**LAPORAN PRAKTIKUM   
ALGORITMA & STRUKTUR DATA  
MODUL 3**



**Single 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 3

Laporan Praktikum Algoritma & Struktur Data Modul 3 : Single 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](#_Toc198554054)

[DAFTAR ISI ii](#_Toc198554055)

[DAFTAR TABEL iv](#_Toc198554056)

[DAFTAR GAMBAR v](#_Toc198554057)

[SOAL 1 1](#_Toc198554058)

[A Output Program 12](#_Toc198554059)

[B Pembahasan 18](#_Toc198554060)

[SOAL 2 23](#_Toc198554061)

[A Output Program 23](#_Toc198554062)

[B Pembahasan 23](#_Toc198554063)

[SOAL 3 24](#_Toc198554064)

[A Output Program 24](#_Toc198554065)

[B Pembahasan 24](#_Toc198554066)

[SOAL 4 25](#_Toc198554067)

[A Output Program 25](#_Toc198554068)

[B Pembahasan 25](#_Toc198554069)

[SOAL 5 26](#_Toc198554070)

[A Output Program 26](#_Toc198554071)

[B Pembahasan 26](#_Toc198554072)

[SOAL 6 27](#_Toc198554073)

[A Output Program 27](#_Toc198554074)

[B Pembahasan 27](#_Toc198554075)

[SOAL 7 28](#_Toc198554076)

[A Output Program 28](#_Toc198554077)

[B Pembahasan 28](#_Toc198554078)

[SOAL 8 29](#_Toc198554079)

[A Output Program 29](#_Toc198554080)

[B Pembahasan 29](#_Toc198554081)

[SOAL 9 30](#_Toc198554082)

[A Pembahasan 30](#_Toc198554083)

[SOAL 10 31](#_Toc198554084)

[A Pembahasan 31](#_Toc198554085)

[TAUTAN GIT HUB 32](#_Toc198554086)

# DAFTAR TABEL

[Tabel 1 Source Code Linked List 1](#_Toc198554111)

# DAFTAR GAMBAR

[Gambar 1 Tampilan Awal Program 12](#_Toc198554174)

[Gambar 2 Tampilan Tambah Data dari Depan 12](#_Toc198554175)

[Gambar 3 Tampilan Data Setelah Dilakukan Tambah Data Depan 12](#_Toc198554176)

[Gambar 4 Tampilan Tambah Data dari Belakang 13](#_Toc198554177)

[Gambar 5 Tampilan Data Setelah Dilakukan Tambah Data Belakang 13](#_Toc198554178)

[Gambar 6 Tampilan Cari Data Yang Tidak Ada 13](#_Toc198554179)

[Gambar 7 Tampilan Cari Data Yang Ada 14](#_Toc198554180)

[Gambar 8 Tampilan Hapus Data Belakang 14](#_Toc198554181)

[Gambar 9 Tampilan Data Setelah Dilakukan Hapus Data Belakang 14](#_Toc198554182)

[Gambar 10 Tampilan Hapus Setiap Data Tertentu 15](#_Toc198554183)

[Gambar 11 Tampilan Data Setelah Dilakukan Hapus Setiap Data Tertentu 15](#_Toc198554184)

[Gambar 12 Tampilan Hapus Data Depan 15](#_Toc198554185)

[Gambar 13 Tampilan Data Setelah Dilakukan Hapus Data Depan 16](#_Toc198554186)

[Gambar 14 Tampilan Sisipkan Node Sebelum Data Tertentu 16](#_Toc198554187)

[Gambar 15 Tampilan Data Setelah Dilakukan Sisipkan Node Sebelum 16](#_Toc198554188)

[Gambar 16 Tampilan Sisipkan Node Setelah Data Tertentu 17](#_Toc198554189)

[Gambar 17 Tampilan Data Setelah Dilakukan Sisipkan Node Setelah 17](#_Toc198554190)

