**LAPORAN PRAKTIKUM**

**ALGORITMA DAN STRUKTUR DATA**

**IMPLEMENTASI LINKED LIST**



**DISUSUN OLEH:**

**Filfimo Yulfiz Ahsanul Hulqi**

**(19102143)**

|  |  |
| --- | --- |
|  |  |

Dikumpulkan Tanggal :

Dosen Pengampu :

Ipam Fuaddina Adam, S.T., M.Kom.

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

### FAKULTAS INFORMATIKA

**INSTITUT TEKNOLOGI TELKOM PURWOKERTO 2020**

1. Tujuan Praktikum

Mahasiswa diharapkan dapat :

1. Memahami konsep linked list
2. Membuat program dari linked list
3. Dasar Teori
4. Pengertian

Dalam ilmu computer, daftar tertaut adalah kumpulan linier elemen data, yang urutannya tidak diberikan oleh penempatan fisik mereka dalam memori. Sebaliknya, setiap elemen menunjuk ke yang berikutnya. Ini adalah struktur data yang terdiri dari kumpulan node yang bersama mewakili urutan. Dalam bentuknya yang paling mendasar, setiap simpul berisi : Data, dan Reference ( dengan kata lain, tautan) ke simpul berikutnya dalam urutan Struktur ini memungkinkan penyisipan atau penghapusan elemen yang efisien dari posisi apa pun dalam urutan. Varian yang lebih kompleks menambah tautan tambahan, yang memungkinkan penyisipan atau penghapusan node yang lebih efisen pada posisi sewenang-wenang. Kelemahan dari *linked list* adalah bahwa waktu akses adalah linier (dan sulit untuk disalurkan). Array memiliki lokalitas cache yang lebih baik dibandingkan dengan list yang ditautkan

1. Cara Kerja

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** |

Data Array disimpan sebagai satu potong memori yang teralokasi secara berurutan yang terbagi secara logis. Data yang disimpan di dalam array ditempatkan di dalam salh satu dari bagian- bagian itu dan dirujuk dengan lokasinya atau indeksnya di dalam array

Ini adalah cara yang baik untuk menyimpan data. Kebanyakan Bahasa pemrograman memudahkan kegiatan alokasi array dan operasi pada konten dari array itu. Penyimpanan data yang berurutan meberikan keuntungan secara performa (alokasi data), kemudahan dalam melakukan literasi pada data, dan data itu bias diakses secara langsung melalui indeksnya (random) secara konstan.

Dan ada saat di mana array bukanlah solusi ideal jika program memiliki tuntutan khusus

1. Penjelasan

struct Node

{

string data;

int posisi;

int teks;

struct Node\* next;

};

struct Node \*a = NULL;

int line=1;

Struct untuk membuat Node

void sortedInsert(struct Node\*\*, struct Node\*);

void insertionSort(struct Node \*\*head\_ref){

struct Node \*sorted = NULL;

struct Node \*current = \*head\_ref;

while (current != NULL){

struct Node \*next = current->next;

sortedInsert(&sorted, current);

current = next;

}

\*head\_ref = sorted;

}

void sortedInsert(struct Node\*\* head\_ref, struct Node\* new\_node){

struct Node\* current;

if (\*head\_ref == NULL || (\*head\_ref)->data >= new\_node->data) {

new\_node->next = \*head\_ref;

\*head\_ref = new\_node;

}

else{

current = \*head\_ref;

while (current->next!=NULL &&

current->next->data < new\_node->data){

current = current->next;

}

new\_node->next = current->next;

current->next = new\_node;

}

}

Pembuatan beberapa fungsi yaitu

Void sortedinsert, untuk mengubah node

Void insertsort, sesuai namanya yaitu sort, berfungsi untuk mengurtkan list yang telah dibuat menggunakan insertion sort sebagai pengurutannya

Void sortedinsert, Fungsi untuk menyisipkan new\_node dalam daftar. Fungsi ini mengharapkan dari pointer ke head\_ref karena dapat memodifikasi/merubah head dari input linked list (similar to push()).

void printList(struct Node \*head)

{

struct Node \*temp = head;

char abjad;

while(temp != NULL)

