**LAPORAN PRAKTIKUM   
ALGORITMA & STRUKTUR DATA  
MODUL 4**



**Double Linked List**

|  |  |
| --- | --- |
| **Oleh:** | |
| **Rizki Adhitiya Maulana** | **NIM. 2410817110014** |

**PROGRAM STUDI TEKNOLOGI INFORMASI   
FAKULTAS TEKNIK   
UNIVERSITAS LAMBUNG MANGKURAT   
MEI 2025**

# LEMBAR PENGESAHAN LAPORAN PRAKTIKUM ALGORITMA & STRUKTUR DATA MODUL 4

Laporan Praktikum Algoritma & Struktur Data Modul 4 : Double Linked List ini disusun sebagai syarat lulus mata kuliah Praktikum Algoritma & Struktur Data. Laporan Praktikum ini dikerjakan oleh:

Nama Praktikan : Rizki Adhitiya Maulana

NIM : 2410817110014

|  |  |
| --- | --- |
| Menyetujui,  Asisten Praktikum  Muhammad Fauzan Ahsani  NIM. 2310817310009 | Mengetahui,  Dosen Penanggung Jawab Praktikum  Muti’a Maulida, S.Kom., M.TI.  NIP. 198810272019032013 |

# DAFTAR ISI

[LEMBAR PENGESAHAN i](#_Toc198583073)

[DAFTAR ISI ii](#_Toc198583074)

[DAFTAR TABEL iii](#_Toc198583075)

[DAFTAR GAMBAR iv](#_Toc198583076)

[SOAL 1 1](#_Toc198583077)

[A Source Code 6](#_Toc198583078)

[B Output Program 14](#_Toc198583079)

[C Pembahasan 22](#_Toc198583080)

[SOAL 2 27](#_Toc198583081)

[A Pembahasan 27](#_Toc198583082)

[SOAL 3 28](#_Toc198583083)

[A Pembahasan 28](#_Toc198583084)

[SOAL 4 29](#_Toc198583085)

[A Output Program 29](#_Toc198583086)

[B Pembahasan 31](#_Toc198583087)

[TAUTAN GIT HUB 32](#_Toc198583088)

# DAFTAR TABEL

[Tabel 1 Source Code Linked List 6](#_Toc198583206)

# DAFTAR GAMBAR

[Gambar 1 Tampilan Menu Double Linked List Non Circular 14](#_Toc198583276)

[Gambar 2 Masuk Ke Tampilan Menu Head DLLNC 14](#_Toc198583277)

[Gambar 3 Tambah Data Dari Depan 14](#_Toc198583278)

[Gambar 4 Tampilan Data Setelah Dilakukan Tambah Depan 15](#_Toc198583279)

[Gambar 5 Tambah Data Dari Belakang 15](#_Toc198583280)

[Gambar 6 Tampilan Data Setelah Dilakukan Tambah Belakang 15](#_Toc198583281)

[Gambar 7 Hapus Data Dari Depan 16](#_Toc198583282)

[Gambar 8 Tampilan Data Setelah Dilakukan Hapus Depan 16](#_Toc198583283)

[Gambar 9 Hapus Data Dari Belakang 16](#_Toc198583284)

[Gambar 10 Tampilan Data Setelah Dilakukan Hapus Belakang 17](#_Toc198583285)

[Gambar 11 Reset Data Yang Ada 17](#_Toc198583286)

[Gambar 12 Tampilan Data Setelah Dilakukan Reset Data 17](#_Toc198583287)

[Gambar 13 Masuk Ke Tampilan Menu Head Dan Tail DLLNC 18](#_Toc198583288)

[Gambar 14 Tambah Data Dari Depan 18](#_Toc198583289)

[Gambar 15 Tampilan Data Setelah Dilakukan Tambah Depan 18](#_Toc198583290)

[Gambar 16 Tambah Data Dari Belakang 19](#_Toc198583291)

[Gambar 17 Tampilan Data Setelah Dilakukan Tambah Belakang 19](#_Toc198583292)

[Gambar 18 Hapus Data Dari Depan 19](#_Toc198583293)

[Gambar 19 Tampilan Data Setelah Dilakukan Hapus Depan 20](#_Toc198583294)

[Gambar 20 Hapus Data Dari Belakang 20](#_Toc198583295)