[Gambar 18 Tampilan Hapus Semua Elemen 17](#_Toc198554191)

[Gambar 19 Tampilan Data Setelah Dilakukan Hapus Semua Elemen 18](#_Toc198554192)

[Gambar 20 Tampilan Setelah Selesai Menyelesaiakan Program 18](#_Toc198554193)

[Gambar 21 Tampilan Nilai yang Diinput dari Depan 23](#_Toc198554194)

[Gambar 22 Tampilan Nilai yang Diinput dari Belakang 24](#_Toc198554195)

[Gambar 23 Pencarian Nilai 2 pada Linked List 25](#_Toc198554196)

[Gambar 24 Pencarian Nilai 7 pada Linked List 26](#_Toc198554197)

[Gambar 25 Menghapus Data dari Belakang Linked List 27](#_Toc198554198)

[Gambar 26 Tampilan Linked List Setelah Dihapus 27](#_Toc198554199)

[Gambar 27 Menghapus Setiap Nilai 3 yang Ada di Linked List 28](#_Toc198554200)

[Gambar 28 Tampilan Linked List Setelah Setiap Nilai 3 Dihapus 28](#_Toc198554201)

[Gambar 29 Tampilan Isi dari Linked List 29](#_Toc198554202)

# SOAL 1

Cobalah program berikut, running, simpan program, dan screenshoot hasil running !

