



## PENGERTIAN LINKED LIST

**ZALMI RANDA R**

**185100034**

Fakultas Komputer

[randa.student@umitra.ac.id](mailto:randa.student@umitra.ac.id)

### **Abstract**

Linked List adalah suatu struktur data linier. Berbeda dengan array yang juga merupakan struktur data linier dan tipe data komposit, linked list dibentuk secara dinamik. Pada saat awal program dijalankan elemen linked list belum data. Elemen linked list (disebut node) dibentuk sambil jalan sesuai instruksi. Apabila setiap elemen array dapat diakses secara langsung dengan menggunakan indeks, sebuah node linked list diakses dengan menggunakan pointer yang mengacu (menunjuk) ke node tersebut. Awal atau kepala linked list harus diacu sebuah pointer yang biasa diberi nama head. Pointer current (disingkat curr) digunakan untuk memindahkan pengacuan kepada node tertentu.

Kata Kunci : **PENGERTIAN LINKED LIST**

## A. PENDAHULUAN

Linked List atau dikenal juga dengan sebutan senarai berantai adalah struktur data yang terdiri dari urutan record data dimana setiap record memiliki field yang menyimpan alamat/referensi dari record selanjutnya (dalam urutan). Elemen data yang dihubungkan dengan link pada Linked List disebut Node. Biasanya didalam suatu linked list, terdapat istilah head dan tail.

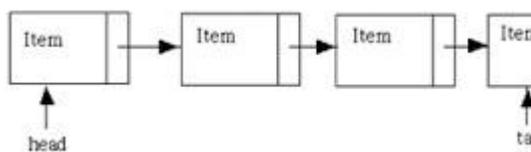
- Head adalah elemen yang berada pada posisi pertama dalam suatu linked list
- Tail adalah elemen yang berada pada posisi terakhir dalam suatu linked list

Ada beberapa macam Linked List, yaitu :

## 1. Single Linked List

Single Linked List merupakan suatu linked list yang hanya memiliki satu variabel pointer saja. Dimana pointer tersebut menunjuk ke node selanjutnya. Biasanya field pada tail menunjuk ke NULL.

contoh :



contoh codingannya :

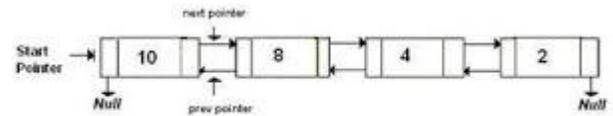
```
struct Mahasiswa{
char nama[25];
int usia;
```

```
struct Mahasiswa *next;
}*head,*tail;
```

## 2. Double Linked List

Double Linked List merupakan suatu linked list yang memiliki dua variabel pointer yaitu pointer yang menunjuk ke node selanjutnya dan pointer yang menunjuk ke node sebelumnya. Setiap head dan tailnya juga menunjuk ke NULL.

contoh :



contoh codingannya :

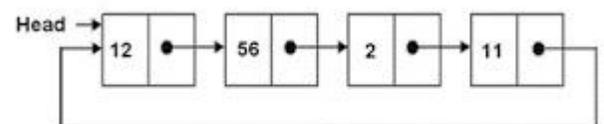
```
struct Mahasiswa{
char nama[25];
int usia;
struct Mahasiswa *next,*prev;
}*head,*tail;
```

## 3. Circular Linked List

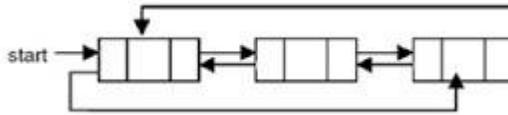
Circular Linked List merupakan suatu linked list dimana tail (node terakhir) menunjuk ke head (node pertama). Jadi tidak ada pointer yang menunjuk NULL. Ada 2 jenis Circular Linked List, yaitu :

- Circular Single Linked List

contoh :



- Circular Double Linked List  
contoh :



