Laporan Praktikum 9

Struktur Data Algoritma

Materi
“Double Linked List & Stack”

Nama :
Muhammad Azhar Rasyad

NIM :
0110217029

Program Studi :
Teknik Informatika 1

Double Linked List

Berikut contoh program double linked list dengan C++ :

```
#include <iostream>

using namespace std;

struct node
{
  int data;
  node *next;
  node *prev;
} *head, *tail, *baru, *bantu, *help;

int nilai;
int counter = 0;
int posisi;
int i;

// Buat Node
void buatNode(int nilai);
// Tambah Node
void tambahDepan();
void tambahTengah();
void tambahBelakang();
// Ubah Node
void ubahDepan();
void ubahTengah();
void ubahBelakang();
// Hapus Node
void hapusDepan();
void hapusTengah();
void hapusBelakang();
// Lihat Node
void lihatDepan();
void lihatTengah();
void lihatBelakang();
void lihatSemua();
// Lainnya
void tampilan();
void pause();

int main()
{
  int menu;

  menu:
  cout << "<----------Double Linked List---------->" << endl;
```

```
  cout << "\n<----------Menu Tambah---------->" << endl;
  cout << "1. Tambah Node di Depan" << endl;
  cout << "2. Tambah Node di Tengah" << endl;
  cout << "3. Tambah Node di Belakang" << endl;
  cout << "\n<----------Menu Ubah---------->" << endl;
  cout << "4. Ubah Node di Depan" << endl;
  cout << "5. Ubah Node di Tengah" << endl;
  cout << "6. Ubah Node di Belakang" << endl;
  cout << "\n<----------Menu Hapus---------->" << endl;
  cout << "7. Hapus Node di Depan" << endl;
  cout << "8. Hapus Node di Tengah" << endl;
  cout << "9. Hapus Node di Belakang" << endl;
  cout << "\n<----------Menu Lihat---------->" << endl;
  cout << "10. Lihat Node di Depan" << endl;
  cout << "11. Lihat Node di Tengah" << endl;
  cout << "12. Lihat Node di Belakang" << endl;
  cout << "13. Lihat Node Semua" << endl;
  tampilan();
  cout << "\n14. Keluar" << endl << endl;
  cout << "Masukkan Pilihan : ";
  cin >> menu;

  switch (menu)
  {
    case 1:
      cout << "\nMasukkan Nilai Node di Depan = ";
      cin >> nilai;
      buatNode(nilai);
      tambahDepan();
      tampilan();
      pause();
      goto menu;
      break;

    case 2:
      cout << "\nMasukkan Nilai Node di Tengah = ";
      cin >> nilai;
      buatNode(nilai);
      tambahTengah();
      tampilan();
      pause();
      goto menu;
      break;

    case 3:
      cout << "\nMasukkan Nilai Node di Belakang = ";
      cin >> nilai;
      buatNode(nilai);
      tambahBelakang();
      tampilan();
```

```
      pause();
      goto menu;
      break;

    case 4:
      ubahDepan();
      tampilan();
      pause();
      goto menu;
      break;

    case 5:
      ubahTengah();
      tampilan();
      pause();
      goto menu;
      break;

    case 6:
      ubahBelakang();
      tampilan();
      pause();
      goto menu;
      break;

    case 7:
      hapusDepan();
      tampilan();
      pause();
      goto menu;
      break;

    case 8:
      hapusTengah();
      tampilan();
      pause();
      goto menu;
      break;

    case 9:
      hapusBelakang();
      tampilan();
      pause();
      goto menu;
      break;

    case 10:
      lihatDepan();
      tampilan();
      pause();
```

```
      goto menu;
      break;

    case 11:
      lihatTengah();
      tampilan();
      pause();
      goto menu;
      break;

    case 12:
      lihatBelakang();
      tampilan();
      pause();
      goto menu;
      break;

    case 13:
      lihatSemua();
      tampilan();
      pause();
      goto menu;
      break;

    case 14:
      cout << "\n<----------Exit Program---------->" << endl;
      break;

    default:
       cout << "\n<----------Pilihan Tidak Ada---------->" << endl
<< endl;
      pause();
      goto menu;
      break;
  }
}

void buatNode(int nilai)
{
  baru = new node; // Membuat Node Baru
  baru -> data = nilai; // Mengisi Data Node Baru
  baru -> next = NULL; // Membuat Alamat Next Node Baru
  baru -> prev = NULL; // Membuat Alamat Prev Node Baru
}

void tambahDepan()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    head = baru;
```

```
    tail = baru;
  }
  else // Jika Ada 1 Node atau Lebih
  {
    baru -> next = head;
    baru -> prev = NULL;
    head = baru;
  }
}

void tambahTengah()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    head = baru;
    tail = baru;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
    cout << "\nNode Hanya Ada 1 Gagal Ditambahkan" << endl;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
    counter = 0;
    while(bantu != NULL)
    {
      counter++;
      bantu = bantu -> next;
    }

    if(counter == 2)
    {
      bantu = head;
      help = bantu -> next;
      baru -> next = help;
      bantu -> next = baru;
      baru -> prev = head;
    }
    else
    {
      tampilan();

       cout << "\nNode Yang Bisa Ditambahkan Ditengah Hanya Posisi
2 Sampai " << counter << endl;

      cout << "\nMasukkan Posisi Node Yang Ingin Ditambah = ";
      cin >> posisi;

      if(1 < posisi && posisi <= counter)
```

```
      {
          bantu = head;

          for(i = 2; i < posisi; i++)
          {
            bantu = bantu -> next;
          }

          help = bantu -> next;
          baru -> next = help;