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  300301  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  356  357  358  359  360  361  362  363  364  365  366  367  368  369  370  371  372  373  374  375  376  377  378  379  380  381  382  383  384  385  386  387  388  389  390  391  392  393  394  395  396  397  398  399  400  401  402  403  404  405  406  407  408  409  410  411  412  413  414  415  416  417  418  419  420  421  422  423  424  425  426  427  428  429  430  431  432  433  434  435  436  437  438  439  440  441  442  443  444  445  446  447  448  449  450  451  452  453  454  455  456  457  458  459  460  461  462  463  464  465  466  467  468  469  470  471  472  473  474  475  476  477  478  479  480  481  482  483  484  485  486  487  488  489  490  491 | #include <conio.h>  #include <iostream>  #include <stdlib.h>  using namespace std;  typedef struct TNode  {      string data;      TNode \*next;  };  TNode \*head, \*tail;  int pil;  char pilihan [2];  string dataBaru, dataDelete;  void Init();  int Kosong();  void TambahDepan();  void TambahBelakang();  void HapusDepan();  void HapusBelakang();  void Tampilkan();  void Reset();  void CariData();  void HapusData();  void SisipkanSebelum();  void SisipkanSetelah();  int main()  {      do      {          cout<<"Single Linked List Circular (SLLC)"<<endl;          cout<<"=================================="<<endl;          cout<<"1. Tambah Depan"<<endl;          cout<<"2. Tambah Belakang"<<endl;          cout<<"3. Hapus Depan"<<endl;          cout<<"4. Hapus Belakang"<<endl;          cout<<"5. Tampilkan Data"<<endl;          cout<<"6. Hapus Semua elemen"<<endl;          cout<<"7. Cari Data"<<endl;          cout<<"8. Hapus Setiap Data Tertentu"<<endl;          cout<<"9. Sisipkan Node/Data Baru Sebelum Data Tertentu"<<endl;          cout<<"10. Sisipkan Node/Data Baru Setelah Data Tertentu"<<endl;          cout<<"11. Keluar"<<endl;          cout<<"=================================="<<endl;          cout<<"Pilihan : ";          cin>>pilihan;          pil = atoi(pilihan);          switch(pil)          {          case 1:              TambahDepan();              break;          case 2:              TambahBelakang();              break;          case 3:              HapusDepan();              cout<<"Data \""<<dataDelete<<"\" yang berada di depan telah berhasil dihapus."<<endl;              break;          case 4:              HapusBelakang();              cout<<"Data \""<<dataDelete<<"\" yang berada di belakang telah berhasil dihapus."<<endl;              break;          case 5:              Tampilkan();              break;          case 6:              Reset();              break;          case 7:              CariData();              break;          case 8:              HapusData();              break;          case 9:              SisipkanSebelum();              break;          case 10:              SisipkanSetelah();              break;          default:              cout<<"\nTERIMA KASIH"<<endl;              cout<<"Program was made by Rizki Adhitiya Maulana (2410817110014)"<<endl;          }          cout<<"\nPress any key to continue!"<<endl;          getch();          system("cls");      }      while (pil<11);  }  void Init()  {      head = NULL;      tail = NULL;  }  int Kosong()  {      if(head == NULL)          return 1;      else          return 0;  }  void TambahDepan()  {      cout<<"Masukkan Data : ";      TNode \*baru;      baru = new TNode;      cin>>dataBaru;      baru->data = dataBaru;      baru->next = baru;      if(Kosong() == 1)      {          head = baru;          tail = baru;      }      else      {          baru->next = head;          head = baru;          tail->next = head;      }      cout<<"Data \""<<dataBaru<<"\" telah berhasil dimasukkan di bagian depan."<<endl;  }  void TambahBelakang()  {      cout<<"Masukkan Data : ";      TNode \*baru;      baru = new TNode;      cin>>dataBaru;      baru->data = dataBaru;      baru->next = baru;      if(Kosong() == 1)      {          head = baru;          tail = baru;      }      else      {          tail->next = baru;          tail = baru;          tail->next = head;      }      cout<<"Data \""<<dataBaru<<"\" telah berhasil dimasukkan di bagian belakang."<<endl;  }  void HapusDepan()  {      if(Kosong() == 0)      {          TNode \*hapus;          hapus = head;          dataDelete = hapus->data;          if(head != tail)          {          head = head->next;          tail->next = head;          }          else          {              Init();          }          delete hapus;      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void HapusBelakang()  {      if(Kosong() == 0)      {          TNode \*hapus, \*newTail;          hapus = tail;          dataDelete = hapus->data;          if(head != tail)          {              newTail = head;              while(newTail->next != tail)              {                  newTail = newTail->next;              }              tail = newTail;              tail->next = head;          }          else          {              Init();          }          delete hapus;      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void Tampilkan()  {      if(Kosong() == 0)      {          TNode \*bantu;          bantu = head;          do          {              cout<<bantu->data<<' ';              bantu = bantu->next;          }          while (bantu != head);          cout<<endl;      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void Reset()  {      if(Kosong() == 0)      {          TNode \*bantu, \*hapus;          bantu = head;          do          {              hapus = bantu;              bantu = bantu->next;              delete hapus;          }          while (bantu != head);          Init();          cout<<"Seluruh elemen pada Linked List telah dibersihkan."