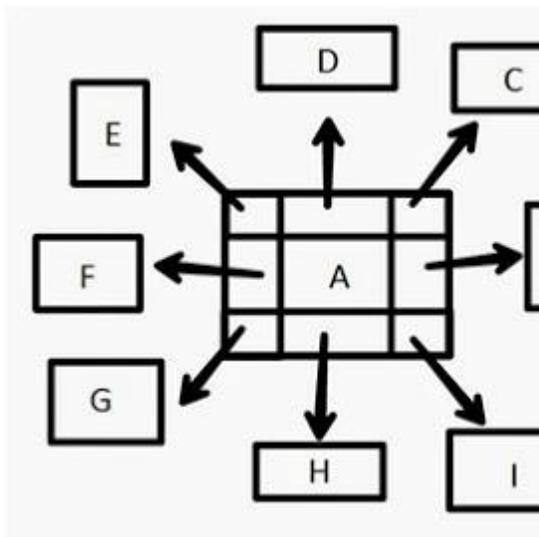
ke node berikutnya, dan juga memiliki field yang berisi data.

- Pada akhir linked list, node terakhir akan menunjuk ke NULL yang akan digunakan sebagai kondisi berhenti pada saat pembacaan isi linked list

## 4. Multiple Linked List

Multiple Linked List merupakan suatu linked list yang memiliki lebih dari 2 buah variabel pointer. contoh :

Contoh program single linked list non circular tambah list di depan :



```
# include<stdio.h>
# include<stdlib.h>
# include<conio.h>
# include<iostream.h>
# include<ctype.h>
# include<string.h>
struct simpul
{
    int angka;
    struct simpul*berikut;
};
struct simpul *awal=NULL;
int bil;
```

## 4. Non Circular linked list

Pengertian:

Single : artinya field pointer-nya hanya satu buah saja dan satu arah.

Linked List : artinya node-node tersebut saling terhubung satu sama lain.

Ilustrasinya

- Setiap node pada linked list mempunyai field yang berisi pointer

```
void tambah_list_didepan(int info);
void isi_list();
void tampil_list();
void hapus_list();
```

```
void main ()
```

```
{
```

```
clrscr();
isi_list();
clrscr();
```

```
tampil_list();
hapus_list();
getch();
}

void tambah_list_didepan(int info)
{
    struct simpul *baru;
    baru=(struct simpul
*)malloc(sizeof(struct simpul));
    baru->angka=info;

    baru->berikut=awal;

    awal=baru;
}

void isi_list()
{
    char jawab;
    do
    {
        clrscr();
        cout<<"\ninput bilangan :";
        cin>>bil;
        tambah_list_didepan(bil);
        cout<<"\ntambah data Y/T : " ;
        cin>>jawab;
    }
    while (toupper(jawab)!='T');
}
```

```
void tampil_list()
{
    struct simpul* baca;
    int i;
    baca=awal;
    i=1;

    while(baca!=NULL)
    {
        cout<<"\nbilangan ke-
"<<i<<"yang dibaca :"<<baca->angka;
        i++;
        baca=baca->berikut;
    }
}

void hapus_list()
{
    struct simpul*hapus;
    hapus=awal;
    while(hapus!=NULL)
    {
        awal=hapus->berikut;
        free(hapus);
        hapus=awal;
    }
}
```

## B. PEMBAHASAN/STUDI KASUS

Linked list adalah sekumpulan elemen bertipe sama, yang mempunyai keterurutan tertentu, yang setiap elemennya terdiri dari dua bagian. Linked list juga merupakan suatu cara untuk menyimpan data dengan struktur sehingga dapat secara otomatis menciptakan suatu tempat baru untuk menyimpan data yang diperlukan. Struktur ini lebih dinamis karena banyaknya elemen dengan mudah ditambah atau dikurangi, berbeda dengan array yang ukurannya tetap. Berikut gambaran kecil mengenai linked list.