          bantu -> next = baru;
          baru -> prev = bantu;
      }
      else
      {
        cout << "\nPosisi Node Tidak Ada" << endl;
      }
    }
  }
}

void tambahBelakang()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    head = baru;
    tail = baru;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
    head -> next = baru;
    tail = baru;
    tail -> prev = head;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = tail;
    tail -> next = baru;
    tail = baru;
    tail -> next = NULL;
    tail -> prev = bantu;
  }
}

void ubahDepan()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Node Yang Diubah" << endl;
  }
```

```
  else  // Jika Ada 1 Node atau Lebih
  {
    tampilan();
     cout << "\nNode : Nilai Node " << head -> data << " Ada Di
Depan" << endl;
    cout << "\nMasukkan Nilai Node Baru = ";
    cin >> nilai;
     cout << "\nNode : Nilai Node " << head -> data << " Diubah
Menjadi Nilai Node " << nilai << endl;
    head -> data = nilai;
  }
}

void ubahTengah()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Node Yang Diubah" << endl;
  }
  else if(head -> next == NULL || head -> next -> next == NULL) //
Jika Ada 1 Node
  {
     cout << "\nNode Gagal Dilihat Tidak Ada Yang Di Tengah" <<
endl;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
    counter = 0;
    while(bantu != NULL)
    {
      counter++;
      bantu = bantu -> next;
    }

    if(counter == 3)
    {
      tampilan();
       cout << "\nNode : Nilai Node " << head -> next -> data << "
Ada Di Tengah" << endl;

      cout << "\nMasukkan Nilai Node Baru = ";
      cin >> nilai;

       cout << "\nNode : Nilai Node " << head -> next -> data << "
Diubah Menjadi Nilai Node " << nilai << endl;
      head -> next -> data = nilai;
    }
    else
    {
```

```
      tampilan();

        cout << "\nNode Yang Bisa Diubah Ditengah Hanya Posisi 2
Sampai " << counter-1 << endl;

      cout << "\nMasukkan Posisi Node Yang Ingin Diubah = ";
      cin >> posisi;

      if(1 < posisi && posisi < counter)
      {
          bantu = head;

          for(i = 1; i < posisi; i++)
          {
            bantu = bantu -> next;
          }

           cout << "\nNode : Nilai Node " << bantu -> data << " Ada
Di Posisi " << posisi << endl;

          cout << "\nMasukkan Nilai Node Baru = ";
          cin >> nilai;

          bantu -> data = nilai;
      }
      else
      {
        cout << "\nPosisi Node Tidak Ada" << endl;
      }
    }
  }
}

void ubahBelakang()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Node Yang Diubah" << endl;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
    tampilan();
     cout << "\nNode : Nilai Node " << head -> data << " Ada Di
Belakang" << endl;
    cout << "\nMasukkan Nilai Node Baru = ";
    cin >> nilai;
     cout << "\nNode : Nilai Node " << head -> data << " Diubah
Menjadi Nilai Node " << nilai << endl;
    head -> data = nilai;
  }
```

```
  else // Jika Ada Lebih Dari 1 Node
  {
    tampilan();
     cout << "\nNode : Nilai Node " << tail -> data << " Ada Di
Belakang" << endl;
    cout << "\nMasukkan Nilai Node Baru = ";
    cin >> nilai;
     cout << "\nNode : Nilai Node " << tail -> data << " Diubah
Menjadi Nilai Node " << nilai << endl;
    tail -> data = nilai;
  }
}

void hapusDepan()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Node Yang Dihapus" << endl;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node atau Lebih
  {
    bantu = head;
     cout << "\nNode : Node Depan " << head -> data << " Terhapus"
<< endl;
    head = NULL;
    tail = NULL;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
     cout << "\nNode : Node Depan " << head -> data << " Terhapus"
<< endl;
    head = head -> next;
    bantu -> next = NULL;
    bantu -> prev = NULL;
  }
}

void hapusTengah()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    head = baru;
    tail = baru;
  }
  else if(head -> next == NULL || head -> next -> next == NULL) //
Jika Ada 1 Node
  {
     cout << "\nNode Gagal Dihapus Tidak Ada Yang Di Tengah" <<
endl;
```

```
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
    counter = 0;
    while(bantu != NULL)
    {
      counter++;
      bantu = bantu -> next;
    }

    if(counter == 3)
    {
      cout << "\nNode : Node Tengah " << head -> next -> data << "
Terhapus" << endl;
      bantu = head -> next;
      head -> next = bantu -> next;
      tail -> prev = head -> next;
      bantu -> next = NULL;
      bantu -> prev = NULL;
    }
    else
    {
      tampilan();

       cout << "\nNode Yang Bisa Dihapus Ditengah Hanya Posisi 2
Sampai " << counter-1 << endl;

      cout << "\nMasukkan Posisi Node Yang Ingin Dihapus = ";
      cin >> posisi;

      if(1 < posisi && posisi < counter)
      {
          bantu = head;

          for(i = 2; i < posisi; i++)
          {
            bantu = bantu -> next;
          }

          cout << "\nNode : Node Tengah " << bantu -> next -> data
<< " Terhapus" << endl;

          help = bantu -> next;
          bantu -> next = help -> next;
          help -> next = NULL;
          help -> prev = NULL;
      }
      else
      {
```