{

abjad = temp->data[0];

cout<< "["<<temp->data[0]<< "]"<<endl;

while(temp != NULL){

if(temp->data[0] == abjad){

cout<< "'"<< temp->data << "'"<< " Posisi :"<< temp->posisi<<" Teks Ke-> "<< temp->teks << endl;

temp = temp->next;

}

else{

break;

}

} cout<<endl;

}

system("pause");

}

Void Printlist, adalah fungsi untuk print hasil dari linked list yang nantinya akan dipanggil

void push(struct Node\*\* head\_ref, string new\_data, int new\_posisi, int new\_teks){

struct Node\* new\_node = new Node;

new\_node->data = new\_data;

new\_node->posisi = new\_posisi;

new\_node->teks = new\_teks;

new\_node->next = (\*head\_ref);

(\*head\_ref) = new\_node;

}

Void push, fungsi ini untuk memasukan simpul atau data di list.

Vector, untuk memnetukan urutan pasangan huruf disesuaikan dengan abjad untuk kata yang diinputkan supaya outputnya sama dengan arahan modul.

vector<string> explode( const string &delimiter, const string &str){

vector<string> arr;

int strleng = str.length();

int delleng = delimiter.length();

if (delleng==0)

return arr;//no change

int i=0;

int k=0;

while( i<strleng ){

int j=0;

while (i+j<strleng && j<delleng && str[i+j]==delimiter[j])

j++;

if (j==delleng)//found delimiter{

arr.push\_back( str.substr(k, i-k) );

i+=delleng;

k=i;

}

else

{

i++;

}

}

arr.push\_back( str.substr(k, i-k) );

return arr;

}

bool search\_text(Node\* head, string x){

Node\* current = head; // Initialize current

while (current != NULL)

{

if (current->data == x){

cout<< " '' " << current->data << " '' "<< " Posisi :"<< current->posisi <<" Teks Ke-> "<< current->teks <<endl;

return true;

}

current = current->next;

}

return false;

}

Search Text, digunakan untuk memcari kata yang nanti telah diinputkan oleh user/dari list

char menu(){

char inputan;

cout << endl;

cout << "=========================" << endl;

cout << "PROGRAM IMPLEMENTASI LIST" << endl;

cout << "=========================" << endl << endl;

cout << "========== MENU ==========" << endl;

cout << "<i> Input File Text " << endl;

cout << "<s> Search File " << endl;

cout << "<p> Tampilkan Isi File " << endl;

cout << "<e> Exit " << endl;

cout << "==========================" << endl << endl;

cout << "-- Masukan Pilihan --> ";

cin >> inputan;

cout << endl;

return inputan;

}

int main(){

char pilih = menu();

if(pilih=='i'){

string str;

cout<< "Masukan Text : ";

cin.ignore();

getline(cin, str);

transform(str.begin(), str.end(), str.begin(), ::tolower);

vector<string> v = explode(" ", str);

for(int i=0; i<v.size();i++){

push(&a, v[i], i+1, line);

}

line++;

}

else if(pilih=='s'){

string kunci;

cout<< "Masukan Kata Kunci :";

cin.ignore();

getline(cin,kunci);

search\_text(a, kunci);

system("pause");

}

Kemudian untuk menu programnya sendiri, tidak terlalu banyak mengalami penambahan sebuah fungsi atau prosedur, dari semester 1. Disini seperti biasanya jika user memilih menu yang disediakan maka akan muncul menu yang telah dibuat, disini kita ambil contoh I, selanjutnya user akan, diminta kembali untuk memasukan kata yang ingin disimpan, jika sudah, program akan kembali ke menu awal, dan meminta input lagi, program akan berhenti jika user menginputkan e sebagai tombola tau fungsi exit.

else if(pilih=='p'){

insertionSort(&a);

printList(a);

}

else if(pilih=='e'){

exit(0);

}

system("cls");

return main();

}

1. Kesimpulan

Kesimpulan yang dapat diambil dari praktikum kali ini adalah :

1. Fungsi dan prosedur yang telah dipelajari makin dapat dikembangkan lagi dan digabungkan dengan materi saat ini sehingga membuat program yang lebih kompleks.
2. Linked list hampir sama dengan array, tapi berbeda secara struktur dan penggunaan, contohnya dalam hal alokasi memori yang mana sudah jelas berbeda.