**Tugas Individu Struktur Data**

**A11.4308**

****

**Linked List**

Arvie Arvearie Yonismara A11.2020.12792

**Program Studi Teknik Informatika**

**Fakultas Ilmu Komputer**

**Universitas Dian Nuswantoro**

# DAFTAR ISI

[DAFTAR ISI 2](#_Toc86697506)

[BAB I 3](#_Toc86697507)

[1. Latar Belakang 3](#_Toc86697508)

[2. Rumusan Masalah 3](#_Toc86697509)

[3. Kegunaan dari Linked List 3](#_Toc86697510)

[BAB II 5](#_Toc86697511)

[1. Komposisi Linked List 5](#_Toc86697512)

[2. Cara Kerja Linked List 7](#_Toc86697513)

[3. Fungsi fungsi menggunakan single linked list 9](#_Toc86697514)

[BAB III 9](#_Toc86697515)

[4. Notasi Algoritmik 10](#_Toc86697516)

[5. Contoh program c++ 10](#_Toc86697517)

[6. Contoh pemrograman C++ berdasarkan review diatas 12](#_Toc86697518)

[DAFTAR PUSTAKA 16](#_Toc86697519)

# BAB I

**PENGANTAR LINKED LIST**

## Latar Belakang

Struktur data adalah metode penyimpanan, pengolahan, dan penyusunan data didalam media penyimpanan atau memori dalam komputer. Tujuan dari struktur data adalah agar data tersebut bisa digunakan secara efisien. Struktur data diperlukan dalamperencanaan algoritma dan penyusunan program sehingga terciptalah algoritma pemrograman yang lebih jelas ,tepat, sederhana dan efisien.

Struktur data memiliki beberapa bentuk algoritma dan setiap bentuk algoritma memiliki peranan dan kegunaanya masing masing. Yang akan dibahas pada makalah ini adalah mengenai Linked List. Pada makalah ini juga akan membahas mengenai salah satu macam dari linked list yaitu Single Linked List.

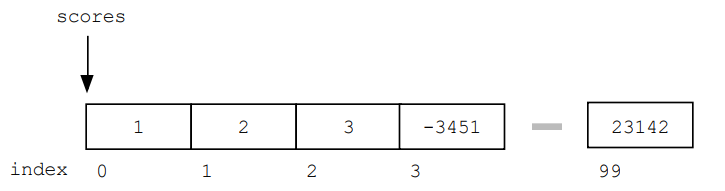
## Rumusan Masalah

1. Apa yang dimaksud dengan Linked list?
2. Apa kegunaan dari Linked list?
3. Apa itu Single Linked List?
4. Apa saja komponen yang ada pada Linked List?
5. Bagaimana cara kerja dari Algoritma Linked List?
6. Apa saja fungsi-fungsi yang dapat dibuat dalam Linked List?
7. Bagaimana contoh algoritma Linked List?
8. Apa perbedaan Linked List dengan Array?

## Kegunaan dari Linked List

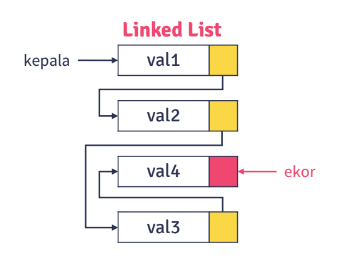
Linked List adalah salah satu bentuk dari struktur data yang mana berbentuk suatu kumpulan elemen yang disebut nodes dan terhubung ke nodes selanjutnya dengan pointer. Linked List mirip dengan Array karena keduanya menyimpan koleksi data. Untuk menjelaskan mengenai kegunaan Linked List, kita harus mengetahui bagaimana array bekerja.

Kita ambil contoh array ‘int scores[100]’ dan kita isikan array kosong tersebut dengan isian tiga elemen pada indeks awal. Berikut ilustrasi dari blok array yang sudah dibuat.