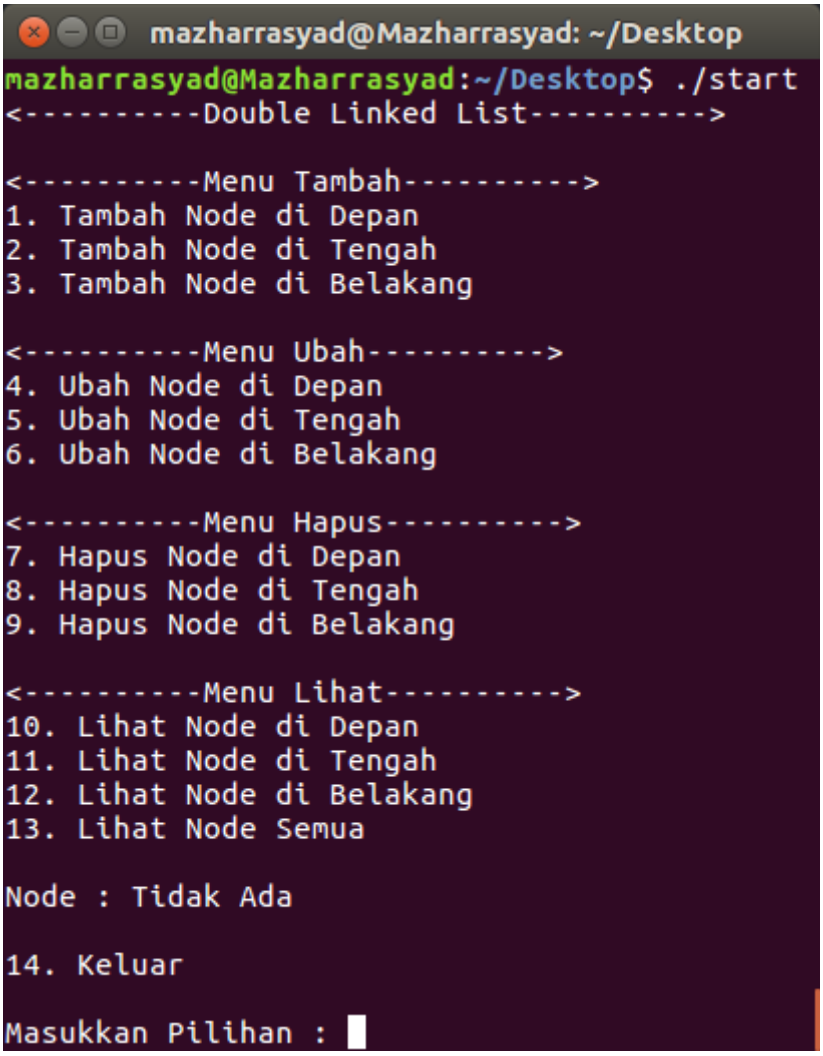
```
        cout << "\nPosisi Node Tidak Ada" << endl;
      }
    }
  }
}

void hapusBelakang()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Node Yang Dihapus" << endl;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
    bantu = head;
      cout << "\nNode : Node Belakang " << head -> data << "
Terhapus" << endl;
    head = NULL;
    tail = NULL;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = tail;
    help = head;
    while(help -> next != tail)
    {
      help = help -> next;
    }
      cout << "\nNode : Node Belakang " << tail -> data << "
Terhapus" << endl;
    tail = help;
    help -> next = NULL;
    bantu -> next = NULL;
    bantu -> prev = NULL;
  }
}

void lihatDepan()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Yang Dilihat" << endl;
  }
  else // Jika Ada 1 Node atau Lebih
  {
     cout << "\nNode : " << "Node " << head -> data << " Terlihat
di Depan" << endl;
  }
}
```

```
void lihatTengah()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Node Yang Dilihat" << endl;
  }
  else if(head -> next == NULL || head -> next -> next == NULL) //
Jika Ada 1 Node
  {
     cout << "\nNode Gagal Dilihat Tidak Ada Yang Di Tengah" <<
endl;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
    counter = 0;
    while(bantu != NULL)
    {
      counter++;
      bantu = bantu -> next;
    }

    if(counter == 3)
    {
      tampilan();
       cout << "\nNode : Nilai Node " << head -> next -> data << "
Ada Di Tengah" << endl;
    }
    else
    {
      tampilan();

       cout << "\nNode Yang Bisa Dilihat Ditengah Hanya Posisi 2
Sampai " << counter-1 << endl;

      cout << "\nMasukkan Posisi Node Yang Ingin Dilihat = ";
      cin >> posisi;

      if(1 < posisi && posisi < counter)
      {
          bantu = head;

          for(i = 1; i < posisi; i++)
          {
            bantu = bantu -> next;
          }
          cout << "\nNode : Nilai Node " << bantu -> data << " Ada
Di Tengah Di Posisi " << posisi << endl;
      }
      else
```