[Gambar 21 Tampilan Data Setelah Dilakukan Hapus Belakang 20](#_Toc198583296)

[Gambar 22 Reset Data Yang Ada 21](#_Toc198583297)

[Gambar 23 Tampilan Data Setelah Dilakukan Reset Data 21](#_Toc198583298)

[Gambar 24 Tampilan Keluar dari Program 21](#_Toc198583299)

[Gambar 25 Fungsi Next Di Single Linked List 27](#_Toc198583300)

[Gambar 26 Fungsi Prev Di Double Linked List 28](#_Toc198583301)

[Gambar 27 Tampilan Tambah Depan Head 29](#_Toc198583302)

[Gambar 28 Tampilan Tambah Belakang Head 29](#_Toc198583303)

[Gambar 29 Tampilan Data Head 30](#_Toc198583304)

[Gambar 30 Tampilan Tambah Depan Head Dan Tail 30](#_Toc198583305)

[Gambar 31 Tampilan Tambah Belakang Head Dan Tail 30](#_Toc198583306)

[Gambar 32 Tampilan Data Head Dan Tail 31](#_Toc198583307)

# SOAL 1

Lengkapi coding pada function tambahDepanH() agar bisa berjalan dengan lancar. running, simpan program, dan screenshoot hasil running!

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated

## Source Code

Tabel 1 Source Code Linked List

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  98  99  100  101  102  103  104  105  106  107  108  109  110  111  112  113  114  115  116  117  118  119  120  121  122  123  124  125  126  127  128  129  130  131  132  133  134  135  136  137  138  139  140  141  142  143  144  145  146  147  148  149  150  151  152  153  154  155  156  157  158  159  160  161  162  163  164  165  166  167  168  169  170  171  172  173  174  175  176  177  178  179  180  181  182  183  184  185  186  187  188  189  190  191  192  193  194  195  196  197  198  199  200  201  202  203  204  205  206  207  208  209  210  211  212  213  214  215  216  217  218  219  220  221  222  223  224  225  226  227  228  229  230  231  232  233  234  235  236  237  238  239  240  241  242  243  244  245  246  247  248  249  250  251  252  253  254  255  256  257  258  259  260  261  262  263  264  265  266  267  268  269  270  271  272  273  274  275  276  277  278  279  280  281  282  283  284  285  286  287  288  289  290  291  292  293  294  295  296  297  298  299  300  301  302  303  304  305  306  307  308  309  310  311  312  313  314  315  316  317  318  319  320  321  322  323  324  325  326  327  328  329  330  331  332  333  334  335  336  337  338  339  340  341  342  343  344  345  346  347  348  349  350  351  352  353  354  355 | #include <conio.h>  #include <iostream>  #include <stdlib.h>  using namespace std;  typedef struct Tnode {      string data;      Tnode \*next;      Tnode \*prev;  };  Tnode \*head, \*tail;  int pil, menu;  char pilihan[1];  string dataBaru;  void initH();  void initHT();  int isEmptyH();  int isEmptyHT();  void tambahDepanH();  void tambahDepanHT();  void tambahBelakangH();  void tambahBelakangHT();  void hapusDepanH();  void hapusDepanHT();  void hapusBelakangH();  void hapusBelakangHT();  void tampilkanH();  void tampilkanHT();  void clearH();  void clearHT();  int main()  {      menu:      cout<<"Double Linked List Non Circular (DLLNC)"<<endl;      cout<<"======================================="<<endl;      cout<<"Silahkan pilihan program DLLNC yang ingin dijalankan!"<<endl;      cout<<"1. DLLNC dengan head"<<endl;      cout<<"2. DLLNC dengan Head dan Tail"<<endl;      cout<<"3. Quit"<<endl;      cout<<"Pilihan"<<endl;      cin>>menu;      system("cls");      if(menu==1){          do {              cout<<"Double Linked List Non Circular (DLLNC) (head)"<<endl;              cout<<"=============================================="<<endl;              cout<<"1. Tambah Depan"<<endl;              cout<<"2. Tambah Belakang"<<endl;              cout<<"3. Tampilkan Data"<<endl;              cout<<"4. Hapus Depan"<<endl;              cout<<"5. Hapus Belakang"<<endl;              cout<<"6. Reset"<<endl;              cout<<"7. Kembali Ke Menu"<<endl;              cout<<"Pilihan : "<<endl;              cin>>pilihan;              pil=atoi(pilihan);              switch (pil){              case 1:                  tambahDepanH();                  break;              case 2:                  tambahBelakangH();                  break;              case 3:                  tampilkanH();                  break;              case 4:                  hapusDepanH();                  break;              case 5:                  hapusBelakangH();                  break;              case 6:                  clearH();                  break;              default:                  system("cls");                  goto menu;              }              cout<<"\npress any key to continue"<<endl;              getch();              system("cls");          } while (pil<7);      } else if(menu==2){          do {              cout<<"Double Linked List Non Circular (DLLNC) (head dan Tail)"<<endl;              cout<<"=============================================="<<endl;              cout<<"1. Tambah Depan"<<endl;              cout<<"2. Tambah Belakang"<<endl;              cout<<"3. Tampilkan Data"<<endl;              cout<<"4. Hapus Depan"<<endl;              cout<<"5. Hapus Belakang"<<endl;              cout<<"6. Reset"<<endl;              cout<<"7. Kembali Ke Menu"<<endl;              cout<<"Pilihan : "<<endl;              cin>>pilihan;              pil=atoi(pilihan);              switch (pil){              case 1:                  tambahDepanHT();                  break;              case 2:                  tambahBelakangHT();                  break;              case 3:                  tampilkanHT();                  break;              case 4:                  hapusDepanHT();                  