<<endl;      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void CariData()  {      if(Kosong() == 0)      {          string cari;          cout<<"Masukkan data yang ingin dicari : ";          cin>>cari;          TNode \*bantu, \*hapus, \*newTail, \*bantuTampilkan;          bool apaDitemukan = false;          bantu = head;          do          {              if(cari == bantu->data)              {                  cout<<"Setiap data yang berada di dalam tanda kurung siku ([...]) "                      <<"merupakan data yang anda cari."<<endl;                  cout<<"Linked List : ";                  bantuTampilkan = head;                  do                  {                      if(cari == bantuTampilkan->data)                      {                          cout<<"["<<bantuTampilkan->data<<"] ";                      }                      else                      {                          cout<<bantuTampilkan->data<<' ';                      }                      bantuTampilkan = bantuTampilkan->next;                  }                  while (bantuTampilkan != head);                  apaDitemukan = true;                  cout<<endl;                  break;              }              bantu = bantu->next;          }          while (bantu != head);          if(apaDitemukan == false)          {              cout<<"Data \""<<cari<<"\" tidak ditemukan pada Linked List."<<endl;          }      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void HapusData()  {      if(Kosong() == 0)      {          string cari;          cout<<"Masukkan data yang ingin dihapus : ";          cin>>cari;          TNode \*bantu, \*sebelum, \*hapus[255], \*bantu2;          int hitung = 0;          bool apaDitemukan = false;          bantu = head;          do          {              bantu2 = bantu;              if(cari == bantu->data)              {                  hapus[hitung++] = bantu;                  apaDitemukan = true;                  if(bantu != head && bantu != tail)                  {                      sebelum->next = bantu->next;                      bantu2 = sebelum;                  }              }              sebelum = bantu2;              bantu = bantu->next;          }          while (bantu != head);            if(apaDitemukan == true)          {              for(int i=0; i<hitung; i++)              {                  if(hapus[i] == head)                  {                      HapusDepan();                  }                  else if(hapus[i] == tail)                  {                      HapusBelakang();                  }                  else                  {                      delete hapus[i];                  }              }              cout<<"Setiap data \""<<cari<<"\" yang terdapat pada Linked List telah dihapus."<<endl;          }          else          {              cout<<"Data \""<<cari<<"\" tidak ditemukan pada Linked List."<<endl;          }      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void SisipkanSebelum()  {      if(Kosong() == 0)      {          TNode \*bantu, \*sebelum;          string nextData;          bool apaAda;          bantu = head;          sebelum = tail;          cout<<"Sisipkan data baru sebelum data : ";          cin>>nextData;          do          {              if(nextData == bantu->data)              {                  apaAda = true;                  break;              }              else              {                  sebelum = bantu;                  bantu = bantu->next;              }          }          while (bantu != head);          if(apaAda == true)          {              cout<<"Masukkan data yang ingin ditambahkan : ";              cin>>dataBaru;              TNode \*baru;              baru = new TNode;              baru->data = dataBaru;              baru->next = bantu;              sebelum->next = baru;              if(bantu == head)              {                  head = baru;              }              cout<<"Data \""<<dataBaru<<"\" berhasil disisipkan sebelum data \""<<nextData<<"\"."<<endl;          }          else          {              cout<<"Tidak terdapat data \""<<nextData<<"\" pada Linked List."<<endl;          }      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;      }  }  void SisipkanSetelah()  {      if(Kosong() == 0)      {          TNode \*bantu;          string prevData;          bool apaAda;          bantu = head;          cout<<"Sisipkan data baru setelah data : ";          cin>>prevData;          do          {              if(prevData == bantu->data)              {                  apaAda = true;                  break;              }              else              {                  bantu = bantu->next;              }          }          while (bantu != head);          if(apaAda == true)          {              cout<<"Masukkan data yang ingin ditambahkan : ";              cin>>dataBaru;              TNode \*baru;              baru = new TNode;              baru->data = dataBaru;              baru->next = bantu->next;              bantu->next = baru;              if(bantu == tail)              {                  tail = baru;              }              cout<<"Data \""<<dataBaru<<"\" berhasil disisipkan setelah data \""<<prevData<<"\"."<<endl;          }          else          {              cout<<"Tidak terdapat data \""<<prevData<<"\" pada Linked List."<<endl;          }      }      else      {          cout<<"Tidak terdapat data pada Linked List."<<endl;}  } |