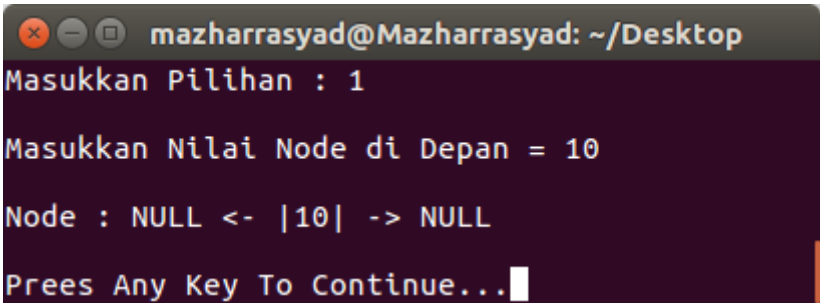
```
      {
        cout << "\nPosisi Node Tidak Ada" << endl;
      }
    }
  }
}

void lihatBelakang()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Yang Dilihat" << endl;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
    cout << "\nNode : " << "Node " << head -> data << " Terlihat
di Belakang" << endl;
  }
  else  // Jika Ada Lebih Dari 1 Node
  {
    cout << "\nNode : " << "Node " << tail -> data << " Terlihat
di Belakang" << endl;
  }
}

void lihatSemua()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada Yang Dilihat" << endl;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
    cout << "\nNode 1 = " << head -> data << endl;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
    cout << "\nSemua Node : ";
    counter = 0;
    while(bantu != NULL)
    {
      counter++;
      cout << "\nNode " << counter << " = " << bantu -> data;
      bantu = bantu -> next;
    }
    cout << endl;
  }
}
```

```
void tampilan()
{
  if(head == NULL) // Jika Tidak Ada Node
  {
    cout << "\nNode : Tidak Ada" << endl;
  }
  else if(head -> next == NULL) // Jika Ada 1 Node
  {
     cout << "\nNode : NULL <- " << "|" << head -> data << "| ->
NULL" << endl;
  }
  else // Jika Ada Lebih Dari 1 Node
  {
    bantu = head;
    cout << "\nNode : NULL <- ";
    while(bantu != NULL)
    {
      cout << "|" << bantu -> data;
      if(bantu -> next == NULL)
      {
        cout << "| -> NULL";
      }
      else
      {
        cout << "| <-> ";
      }
      bantu = bantu -> next;
    }
    cout << endl;
  }
}

void pause()
{
  cout << "\nPrees Any Key To Continue...";
  cin.ignore();
  cin.get();
  cout << endl;
}
```