Ada beberapa macam Linked List, yaitu :

1. Single Linked List
2. Double Linked List
3. Operasi Pada Single Linked List

### Single Linked List

Single Linked List merupakan suatu linked list yang hanya memiliki satu variabel pointer saja. Dimana pointer tersebut menunjuk ke node selanjutnya. Biasanya field pada tail menunjuk ke NULL.

### Double Linked List

Double Linked List merupakan suatu linked list yang memiliki dua variabel pointer yaitu pointer yang menunjuk ke node selanjutnya dan pointer yang menunjuk ke node sebelumnya. Setiap head dan tailnya juga menunjuk ke NULL.

### Operasi Pada Single Linked List

1. **Insert** = Istilah Insert berarti menambahkan sebuah simpul baru ke dalam suatu linked list.
2. **Konstruktor** = Fungsi ini membuat sebuah linked list yang baru dan masih kosong.
3. **IsEmpty** = Fungsi ini menentukan apakah linked list kosong atau tidak.
4. **Find First** = Fungsi ini mencari elemen pertama dari linked list
5. **Find Next** = Fungsi ini mencari elemen sesudah elemen yang ditunjuk now.
6. **Retrieve** = Fungsi ini mengambil elemen yang ditunjuk oleh now. Elemen tersebut lalu dikembalikan oleh fungsi.

7. **Update** = Fungsi ini mengubah elemen yang ditunjuk oleh now dengan isi dari sesuatu.
8. **Delete Now** =  
Fungsi ini menghapus elemen yang ditunjuk oleh now. Jika yang dihapus adalah elemen pertama dari linked list (head), head akan berpindah ke elemen berikut.

## 2. JELASKAN FUNGSI STACK

### Pengertian Stack

Stack adalah metode atau teknik dalam menyimpan atau mengambil data ke dan dari memori. Stack dapat diibaratkan sebuah tumpukan dalam suatu tempat yang hanya memiliki satu pintu di atasnya. kaidah stack adalah First In Last Out atau Last In First Out (LIFO), benda yang terakhir masuk akan menjadi benda yang pertama dikeluarkan.

### Stack dengan Array

Sesuai dengan sifat stack, maka pengambilan atau penghapusan elemen dalam stack harus dimulai

dari elemen teratas. Ilustrasinya sebagai berikut:

Dapat dilihat dalam visualisasi stack dengan array, dimana index array ke-1 diisi oleh TV, index ke-2 diisi oleh VCD dan index ke-3 diisi oleh COMBO.

TOP dianggap sebagai puncak dari stack. Harus diingat bahwa banyak stack yang mungkin harus dibatasi karena alokasi memory pada array yang bersifat statis dan terbatas.

### C. ID SECURITY

QB TD 4452377-ASP-5244107

### D. KESIMPULAN

pengertian dari list sendiri adalah koleksi dari obyek-obyek dengan sifat setiap elemen memiliki penerus dan setiap elemen memiliki pendahulu.

Kesimpulan, linked-list terdiri dari 2 bagian, yaitu data dia sendiri dan alamat dari data lainnya.

Singly linked-list berarti linked-list yang setiap datanya memegang satu alamat dari data lain. (sesudah data tersebut)

Doubly linked-list berarti linked-list yang setiap datanya memegang dua alamat dari data lain. (sebelum dan sesudah data tsb)

Circular linked-list berarti linked-list yang head dan tailnya dihubungkan kembali, jadi data pertama berhubungan dengan data terakhir. Dalam hal ini sangat diperlukan kehati-hatian, karena bisa saja pencarian data mengakibatkan "looping forever".

## E. DISKUSI

SAYA : APAKAH ARTIKEL INI BERGUNA?

ARIA : Menurut saya artikel ini sangat berguna karena saya dapat mempelajari lebih dalam lagi tentang linked list.

SAYA : APAKAH KEKURANGAN ARTIKEL INI?

FAJAR : Menurut saya kekurangan artikel ini adalah kata-kata nya masih sedikit berantakan.

[1] A. S. Putra And O. M. Febriani, "Knowledge Management Online Application In Pdam Lampung Province," In *Prosiding International Conference On Information Technology And Business (Icibt)*, 2018, Pp. 181–187.