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 1 Tampilan Awal Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 2 Tampilan Tambah Data dari Depan

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

Konten yang dihasilkan AI mungkin salah.

Gambar 3 Tampilan Data Setelah Dilakukan Tambah Data Depan

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

Konten yang dihasilkan AI mungkin salah.

Gambar 4 Tampilan Tambah Data dari Belakang

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

Konten yang dihasilkan AI mungkin salah.

Gambar 5 Tampilan Data Setelah Dilakukan Tambah Data Belakang

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

Konten yang dihasilkan AI mungkin salah.

Gambar 6 Tampilan Cari Data Yang Tidak Ada

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

Konten yang dihasilkan AI mungkin salah.

Gambar 7 Tampilan Cari Data Yang Ada

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

Konten yang dihasilkan AI mungkin salah.

Gambar 8 Tampilan Hapus Data Belakang

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

Konten yang dihasilkan AI mungkin salah.

Gambar 9 Tampilan Data Setelah Dilakukan Hapus Data Belakang

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

Konten yang dihasilkan AI mungkin salah.

Gambar 10 Tampilan Hapus Setiap Data Tertentu

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

Konten yang dihasilkan AI mungkin salah.

Gambar 11 Tampilan Data Setelah Dilakukan Hapus Setiap Data Tertentu

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

Konten yang dihasilkan AI mungkin salah.

Gambar 12 Tampilan Hapus Data Depan

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

Konten yang dihasilkan AI mungkin salah.

Gambar 13 Tampilan Data Setelah Dilakukan Hapus Data Depan

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

Konten yang dihasilkan AI mungkin salah.

Gambar 14 Tampilan Sisipkan Node Sebelum Data Tertentu

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

Konten yang dihasilkan AI mungkin salah.

Gambar 15 Tampilan Data Setelah Dilakukan Sisipkan Node Sebelum

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

Konten yang dihasilkan AI mungkin salah.

Gambar 16 Tampilan Sisipkan Node Setelah Data Tertentu

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

Konten yang dihasilkan AI mungkin salah.

Gambar 17 Tampilan Data Setelah Dilakukan Sisipkan Node Setelah

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

Konten yang dihasilkan AI mungkin salah.

Gambar 18 Tampilan Hapus Semua Elemen

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

Konten yang dihasilkan AI mungkin salah.

Gambar 19 Tampilan Data Setelah Dilakukan Hapus Semua Elemen

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

Konten yang dihasilkan AI mungkin salah.

Gambar 20 Tampilan Setelah Selesai Menyelesaiakan 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] sampai [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. Kemudian, *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.

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* yang mana digunakan untuk menyimpan variabel Integer atau bilangan bulat*.* Terus *char pilihan [2]* yang mana digunakan untuk menyimpan variabel character, ditambah array sebagai batasan input dari user*, string dataBaru* dan *dataDelete* yang digunakan untuk menyimpan variabel string atau katakter.

Pada baris [19] sampai [31] terdapat penamaan fungsi yang akan dimasukkan ke dalam program Linked list.

Pada baris [33] sampai [99] terdapat int main() yang mana digunakan untuk menjalankan dan menampilkan menu SLLC. Menu yang akan ditampilkan di dalam sistem ada sebanyak 11 buah, terdiri dari tambah depan dan tambah belakang, hapus depan dan hapus belakang, tampilkan data, sapus semua elemen, cari data, hapus data tertentu, sisipkan data sebelum data tertentu dan sisipkan data setelah data terntentu. 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 sebelas (11) untuk keluar atau menghentikan program yang ada.

Pada baris [101] sampai [105] terdapat *void Init()* yang mana digunakan untuk menginisialisasikan kondisi awal dari linked list circular. Pointer atau variabel head akan disetting atau diatur dalam keadaan NULL, begitu juga dengan pointer atau variabel tail akan disetting atau diatur dalam keadaan NULL.

Pada baris [107] sampai [113] terdapat *int Kosong()* yang mana digunakan untuk melakukan pengecekan pada linked list, apakah dalam keadaan kosong atau tidak. Fungsi ini akan mengembalikan nilai 1 apabila linked list dalam keadaan kosong dan 0 apabila tidak dalam keadaan kosong. Fungsi ini akan dipanggil pada saat melakukan operasi penambahan, penghapusan dan menampilkan.