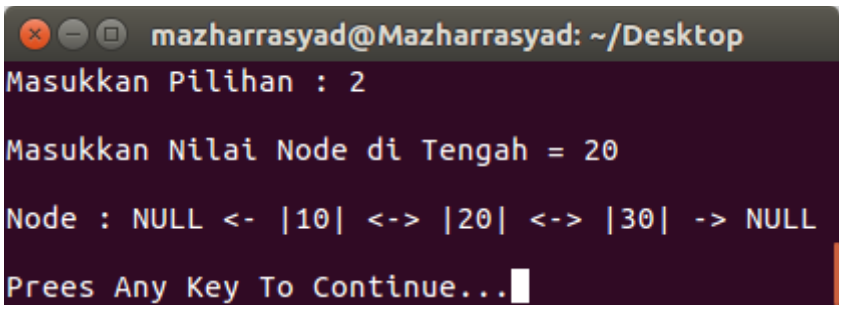
Berikut penjelasan dari program double linked list diatas :



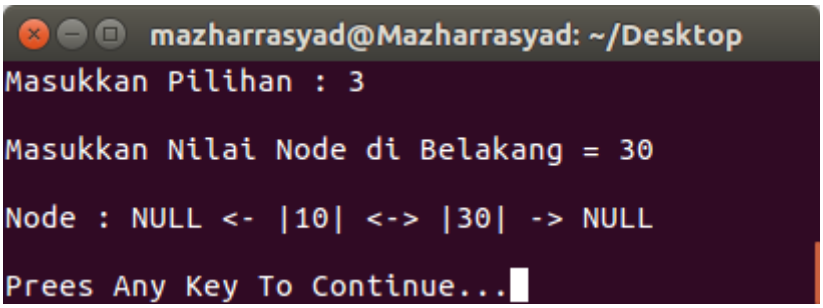
Tampilan diatas merupakan menu utama double linked list.



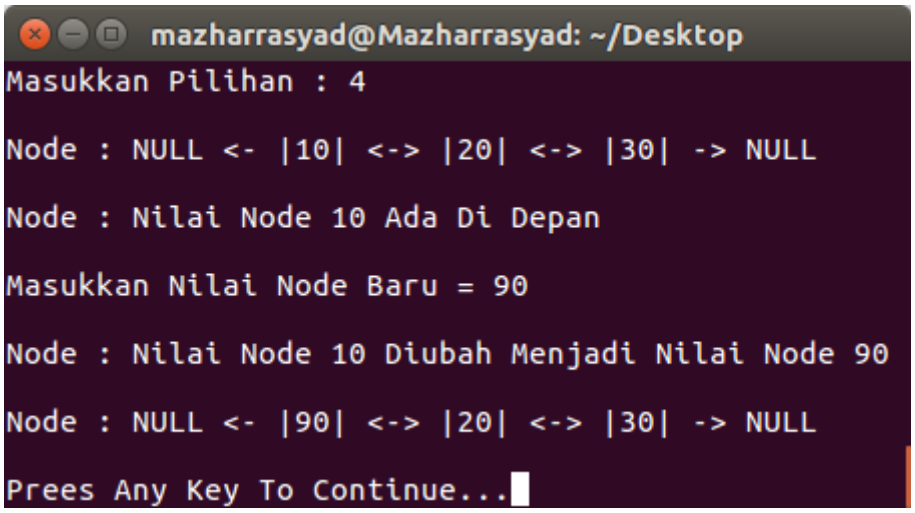
Tampilan diatas merupakan fungsi tambah node di depan.



