
6. Tampilkan informasi variable data yang ada dipointer bantu

Lihat Belakang

1. Jika linkedlist masih kosong lakukan tampilkan informasi bahwa linkledlist koson
2. Jika tidak tampilkan variable data pada pointer akhir

Ubah Depan

1. Jika pointer awal ada isinya lakukan step 2 - 5, jika tidak tampilkan info bahwa linkedlist kosong
2. Tampilkan informasi data yang akan diubah adalah data pata pointer awal
3. Masukkan data baru
4. Pointer head dengan variable data = data baru
5. Tampilkan informasi data berhasil diubah

Ubah Tengah

1. Jika pointer awal != kosong dan pointer akhir, lakukan step 2 – 6, jika tidak tampilkan informasi gagal
2. Pointer bantu = pointer awal
3. Masukkan input posisi yang ingin diubah
4. Untuk $i = 1$ dimana $i < \text{posisi} - 1$ dan i selalu ditambah 1, lakukan step 5
5. Pointer bantu = pointer baru dengan variable alamat selanjutnya
6. Tampilkan informasi variable data yang diubah dipointer bantu
7. Masukkan data baru
8. Pointer bantu dengan variable data = data baru
9. Tampilkan informasi data berhasil diubah

Ubah Belakang

1. Jika linkedlist masih kosong lakukan step 2 – 5, jika tidak tampilkan info linked list kosong
2. Tampilkan info data yang diubah ada di pointer akhir
3. Masukkan data baru
4. Pointer akhir dengan variable data = data baru
5. Tampilkan informasi data berhasil diubah

Hapus Depan

1. Jika linked list kosong lakukan step 2 – 9, jika tidak tampilkan info linkedlist kosong
2. Jika pointer awal dengan variable alamat selanjutnya kosong lakukan step 3, jika tidak lakukan step 4 - 9
3. Pointer awal dan pointer akhir = kosong
4. Data yang dihapus ada di pointer awal dengan variable data
5. Pointer bantu = pointer head
6. Pointer bantu = pointer bantu dengan variable alamat selanjutnya
7. Pointer awal dengan variable alamat selanjutnya = kosong
8. Pointer awal = pointer bantu
9. Tampilkan info data berhasil dihapus

Hapus Tengah

1. Jika pointer awal != kosong dan pointer akhir, lakukan step 2 – 6, jika tidak tampilkan informasi gagal
2. Pointer bantu = pointer awal
3. Masukkan input posisi yang ingin dihapus
4. Untuk $i = 1$ dimana $i < \text{posisi} - 1$ dan i selalu ditambah 1, lakukan step 5
5. Pointer bantu = pointer baru dengan variable alamat selanjutnya
6. Pointer bantu1 = pointer bantu dengan variable alamat selanjutnya
7. Pointer bantu dengan variable alamat selanjutnya = pointer bantu1 dengan variable alamat selanjutnya
8. Pointer bantu1 dengan variable alamat selanjutnya = kosong
9. Tampilkan bahwa data di pointer bantu1 dihapus
10. Tampilkan informasi data berhasil dihapus

Hapus Belakang

1. Jika linkedlist masih kosong lakukan step 2 – 5, jika tidak tampilkan info linked list kosong
2. Jika pointer akhir dengan variable alamat selanjutnya kosong, lakukan step 3
3. Tampilkan info data yang dihapus ada di pointer akhir
4. Pointer bantu = pointer awal
5. Selama pointer bantu dengan variable alamat selanjutnya bukan pointer akhir, lakukan step 5
6. Pointer bantu = pointer bantu dengan variable alamat selanjutnya
7. Pointer bantu dengan variable alamat selanjutnya = kosong
8. Pointer bantu = pointer akhir
9. Tampilkan informasi data berhasil dihapus

Hapus Semua

Coba Cari tau!!!