Pada baris [115] sampai [136] terdapat *void TambahDepan()* yang mana digunakan untuk menambahkan node baru ke bagian depan dari linked list. Apabila linked list dalam keadaan kosong, node yang baru ditambahkan akan menjadi head dan tail. Kemudian, apabila linked list dalam keadaan tidak kosong, node yang baru ditambahkan akan menjadi head baru dan node sebelumnya yang jadi head dan tail dalam satu waktu akan menjadi tail.

Pada baris [138] sampai [159] terdapat *void TambahBelakang()* yang mana digunakan untuk menambahkan node baru ke bagian belakang dari linked list. Apabila linked list dalam keadaan kosong, node yang baru ditambahkan akan menjadi tail dan head. Kemudian, apabila linked list dalam keadaan tidak kosong, node yang baru ditambahkan akan menjadi tail baru dan node sebelumnya yang jadi tail dan head dalam satu waktu akan menjadi head.

Pada baris [161] sampai [185] terdapat *void HapusDepan()* yang mana digunakan untuk menghapus node pertama yang terdapat pada linked list. Apabila terdapat satu node saja pada linked list, maka setelah dilakukan hapus depan linked list akan diatur menjadi kosong atau NULL menggunakan fungsi *Init().* Kemudian, apabila terdapat lebih dari satu node yang terdapat pada linked list, node pertama yang dihapus akan digantikan dengan elemen yang ada di setelah sebagai head terbaru dan node pertama baru.

Pada baris [187] sampai [215] terdapat *void HapusBelakang()* yang mana digunakan untuk menghapus node terakhir yang terdapat pada linked list. Apabila terdapat satu node saja pada linked list, maka setelah dilakukan hapus belakang linked list akan diatur menjadi kosong atau NULL menggunakan fungsi *Init().* Kemudian, apabila terdapat lebih dari satu node yang terdapat pada linked list, node terakhir yang dihapus akan digantikan dengan elemen yang ada di sebelumnya sebagai tail terbaru dan node terakhir baru.

Pada baris [217] sampai [236] terdapat *void Tampilkan()* yang mana digunakan untuk menampilkan seluruh isi linked list yang ada, dimulai dari node pertama hingga ke node terakhir. Walapun hanya terdapat satu node di dalam linked list, program akan tetap mencetaknya dan akan berhenti ketika semua elemen yang ada di dalam linked list ditampilkan semua.

Pada baris [238] sampai [259] terdapat *void Reset()* 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 *Init()* untuk mengatur linked list ke dalam kondisi kosong.

Pada baris [261] sampai [314] terdapat *void CariData()* yang mana digunakan untuk mencari data tertentu yang ada di dalam linked list. Apabila ditemukan kesamaan data yang ada di dalam tanda kurung siku dengan yang ada di dalam linked list, maka data yang ada di dalam linked list akan ikut di cetak dalam tanda kurung siku. Namun, apabila tidak ditemukan kesamaan antara data yang ada di dalam tanda kurung siku dengan yang ada di dalam linked list, maka akan muncul tampilan pesan kepada pengguna kalo data A tidak ditemukan.

Pada baris [316] sampai [376] terdapat *void HapusData()* yang mana digunakan untuk menghapus semua node yang ada di dalam linked list, berdasarkan data yang di inputkan oleh pengguna. Fungsi ini akan menghapus node tertentu di dalam linked list ketika pengguna memasukkan sebuah data atau nilai yang ingin di hapus. Ketika selesai memasukkan data yang diinginkan, sistem akan menghapus setiap node yang memiliki kesamaan dengan nilai yang dimasukkan hingga tidak terdapatnya data tersebut di dalam linked list.

Pada baris [378] sampai [435] terdapat *void SisipkanSebelum()* yang mana digunakan untuk menyisipkan atau memasukkan node baru sebelum node tertentu, sesuai dengan data yang dimasukkan oleh pengguna sebagai data yang akan disisipkan dan data target untuk penyisipan. Sebelum dilakukan penyisipan, terlebih dahulu dilakukan penelusuran pada linked list dari head menuju tail untuk menemukan sebuah node yang menjadi target untuk disisipkan. Setelah target untuk penyisipan ditemukan, maka akan ditambahkan atau disisipkan sebuah node di depannya atau sebelum node tersebut. Apabila node yang menjadi target sisipkan sebelum adalah head dalam linked list maka node yang baru di tambahkan akan menjadi head yang baru.