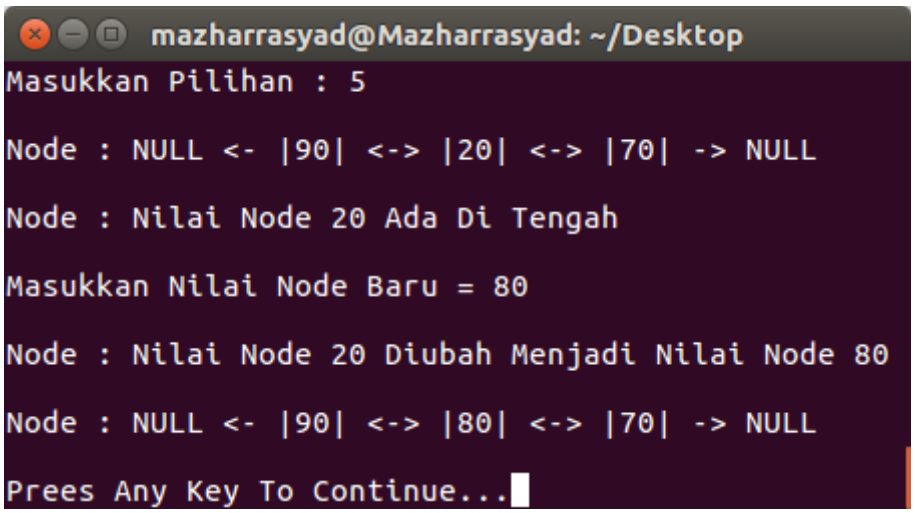
Tampilan diatas merupakan fungsi tambah node di tengah.



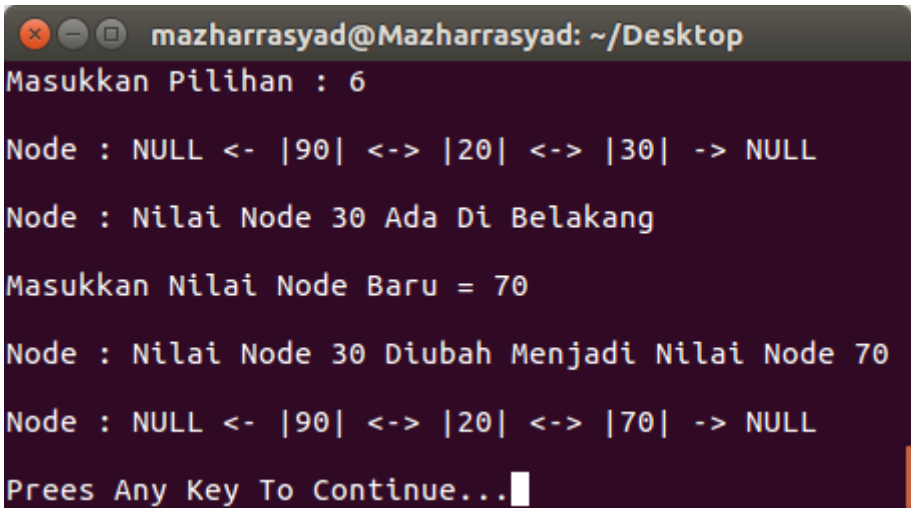
Tampilan diatas merupakan fungsi tambah node di belakang.



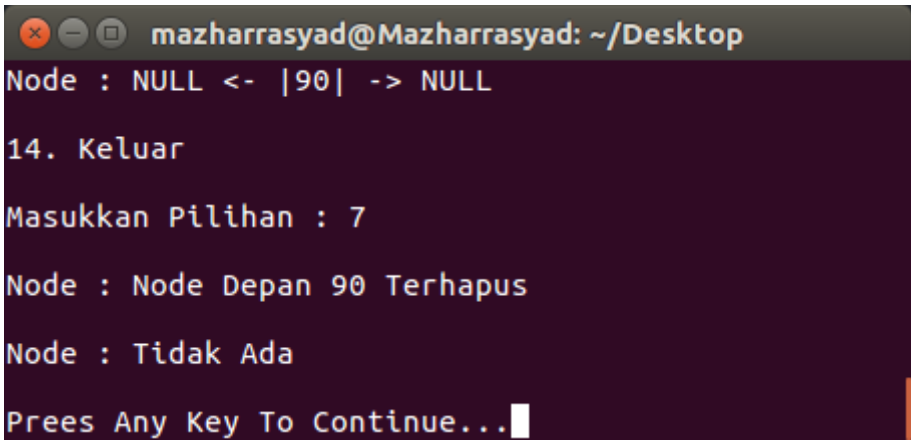
Tampilan diatas merupakan fungsi ubah node di depan.