break;              case 5:                  hapusBelakangHT();                  break;              case 6:                  clearHT();                  break;              default:                  system("cls");                  goto menu;              }              cout<<"\npress any key to continue"<<endl;              getch();              system("cls");          } while (pil<7);      } else {          cout<<"\nTERIMA KASIH"<<endl;          cout<<"Program was made by Rizki Adhitiya Maulana (2410817110014)"<<endl;      }  }  void initH(){      head=NULL;  }  void initHT(){      head=NULL;      tail=NULL;  }  int isEmptyH(){      if(head == NULL) return 1;      else return 0;  }  int isEmptyHT(){      if(tail == NULL) return 1;      else return 0;  }  void tambahDepanH(){      cout<<"Masukan Data : ";      cin>>dataBaru;      Tnode \*baru;      baru = new Tnode;      baru->data = dataBaru;      baru->next = NULL;      baru->prev = NULL;      if(isEmptyH() == 1) {          head = baru;      } else {          baru->next = head;          head->prev = baru;          head = baru;      }      cout<<"Data \""<<dataBaru<<"\" berhasil dimasukan di bagian depan.";  }  void tambahDepanHT() {      cout<<"Masukan Data : ";      cin>>dataBaru;      Tnode \*baru;      baru = new Tnode;      baru->data = dataBaru;      baru->next = NULL;      baru->prev = NULL;      if(isEmptyHT() == 1) {          head = baru;          tail = baru;      } else {          baru->next = head;          head->prev = baru;          head = baru;      }      cout<<"Data \""<<dataBaru<<"\" berhasil dimasukan di bagian depan.";  }  void tambahBelakangH() {      cout<<"Masukan Data : ";      cin>>dataBaru;      Tnode \*baru, \*bantu;      baru = new Tnode;      baru->data = dataBaru;      baru->next = NULL;      baru->prev = NULL;      if(isEmptyH() == 1) {          head = baru;      } else {          bantu = head;          while(bantu->next != NULL) {              bantu = bantu->next;          }          bantu->next = baru;          baru->prev = bantu;      }      cout <<"Data \""<<dataBaru<<"\" berhasil dimasukan di bagian belakang.";  }  void tambahBelakangHT() {      cout<<"Masukan Data : ";      cin>>dataBaru;      Tnode \*baru;      baru = new Tnode;      baru->data = dataBaru;      baru->next = NULL;      baru->prev = NULL;      if(isEmptyHT() == 1) {          head = baru;          tail = baru;      } else {          tail->next = baru;          baru->prev = tail;          tail = baru;      }      cout<<"Data \""<<dataBaru<<"\" berhasil dimasukan di bagian belakang.";  }  void tampilkanH() {      Tnode \*bantu;      bantu = head;      if(isEmptyH() == 0) {          while(bantu != NULL) {              cout<<bantu->data<<' ';              bantu = bantu->next;          }          cout<<endl;      } else cout<<"Tidak terdapat data pada Linked List";  }  void tampilkanHT() {      Tnode \*bantu;      bantu = head;      if(isEmptyHT() == 0) {          while(bantu != tail->next) {              cout<<bantu->data<<' ';              bantu = bantu->next;          }          cout<<endl;      } else cout<<"Tidak terdapat data pada Linked List";  }  void hapusDepanH() {      Tnode \*hapus;      string data;      if(isEmptyH() == 0) {          hapus = head;          data = hapus->data;          if(head->next != NULL) {              head = head->next;              head->prev = NULL;          } else {              initH();          }          delete hapus;          cout<<"Data \""<<data<<"\" berhasil dihapus dari bagian depan."<<endl;      } else cout<<"Tidak ada data pada Linked List"<<endl;  }  void hapusDepanHT() {      Tnode \*hapus;      string data;      if(isEmptyHT() == 0) {          hapus = head;          data = hapus->data;          if(head->next != NULL) {              head = head->next;              head->prev = NULL;          } else {              initHT();          }          delete hapus;          cout<<"Data \""<<data<<"\" berhasil dihapus dari bagian depan."<<endl;      } else cout<<"Tidak ada data pada Linked List"<<endl;  }  void hapusBelakangH() {      Tnode \*hapus;      string data;      if(isEmptyH() == 0) {          hapus = head;          while(hapus->next != NULL) {              hapus = hapus->next;          }          data = hapus->data;          if(head->next != NULL) {              hapus->prev->next = NULL;          } else {              initH();          }          delete hapus;          cout<<"Data \""<<data<<"\" berhasil dihapus dari bagian belakang."<<endl;      } else cout<<"Tidak ada data pada Linked List"<<endl;  }  void hapusBelakangHT() {      Tnode \*hapus;      string data;      if(isEmptyHT() == 0) {          hapus = tail;          data = hapus->data;          if(head->next != NULL) {              tail = tail->prev;              tail->next = NULL;          } else {              initHT();          }          delete hapus;          cout<<"Data \""<<data<<"\" berhasil dihapus dari bagian belakang."<<endl;      } else cout<<"Tidak ada data pada Linked List"<<endl;  }  void clearH() {      Tnode \*bantu, \*hapus;      bantu = head;      while(bantu != NULL) {          hapus = bantu;          bantu = bantu->next;          delete hapus;      }      initH();      cout<<"Seluruh data pada Linked List telah dibersihkan"<<endl;  }  void clearHT() {      Tnode \*bantu, \*hapus;      bantu = head;      while(bantu != NULL) {          hapus = bantu;          bantu = bantu->next;          delete hapus;      }      initHT();      cout<<"Seluruh data pada Linked List telah dibersihkan"<<endl;  } |