Pada baris [437] sampai [491] terdapat *void SisipkanSesudah()* yang mana digunakan untuk menyisipkan atau memasukkan node baru setelah node tertentu sesuai dengan data yang dimasukkan oleh pengguna sebagai data yang akan disisipkan dan data target untuk penyisipan. Sebelum dilakukan penyisipan, terlebih dahulu dilakukan penelusuran pada linked list dari head menuju tail untuk menemukan sebuah node yang menjadi target untuk disisipkan. Setelah target untuk penyisipan ditemukan, maka akan ditambahkan atau disisipkan sebuah node di belakang atau setelah node tersebut. Apabila node yang menjadi target sisipkan setelah adalah tail dalam linked list maka node yang baru di tambahkan akan menjadi tail yang baru.

# SOAL 2

Lakukan tambah data depan 3, 4, 7, 9, 10, 12 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 21 Tampilan Nilai yang Diinput dari Depan

## Pembahasan

Ketika data dimasukkan ke dalam Linked List melalui bagian depan, dengan urutan masuk mulai dari 3, 4, 7, 9, 10, dan 12. Angka 3 yang pertama kali dimasukkan akan muncul di sebelah kiri ketika di tampilkan. Data-data yang muncul setelah data pertama ditambahkan akan menjadi head dari Linked List dan data sebelumnya akan digeser ke samping kiri atau ke belakang.

# 

# SOAL 3

Lakukan tambah data belakang 3, 7, 1, 4, 3 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 22 Tampilan Nilai yang Diinput dari Belakang

## Pembahasan

Ketika data dimasukkan ke dalam Linked List melalui bagian belakang, dengan urutan masuk mulai dari 3, 7, 1, 4, dan 3. Angka 3 yang pertama kali dimasukkan akan muncul di sebelah kiri ketika di tampilkan. Data-data yang muncul setelah data pertama ditambahkan akan menjadi tail dari Linked List dan tampilkan dari Linked List akan sama seperti saat kita memasukkan data ke dalam Linked List.

# SOAL 4

Apa yang terjadi jika mencari angka 2 pada Single Linked List Circular (SLLC) pada data yang telah ditambahkan/dimasukkan sebelumnya dan screenshoot hasilnya

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 23 Pencarian Nilai 2 pada Linked List

## Pembahasan

Berdasarkan data yang telah dimasukkan pada soal 2 dan soal 3, ketika dilakukan sebuah pencarian untuk nilai 2. Sistem akan memberikan informasi bahwa nilai tersebut tidak terdapat atau tidak ditemukan pada Linked List.

# SOAL 5

Coba cari angka 7 dan screenshoot hasilnya !

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 24 Pencarian Nilai 7 pada Linked List

## Pembahasan

Berdasarkan data yang telah dimasukkan pada soal 2 dan soal 3, ketika dilakukan sebuah pencarian untuk nilai 7. Sistem akan memberikan informasi bahwa nilai yang dicari akan ditandai dengan kurung siku […] pada Linked List.

# SOAL 6

Lakukan hapus belakang dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 25 Menghapus Data dari Belakang Linked List

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

Konten yang dihasilkan AI mungkin salah.

Gambar 26 Tampilan Linked List Setelah Dihapus

## Pembahasan

Berdasarkan data yang telah dimasukkan pada soal 2 dan soal 3, ketika dilakukan sebuah penghapusan data untuk bagian belakang atau tail. Nilai 3 yang terletak di bagian paling belakang pada Linked List (tail) akan dihapuskan dan ketika Linked List di tampilkan nilai yang ada disebelah kiri dari nilai yang dihapus akan menjadi tail yang baru.