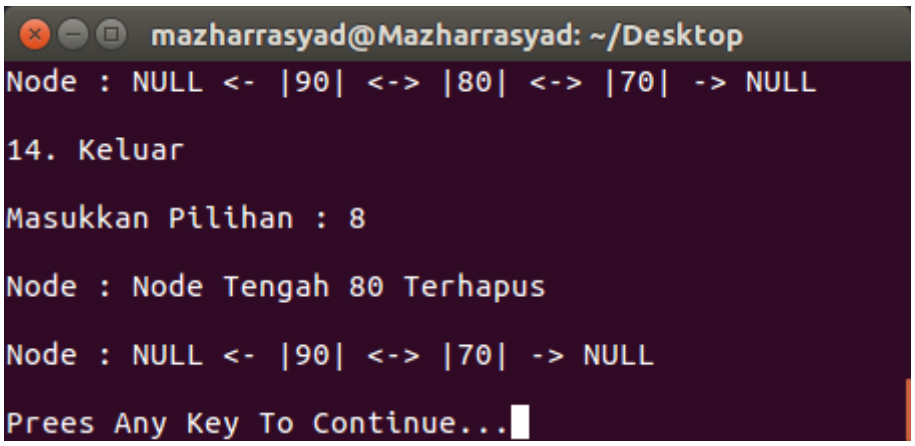
Tampilan diatas merupakan fungsi ubah node di tengah.



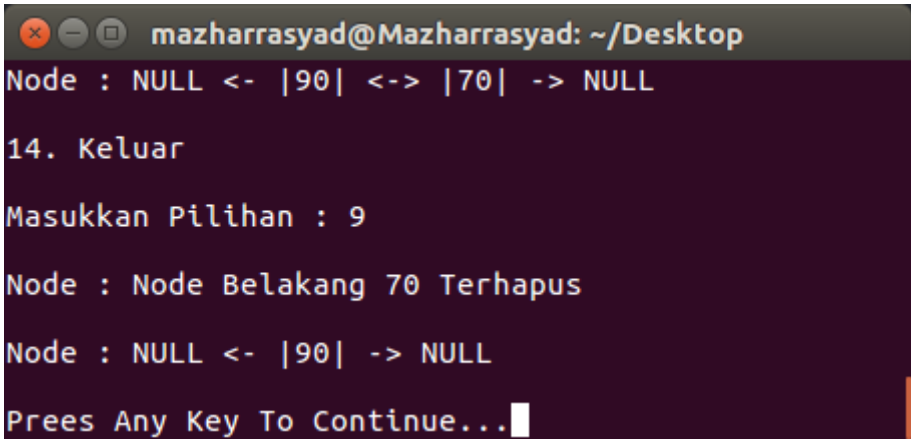
Tampilan diatas merupakan fungsi ubah node di belakang.



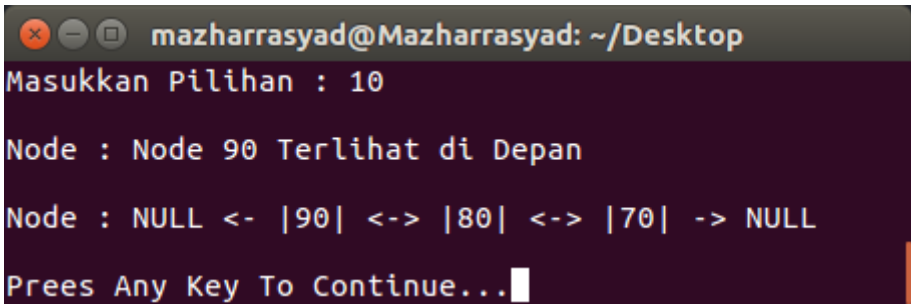
Tampilan diatas merupakan fungsi hapus node di depan.



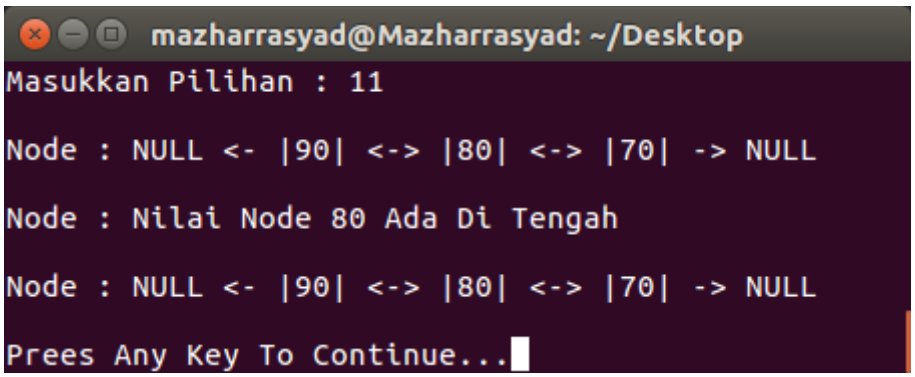
Tampilan diatas merupakan fungsi hapus node di tengah.