Kuncinya adalah seluruh bagian array dialokasikan sebagai satu blok memori. Setiap element dalam array mempunyai masing-masing tempatnya dalam array. Dalam hal ini dapat kita lihat bahwa ukuran array tidak dapat dirubah (dalam contoh ini 100 element). Selain itu, setiap elemen yang ada dalam array tertampung dalam lokasi memori yang berurutan. Sehingga apabila ada perubahan data memerlukan waktu yang lebih Panjang.

Maka dari itu, dari kekurangan-kekurangan yang terdapat pada array munculah struktur data Linked List. Linked Liist memiliki keunggulan dan kekurangannya tersendiri, kana tetapi linked list muncul untuk mengatasi bagian kekurangan dari array.



Array menggunakan strategi mengalokasikan memory untuk semua elemen dalam satu blok memori. Sedangkan linked list menggunakan strategi yang berbeda yaitu dengan mengalokasikan setiap masing-masing element terpisah dan dialokasikan Ketika dibutuhkan saja. Struktur data Linked list memiliki beberapa keunggulan dari pada array. Seperti ukuran yang lebih dinamis, alokasi penggunaan memori yang dapat disesuaikan dengan penggunanya, dan penambahan maupun penghapusan dapat dilakukan dalam waktu yang lebih cepat.

# BAB II

**PEMBAHASAN**

Komposisi Linked List

Untuk menggunakan / memnuat program linked list, terdapat beberapa komponen yang kita perlukan. Komponen-komponen inilah yang nantinya akan digunakan dalam linked list sehingga menciptakan kode program yang dinamis dan efisien.

1. Node

Komposisi dalam Linke list yang pertama adalah node. Node pada linked list berisi elemen dan pointer yang menuju ke node selanjutnya. Node sendiri dapat diisi dengan berbagai macam tipe data, seperti string, integer, dan lainnya. Node inilah yang nanti akan saling dihubngkan dengan pointer. Untuk membuat sebuah node kita membutuhkan tipe data ‘struct …nama node’.

struct node {

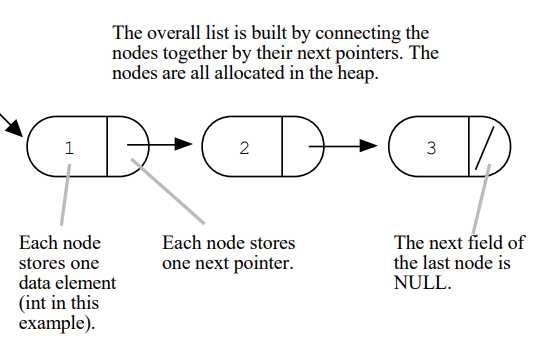
int data;

struct node\* next;

};

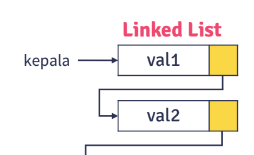
1. Pointer

Komposisi linked list yang selanjutnya adalah pointer. Pointer ini merupakan komponen yang sangat penting dikarenakan pointer ini ynag menjadi pembeda antara array dan linked list. Pointer disini berfungsi untuk menghubungkan antara node1 ke node selanjutnya. Pointer ini juga yang menjadikan struktur data linked list menjadi dinamis.

Cara kerja pointer pada linked list yaitu sebuah "pointer" menyimpan referensi ke variabel lain. Di sisi lain sebuah pointer akan bernilai ‘NULL’ Ketika sebuah pointer sedang tidak menunjuk ke sebuah pointer selanjutnya atau kosong.

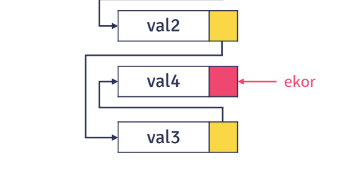
1. Head

Komponen slanjutnya adalah head. Head dalam linked list ini digunakan sebagai penanda untuk node pertama.



1. Tail

Komponen tail dalam linked list digunakan sebagai penanda untuk node yang terakhir.