# SOAL 7

Lakukan hapus setiap angka 3 dan kemudian lakukan tampilkan data lalu screenshoot hasilnya !

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 27 Menghapus Setiap Nilai 3 yang Ada di Linked List

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

Konten yang dihasilkan AI mungkin salah.

Gambar 28 Tampilan Linked List Setelah Setiap Nilai 3 Dihapus

## Pembahasan

Berdasarkan data yang telah dimasukkan pada soal 2 dan soal 3, ketika dilakukan sebuah penghapusan untuk setiap Nilai 3 yang ada pada Linked List. Nilai 3 yang terdapat pada Linked List akan dihapuskan semua dan ketika Linked List di tampilkan tidak akan terdapat nilai 3 di dalamnya.

# SOAL 8

Tampilkan data lalu jelaskan yang mana head dan yang mana tail.

## Output Program

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

Konten yang dihasilkan AI mungkin salah.

Gambar 29 Tampilan Isi dari Linked List

## Pembahasan

Berdasarkan data yang telah dimasukkan pada soal 2 dan soal 3, kemudian dilakukannya penghapusan pada soal 6 dan soal 7. Data yang berada di paling depan atau di samping kiri akan menjadi head yaitu 12, dapat juga dibuktikan dengan melakukan sebuah operasi penambahan atau penghapusan data dari depan yang mana akan mempengaruhi data atau nilai dari 12 itu sendiri. Apabila ditambah data 1 pada Linked List dari depan data yang baru ditambahkan tersebut yang akan menjadi head. Begitu juga dengan penghapusan data atau nilai dari 12, maka data atau nilai yang ada di belakangnya yang akan menjadi head baru. Sedangkan data yang berada di paling belakang atau di samping kanan akan menjadi tail yaitu 4, dapat juga dibuktikan dengan melakukan sebuah operasi penambahan atau penghapusan data dari belakang yang mana akan mempengaruhi data atau nilai dari 4 itu sendiri. Apabila ditambah data 1 pada Linked List dari belakang data yang baru ditambahkan tersebut yang akan menjadi tail. Begitu juga dengan penghapusan data atau nilai dari 4, maka data atau nilai yang ada di depannya yang akan menjadi tail baru.

# SOAL 9

Jika baris ke 103 dan 104 dihapus maka apa yang akan terjadi pada saat memasukkan data, dan jelaskan mengapa?

## Pembahasan

Apabila baris ke-103 dan 104 dihapuskan dari program yang mana pada baris itu terdapat *if(head == NULL) return 1;* dan *else return 0;* di dalamnya. Maka akan fungsi pengecekan untuk mengetahui apakah Linked List kosong atau berisi tidak bisa dilakukan di dalam program. Tanpa kedua baris kode tersebut, program masih bisa berjalan, namun berisiko mengalami kesalahan atau perilaku yang tidak terduga, seperti penggunaan garbage value atau pengaksesan memori yang tidak valid, yang dapat menyebabkan crash atau error lainnya.

# SOAL 10

Jelaskan apa itu variabel head dan tail pada SLLC!

## Pembahasan

Pada Single Linked List Circular (SLLC), variabel head merupakan pointer yang menunjuk ke node pertama dalam Linked List, atau dengan kata lain, variabel head berfungsi sebagai tempat pertama data akan ditelusuri. Sedangkan variabel tail merupakan pointer yang menunjuk ke node terakhir dalam Linked List, atau tempat terakhir data akan ditelusuri.

Karena SLLC bersifat melingkar (*circular*), maka node terakhir (*tail*) akan menunjuk kembali ke node pertama (*head*), yang mana akan menjadikannya sebuah struktur yang terus berputar. Singkatnya, tail sebagai node terakhir akan menunjuk kembali ke head, sehingga memungkinkan traversal list secara terus-menerus tanpa ada akhir.

# TAUTAN GIT HUB

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