[2] A. S. Putra, O. M. Febriani, And B. Bachry, "Implementasi Genetic Fuzzy System Untuk Mengidentifikasi Hasil Curian Kendaraan Bermotor Di Polda Lampung," *J. Sist. Inf. Dan Manaj. Basis Data*, Vol. 1, No. 1, Pp. 21–30, 2018.

[3] O. M. Febriani And A. S. Putra, "Sistem Informasi Monitoring

Inventori Barang Pada Balai Riset Standardisasi Industri Bandar Lampung," *J. Inform.*, Vol. 13, No. 1, Pp. 90–98, 2014.

[4] Putra, Arie Setya. "2018 Artikel Struktur Data, Audit Dan Jaringan Komputer." (2018).

[5] Putra, A. S. (2018, July 17). Paperplain Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia. Retrieved From [Osf.io/Pbrn9](https://osf.io/Pbrn9).

## F. REFERENCE (Based APA )

Putra, A. S., Aryanti, D. R., & Hartati, I. (2018, November). Metode SAW (Simple Additive Weighting) sebagai Sistem Pendukung Keputusan Guru Berprestasi (Studi Kasus: SMK Global Surya). In *Prosiding Seminar Nasional Darmajaya* (Vol. 1, No. 1, pp. 85-97).

Sari, D. P., Febriani, O. M., & Putra, A. S. (2018, November). Perancangan Sistem Informasi SDM Berprestasi pada SD Global Surya. In *Prosiding Seminar Nasional Darmajaya* (Vol. 1, No. 1, pp. 289-294).

Putra, A. S. (2018). Paperplain: Execution Fundamental Create Application With Borland Delphi 7.0 University Of Mitra Indonesia.

Putra, A. S., Sukri, H., & Zuhri, K. Sistem Monitoring Realtime Jaringan Irigasi Desa (JIDES) Dengan Konsep Jaringan Sensor Nirkabel. *IJEIS (Indonesian Journal of Electronics and Instrumentation Systems)*, 8(2), 221-232.

Darmawan, A., Yuliawati, D., Marcella, O., & Firmandala, R. (2016). Sistem Absensi dan Pelaporan Berbasis Fingerprint dan SMS Gateway. *EXPLORE*, 7(1).

Febriani, O. M., Wahyuni, T., & Yusuf, S. (2017). DESIGN OF WEBSITE-BASED INFORMATION SYSTEM FOR EDOCUMENT ADMINISTRASI IN THE COMMUNITY SERVICE UNIT (A Case Study at Rajabasa District). *INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY*, 16(7), 7010-7020.

Febriani, O. M., & Wahyuni, T. (2017, October). PERANCANGAN SISTEM E-DOCUMENT ADMINISTRASI LOGBOOK PENELITIAN PADA UNIT LAYANAN DI BANDAR LAMPUNG. In *Prosiding Seminar Nasional Darmajaya* (Vol. 1, No. 1, pp. 187-194).

Febriani, O. M., & Permadi, A. B. (2017). Implementasi Sistem Aplikasi Data Bimbingan dan Pelanggaran Siswa pada Sekolah Menengah Atas di Lampung Tengah dengan Metode Analisis dan Desain Sistem Terdistribusi (SSAD). *EXPERT*, 7(1).

Febriani, O. M., & Ambarwati, L. (2015). PERANCANGAN APLIKASI

PENGOLAHAN DATA PENJUALAN UKM KELANTING KHAS TELO DESA SIDOHARJO KECAMATAN JATI AGUNG KABUPATEN LAMPUNG SELATAN. *Jurnal Teknologi Informasi dan Bisnis Pengabdian Masyarakat Darmajaya*, 1(1), 77-95.

Febriani, O. M. (2015). Rancang Bangun Aplikasi E-commerce menggunakan Freewebstore pada UKM Kelanting di Desa Sidoharjo Lampung Selatan. *Prosiding Sembistek 2014*, 1(02), 446-458.