## Output Program

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Menu Double Linked List Non Circular



Gambar Masuk Ke Tampilan Menu Head DLLNC

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tambah Data Dari Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Tambah Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tambah Data Dari Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Tambah Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Hapus Data Dari Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Hapus Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Hapus Data Dari Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Hapus Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Reset Data Yang Ada

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Reset Data

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Masuk Ke Tampilan Menu Head Dan Tail DLLNC

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tambah Data Dari Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Tambah Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tambah Data Dari Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Tambah Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Hapus Data Dari Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Hapus Depan

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Hapus Data Dari Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Hapus Belakang

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Reset Data Yang Ada

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Setelah Dilakukan Reset Data

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Keluar dari Program

## Pembahasan

Pada baris [1] sampai [3] terdapat *#include* yang mana digunakan untuk mengakses sebuah file yang diinginkan. *<iostream>* yang ada digunakan untuk input dan output. Kemudian *<conio.h>* digunakan agar menyediakan fungsi fungsi yang berguna ketika ada interaksi langsung dengan keyboard, tanpa perlu menekan Enter. Terus *<stdlib.h>* digunakan untuk fungsi fungsi manajemen memori, konversi angka, kontrol proses, dan lingkungan program.

Pada baris [5] terdapat using *namespace std;* yang mana digunakan untuk menghindari penulisan std.

Pada baris [7] terdapat [11] terdapat *struct TNode* yang mana digunakan untuk menyimpan elemen-elemen dari linked list, dimana variabel *string data* digunakan untuk menyimpan isi atau informasi dari node tersebut seperti angka yang di input ke dalam list. Terus *Tnode \*next* digunakan untuk menunjuk ke node berikutnya dalam urutan linked list, node terakhir yang ada dalam urutan linked list akan menunjuk kembali ke node pertama. Kemudian, *Tnode \*prev* digunakan untuk menunjuk ke node sebelumnya dalam urutan linked list, sehingga memungkinkan dilakukannya penelusuran dua arah, baik maju (menggunakan next) maupun mundur (menggunakan prev).