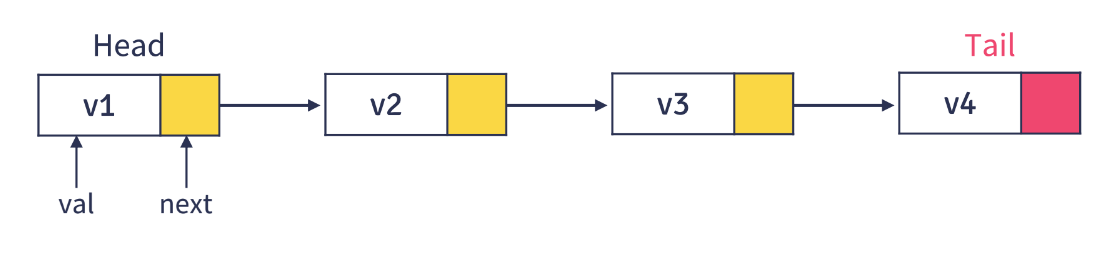


1. fungsi cetak / print

Seperti namanya vriabel ini digunakan untuk mencetak atau menampilkan linked list yang sudah kita buat.

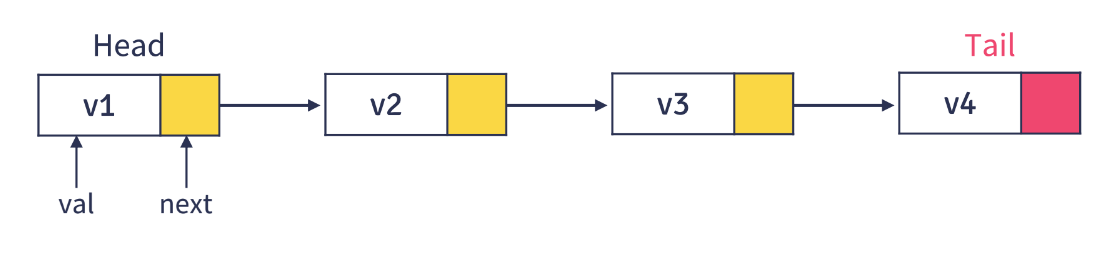
Cara Kerja Linked List

Setelah mengetahui apa saja komponen dalam Linked list, tentunya selanjutnya adalah bagaimana cara kerja linked list. Disini kita contoh kan menggunakan analogi Single Linked List sederhana.



pointer

Dalam contoh cara kerja ini menggunakan Single linked List. Single linked list adalah Linked list yang hanya memiliki satu buah pointer yang menunjuk ke node selanjutnya. Maka dari itu hanya ada satu garis panah(pointer) yang menunjuk ke node selanjutnya.



pointer

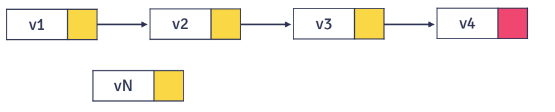
NULL

Pada contoh diatas (single linked list), navigasi item hanya maju. Maka dari itu, Ketika variable pointer sudah mencapai node terakhir(tail) pointer akan menunjuk ke NULL. Yang berarti bahwa current pointer berada di node yang paling akhir dan tidak ada yang bisa terhubung dengan pointer.

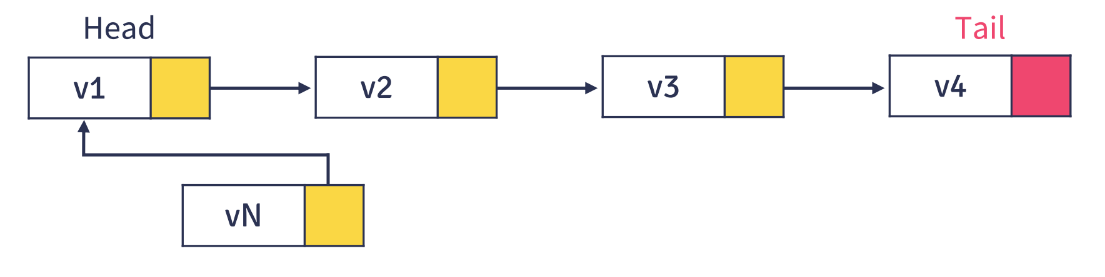
Kemudian kita akan contohkan cara kerja Single linked list dengan fungsi menambahkan variable baru menjadi head.