Pada baris [13] terdapat *Tnode \*head, \*tail* yang mana *TNode \*head* menunjuk pointer node pertama dan *TNode \*tail* menunjuk pointer node terakhir.

Pada baris [15] sampai [17] terdapat *int pil dan menu* yang mana digunakan untuk menyimpan variabel Integer atau bilangan bulat*.* Terus *char pilihan [1]* yang mana digunakan untuk menyimpan variabel character, ditambah array sebagai batasan input dari user*, string dataBaru* yang digunakan untuk menyimpan variabel string atau katakter.

Pada baris [19] sampai [35] terdapat penamaan fungsi yang akan dimasukkan ke dalam program Linked list.

Pada baris [37] sampai [141] terdapat int main() yang mana digunakan untuk menjalankan dan menampilkan menu SLLC. Menu yang akan ditampilkan di dalam sistem ada sebanyak 3 buah, terdiri dari menu pertama untuk DLLNC dengan head, menu kedua DLLNC dengan head dan tail, menu terakhir untuk keluar. Setiap pilihan yang ada akan menampilkan tampilan berbeda sesuai dengan fungsi yang ada di dalam *switch case* yang dimasukkan pada program yang akan dijalankan. Terdapat getch() untuk menunggu tombol yang ditekan oleh pengguna dan membersihkan layar menggunakan system("cls"). Terus program akan terus berjalan selama user tidak memilih pilihan tiga (3) untuk keluar atau menghentikan program yang ada yang terletak di menu pilihan paling awal.

Pada baris [143] sampai [145] terdapat *void initH()* yang mana digunakan untuk menginisialisasikan kondisi awal dari linked list. *InitH()* akan berfokus dalam mengatur pointer atau variabel head ke dalam keadaan NULL.

Pada baris [147] sampai [150] terdapat *void initHT()* yang mana digunakan untuk menginisialisasikan kondisi awal dari linked list. *initHT()* akan berfokus dalam mengatur pointer atau variabel head ke dalam keadaan NULL, begitu juga untuk pointer atau variabel tail yang akan disetting atau diatur kedalam keadaan NULL seperti head.

Pada baris [152] sampai [155] terdapat *isEmptyH()* yang mana digunakan untuk melakukan pengecekan pada linked list, apakah head dalam keadaan kosong atau tidak. Fungsi ini akan mengembalikan nilai 1 apabila linked list dalam keadaan kosong dan 0 apabila tidak dalam keadaan kosong.

Pada baris [157] sampai [160] terdapat *isEmptyHT()* yang mana digunakan untuk melakukan pengecekan pada linked list, apakah head atau tail dalam keadaan kosong atau tidak. Fungsi ini akan mengembalikan nilai 1 apabila linked list dalam keadaan kosong dan 0 apabila tidak dalam keadaan kosong.

Pada baris [162] sampai [178] terdapat *void tambahDepanH()* yang mana digunakan untuk menambahkan node baru ke bagian depan dari linked list, dimana pointer head yang menjadi patokannya. Apabila linked list dalam keadaan kosong, node yang baru ditambahkan akan menjadi head atau node baru. Kemudian, apabila linked list dalam keadaan tidak kosong, node yang baru ditambahkan akan menjadi head atau node pertama dan node yang sudah ada sebelumnya akan menjadi node kedua atau seterusnya.

Pada baris [180] sampai [197] terdapat *void tambahDepanHT()* yang mana digunakan untuk menambahkan node baru ke bagian depan dari linked list, dimana menggunakan head dan tail. Apabila linked list dalam keadaan kosong, head dan tail yang ada akan menunjuk ke node baru. Kemudian, apabila linked list dalam keadaan tidak kosong, node yang baru ditambahkan akan menjadi head atau node pertama dan node yang sudah ada sebelumnya akan menjadi node kedua atau seterusnya.