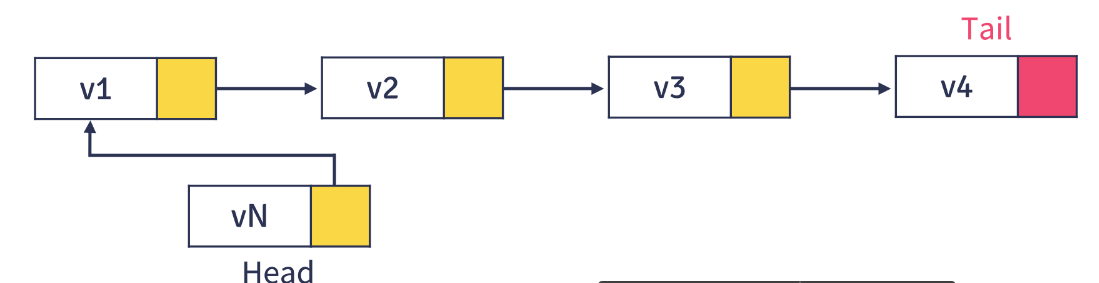
Kita analogikan memiliki satu baris single linked list dengan head berada di v1. Kita ingin menambahkan node bari yaitu vN menjadi head.



Kita ibaratkan variable baru vN sudah disiapkan, akan tetapi belum terhubung dengan linked list diatas.



Kemudian kita buat pointer penghubung antara node baru vN dengan node head v1.



Maka selanjutnya kita alihkan variable pointer head menuju ke variable pointer yang dimiliki oleh node baru vN. Maka secara otomatis variable head akan menjadi milik vN dan v1 menjadi node ke 2 yang terhubung setelah head.

Fungsi fungsi menggunakan single linked list

Ada berbagai macam fungsi yang bisa dibuat dengan menggunakan linked list. Berikut beberapa fungsi yang sudah kami rangkum.

* Add First

Fungsi ini berfungdi untuk menambahkan node baru pada awal / head dalam linked list.

* Add Last

Fungsi ini berfungsi untuk menambahkan node baru pada akhir / tail dalam linked list.

* Remove First

Fungsi ini berfungsi untuk menghapus node awal / head.

* Remove Last

Fungsi ini berfungsi untuk menghapus node yang paling akhir / tail.

* Change First

Fungsi ini berfungsi untuk merubah node head menjadi node head yang baru yang sudah di inputkan.

* Change Last

Fungsi ini berfungsi untuk merubah node tail menjadi node tail yang baru yang sudah di inputkan.

# BAB III

**IMPLEMENTASI SINGLE LINKED LIST**

Notasi Algoritmik

Pada dasarnya notasi algoritmik yang dibuat pada macam fungsi single linked list sama hanya tinggal menyesuaikan tergantung dengan fungsi yang akan dibuat.

|  |
| --- |
| **Struct** “nama node”()  (deklarasi semua variable elemen yang ingin dibuat)  (“nama node” ->pointer next) |
| **Kamus lokal :**  (semua variable dalam struct yang akan dihubungkan dengan pointer next) |
| **Fungsi :** nama fungsi (parameter yang digunakan)  Algoritma yang membuat node dan sekalian dengan penghubung antara pointer dengan setiap variable parameter. |
| **Algoritma:**  Namafungsi(parameter yang digunakan)  Fungsicetak() |

Contoh program c++

#include <iostream>

using namespace std;

struct node{

//deklarasi komponen yang dibutuhkan

int angka;

node \*next;

};

node \*head, \*tail, \*cur;

void createlinkedlist(int angka){

head = new node();

head->angka = angka;

head->next = NULL;

tail = head;

}

void showlinkedlist(){

cur = head;

while (cur != NULL){

cout<<"["<<cur->angka<<"]";

cur = cur->next;

}

}

int main()

{

createlinkedlist(1);

showlinkedlist();

cout<<"->";

createlinkedlist(2);

showlinkedlist();

cout<<"->";

createlinkedlist(3);

showlinkedlist();

cout<<"->";

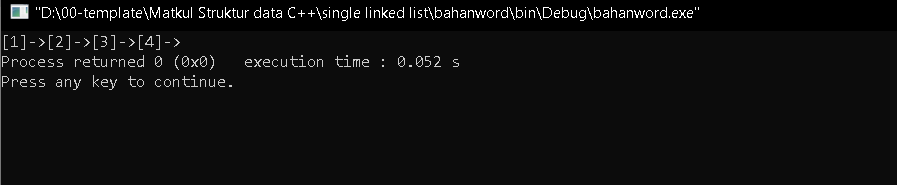
createlinkedlist(4);

showlinkedlist();

cout<<"->";

}

Hasil running:



Contoh pemrograman C++ berdasarkan review diatas

#include <iostream>

using namespace std;

//pemrograman simple menggunakan single linked list

//single linked list PBO menggunakan fungsi

struct toko{

//deklarasi komponen yang dibutuhkan

string produk, merk;

int tanggal;

toko \*next;

};

toko \*head, \*tail, \*cur, \*newnode, \*del; //curr adalah current letak node

//creat single linked list

void buatSingleLinkedList(string produk,string merk,int tb){//tb adalah tahun terbit

head = new toko();

head->produk = produk;

head->merk = merk;

head->tanggal = tb;

head->next = NULL;

tail = head;

}

//menambah awal single linked list

void addfirst(string produk,string merk,int tb){

newnode = new toko();

newnode->produk = produk;

newnode->merk = merk;

newnode->tanggal = tb;

newnode->next = head;

head = newnode;

}

void addlast(string produk,string merk,int tb){

newnode = new toko();

newnode->produk = produk;

newnode->merk = merk;

newnode->tanggal = tb;

newnode->next = NULL;

tail->next = newnode;

tail = newnode;

}

void removefirst(){

del = new toko();

head = head->next;

delete del;

}

void removelast(){

del = tail;

cur = head;

while(cur->next != tail){

cur = cur->next;

}

tail = cur;

tail->next = NULL;

delete del;

}

void changefirst(string produk,string merk,int tb){

head->produk = produk;

head->merk = merk;

head->tanggal = tb;

}

void changelast(string produk,string merk,int tb){

tail->produk = produk;

tail->merk = merk;

tail->tanggal = tb;

}

//print single linked list

void cetakSingleLinkedList(){

cur = head;

while (cur != NULL){

cout<<"produk : "<<cur->produk<<endl;

cout<<"merk : "<<cur->merk<<endl;

cout<<"tahun terbit: "<<cur->tanggal<<endl;

cur = cur->next;

}

}

int main()

{

buatSingleLinkedList("minuman","cocacola", 2019);

cetakSingleLinkedList();

cout << "\n" <<endl;

addfirst("makanan", "citato", 2020);

cetakSingleLinkedList();

cout << "\n" <<endl;

addlast("makanan", "lays", 2021);

cetakSingleLinkedList();

cout << "\n" <<endl;

removefirst();

cetakSingleLinkedList();

cout << "\n" <<endl;

removelast();

cetakSingleLinkedList();

cout << "\n" <<endl;

changefirst("minuman", "milo", 2018);

cetakSingleLinkedList();

cout << "\n" <<endl;