Pada baris [199] sampai [218] terdapat *void tambahBelakangH()* yang mana digunakan untuk menambahkan node baru ke bagian belakang dari linked list, tanpa bantuan tail yang membuat diperlukannya traversal dari head ke node terakhir, sehingga berkurangan efisiensi yang ada. Apabila linked list dalam keadaan kosong, node yang baru ditambahkan akan menjadi tail atau node terakhir. Kemudian, apabila linked list dalam keadaan tidak kosong, node yang baru ditambahkan akan menjadi tail atau node terakhir dan node yang sudah ada sebelumnya akan menjadi node sebelum node akhir.

Pada baris [220] sampai [237] terdapat *void tambahBelakangHT()* yang mana digunakan untuk menambahkan node baru ke bagian belakang dari linked list, dibantu dengan tail untuk menambahkan node dibagian terakhir, efisiensi yang ada meningkat. Apabila linked list dalam keadaan kosong, node yang baru ditambahkan akan menjadi tail atau node terakhir. Kemudian, apabila linked list dalam keadaan tidak kosong, node yang baru ditambahkan akan menjadi tail atau node terakhir dan node yang sudah ada sebelumnya akan menjadi node sebelum node akhir.

Pada baris [239] sampai [249] terdapat *void tampilkanH()* yang mana digunakan untuk menampilkan seluruh isi linked list yang ada, dimulai dari depan hingga ke belakang. Data atau node yang akan ditampilkan akan dilakukan secara traversal yaitu dari HEAD (node pertama) hingga NULL (node terakhir).

Pada baris [251] sampai [261] terdapat *void tampilkanHT()* yang mana digunakan untuk menampilkan seluruh isi linked list yang ada, dimulai dari depan hingga ke belakang. Data atau node yang akan ditampilkan akan dilakukan secara traversal yaitu dari HEAD (node pertama) hingga TAIL (node terakhir).

Pada baris [263] sampai [278] terdapat *void hapusDepanH()* yang mana digunakan untuk menghapus node pertama yang terdapat pada linked list, dimana pointer head yang menjadi patokannya. Apabila terdapat satu node saja pada linked list, maka setelah dilakukan hapus depan linked list head akan diatur menjadi kosong atau NULL menggunakan fungsi *InitH()*. Kemudian, apabila terdapat lebih dari satu node yang terdapat pada linked list, node pertama yang dihapus akan digantikan dengan elemen yang ada di setelah atau node kedua sebagai head terbaru.

Pada baris [280] sampai [295]terdapat *void hapusDepanHT()* yang mana digunakan untuk menghapus node pertama yang terdapat pada linked list, dimana menggunakan head dan tail. Apabila terdapat satu node saja pada linked list, maka setelah dilakukan hapus depan linked list head akan diatur menjadi kosong atau NULL menggunakan fungsi *InitHT()*. Kemudian, apabila terdapat lebih dari satu node yang terdapat pada linked list, node pertama yang dihapus akan digantikan dengan elemen yang ada di setelah atau node kedua sebagai head terbaru.

Pada baris [297] sampai [314] terdapat *void hapusBelakangH()* yang mana digunakan untuk menghapus node terakhir yang terdapat pada linked list, tanpa bantuan tail yang membuat diperlukannya traversal dari head ke node terakhir. Apabila terdapat satu node saja pada linked list, maka setelah dilakukan hapus belakang linked list akan diatur menjadi kosong atau NULL menggunakan fungsi InitH(). Kemudian, apabila terdapat lebih dari satu node yang terdapat pada linked list, node terakhir yang dihapus akan digantikan dengan elemen yang ada di sebelumnya atau node sebelum node akhir sebagai tail terbaru.

Pada baris [316] sampai [331] terdapat *void hapusBelakangHT()* yang mana digunakan untuk menghapus node terakhir yang terdapat pada linked list, dibantu dengan tail untuk menghapus node dibagian terakhir,. Apabila terdapat satu node saja pada linked list, maka setelah dilakukan hapus belakang linked list akan diatur menjadi kosong atau NULL menggunakan fungsi InitHT(). Kemudian, apabila terdapat lebih dari satu node yang terdapat pada linked list, node terakhir yang dihapus akan digantikan dengan elemen yang ada di sebelumnya atau node sebelum node akhir sebagai tail terbaru.