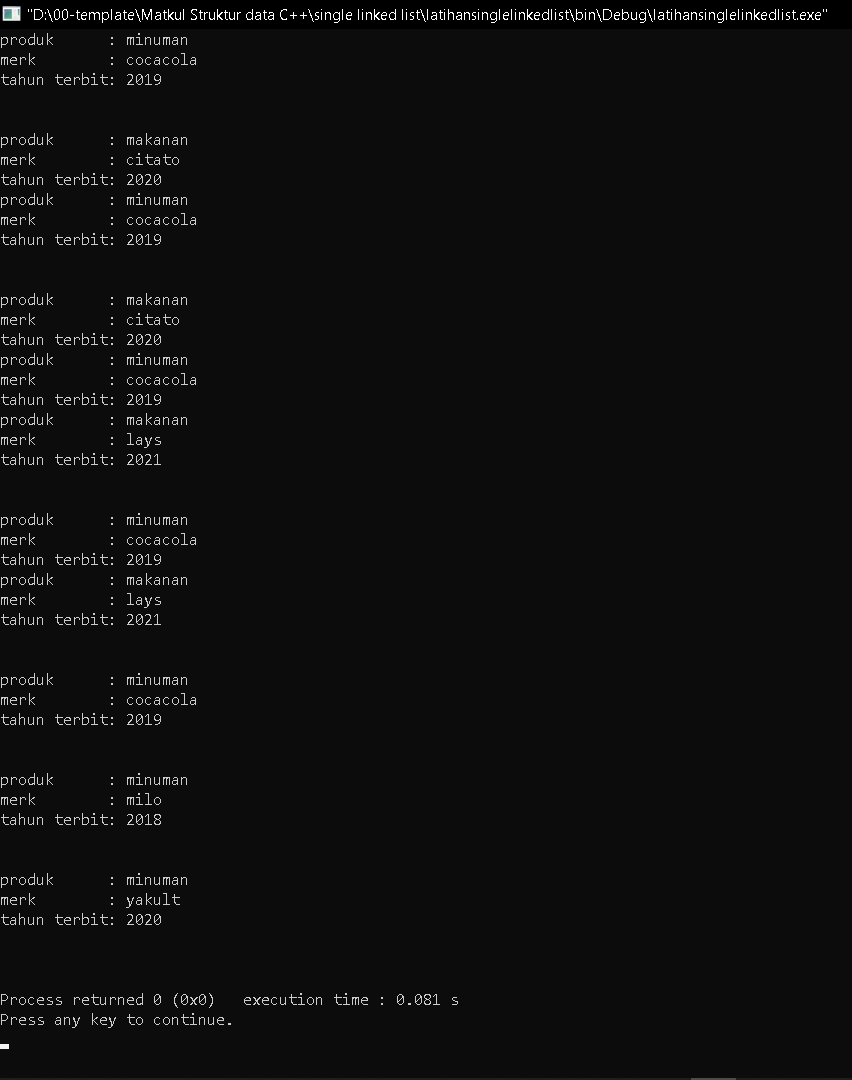
changelast("minuman", "yakult", 2020);

cetakSingleLinkedList();

cout << "\n" <<endl;

}

Hasil running:



# DAFTAR PUSTAKA

Jufri. (n.d.). MAKALAH LIST DAN MULTILIST. 1-3. Retrieved oktober 1, 2021, from https://pdfcoffee.com/makalah-list-dan-multilistdocx-pdf-free.html

Malang, U. N. (2016). MODUL 6 SINGLE & DOUBLE LINKED LIST. *MODUL PRAKTIKUM ALGORITMA & STRUKTUR DATA*. Retrieved oktober 31, 2021

Nard, F. (2018, maret 13). *Linked List*. Retrieved from word-press: https://ferrynard.wordpress.com/2018/03/13/linked-list/

Parlante, N. (1998-2001). Linked List Basics. 2-12. Retrieved november 1, 2021

Saniati, S. (2021). Single Linked List. *Struktur Data*. Retrieved November 1, 2021, from https://drive.google.com/file/d/1RCSwvJIqdDgkkooYTOrBh1ShRR4ApJD1/view

student, S. w. (2021). Single Linked List. *Konsep dan Implementasi Single Linked List Dasar di C++ | STRUKTUR DATA*. Retrieved from https://www.youtube.com/watch?v=VVemCxif9vg&t=235s

University, S. X. (2009). The Journal of Computing. (J. Meinke, Ed.) *Volume 25, Number 1*.