Pada baris [333] sampai [343] terdapat *void clearH()* yang mana digunakan untuk menghapus semua node yang ada pada linked list, baik dari node pertama hingga node terakhir. Setelah di lakukan penghapusan untuk semua node yang ada di dalam linked list, kemudian akan dipanggil fungsi InitH() untuk mengatur linked list ke dalam kondisi kosong.

Pada baris [345] sampai [355] terdapat *void clearHT()* yang mana digunakan untuk menghapus semua node yang ada pada linked list, baik dari node pertama hingga node terakhir. Setelah di lakukan penghapusan untuk semua node yang ada di dalam linked list, kemudian akan dipanggil fungsi InitHT() untuk mengatur linked list ke dalam kondisi kosong.

# SOAL 2

Apa fungsi next pada coding?

## Pembahasan

Fungsi next dalam struktur data linked list dan double linked list digunakan untuk menunjuk ke node berikutnya dalam urutan data. Pada single linked list, setiap node mempunyai satu pointer saja yaitu next yang akan menunjuk ke node berikutnya. Dengan demikian, penelusuran dapat dilakukan dari node pertama (head) ke node terakhir dalam satu arah saja. Pada double linked list, next tetap berfungsi untuk menunjuk ke node berikutnya, tetapi pada struktur ini terdapat penunjuk tambahan yaitu prev yang memungkinkan untuk melakukan penelusuran dua arah.

Sebuah gambar berisi teks, Font, diagram, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

Gambar Fungsi Next Di Single Linked List

# 

# SOAL 3

Apa fungsi prev pada coding?

## Pembahasan

Fungsi prev hanya terdapat pada struktur double linked list, yang mana berfungsi untuk menunjuk ke node sebelumnya dari node yang sedang diakses. Dengan menggunakan prev, penjelajahan tidak hanya dapat dilakukan dengan satu arah yaitu maju dengan menggunakan next, tetapi juga dapat dilakukan dengan arah mundur dari node akhir ke node awal. Hal ini akan sangat berguna ketika dalam keadaan dimana kita harus bergerak mundur, menghapus elemen dari belakang, atau membalikkan arah penelusuran. Tanpa menggunakan prev, seperti pada single linked list, operasi-operasi yang ada akan menjadi lebih sulit dan kurang efisien karena kita harus memulai dari awal list setiap kali melakukan penelusuran.

Sebuah gambar berisi teks, Font, diagram, cuplikan layar

Konten yang dihasilkan AI mungkin salah.

Gambar Fungsi Prev Di Double Linked List

# SOAL 4

Gantilah baris 244 dan 256 dari cout<<bantu->data<<' '; menjadi cout<<head->data<<' '; lalu jawab pertanyaan berikut :

1. Apa yang terjadi jika anda menambahkan beberapa data pada program lalu tampilkan datanya, dan screenshoot hasilnya.
2. Jelaskan mengapa hal tersebut bisa terjadi dan data apa yang ditampilkan oleh program?

## Output Program

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Tambah Depan Head

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Tambah Belakang Head

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Head

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Tambah Depan Head Dan Tail

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Tambah Belakang Head Dan Tail

Sebuah gambar berisi teks, cuplikan layar, software, Software multimedia

Konten yang dihasilkan AI mungkin salah.

Gambar Tampilan Data Head Dan Tail

## Pembahasan

Setelah dilakukan perubahan atau pergantian kode dari *cout<<bantu->data<<' ';* menjadi *cout<<head->data<<' ';*. Hal yang terjadi adalah program hanya akan mencetak data dari node pertama atau head yang terdapat dari linked list secara berulang-ulang mengikuti banyaknya data atau node yang ada di dalam linked list. Output yang keluar atau ditampilkan tidak menampilkan semua node yang dimasukkan sebelumnya dalam urutan, namun hanya mengulang data atau node pertama dalam output yang keluar. Hal ini berlaku baik di “DLLNC menggunakan head” dan “DLLNC menggunakann head dan tail” dari program yang ada.

Variabel bantu yang ada sebelumnya berguna untuk menunjukkan node perta sampai node terakhir di dalam list. Namun karena varibael bantu yang ada diganti menjadi head, program hanya akan menunjukkan node pertama dan tidak menunjukkan node selanjutnya seperti urutan masuk atau node yang ada hingga tail.

# TAUTAN GIT HUB

https://github.com/Rizki-A-M/Rizki-A-M-PRAKTIKUM\_ALGORITMA\_DAN\_STRUKTUR\_DATA.git