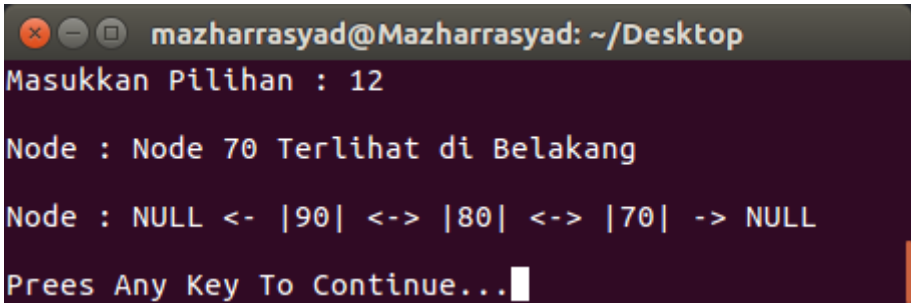
Tampilan diatas merupakan fungsi hapus node di belakang.



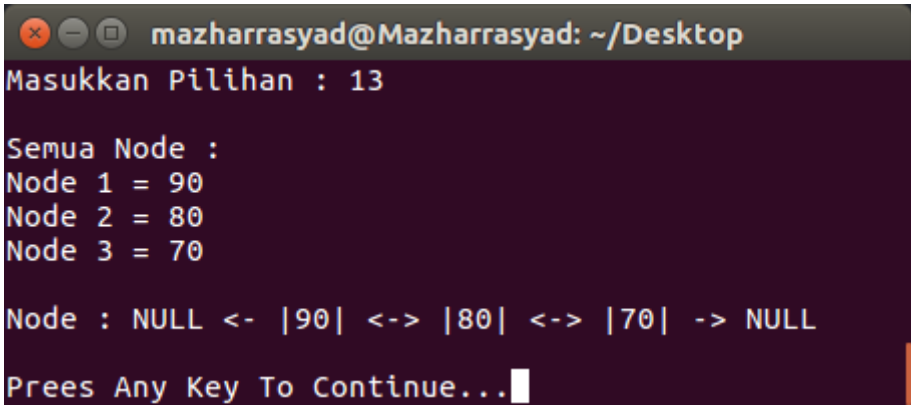
Tampilan diatas merupakan fungsi lihat node di depan.



Tampilan diatas merupakan fungsi lihat node di tengah.



Tampilan diatas merupakan fungsi lihat node di belakang.



Tampilan diatas merupakan fungsi lihat node semua.

Stack

Berikut contoh program stack dengan C++ :

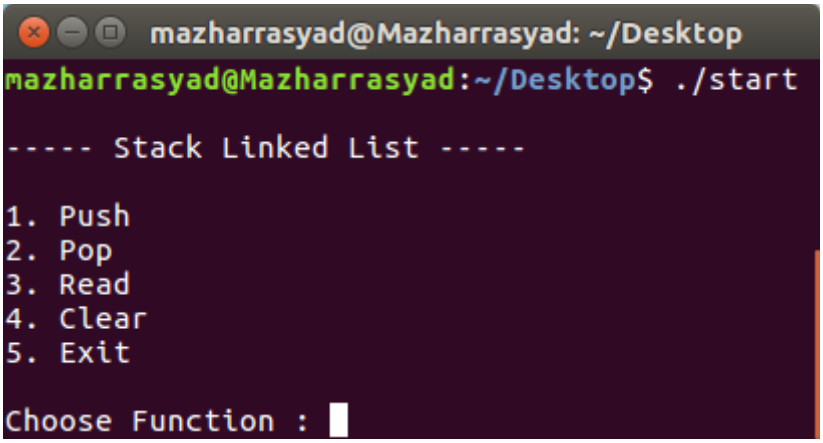
```
#include <iostream>

#define max 5
int value;
int counter = 0;

using namespace std;

struct node
{
  int data;
  node *next;
} *top, *baru, *bantu;

void create(int value);
void push();
void pop();
void read();
void clear();

int main()
{
  int menu;

  menu:
  cout << "\n----- Stack Linked List -----" << endl;
  cout << "\n1. Push";
  cout << "\n2. Pop";
  cout << "\n3. Read";
  cout << "\n4. Clear";
  cout << "\n5. Exit" << endl;
  cout << "\nChoose Function : ";
  cin >> menu;

  switch(menu)
  {
    case 1:
      cout << "\nInput Data = ";
      cin >> value;
      create(value);
      push();
      goto menu;
      break;

    case 2:
```

```
      pop();
      goto menu;
      break;

    case 3:
      read();
      goto menu;
      break;

    case 4:
      clear();
      goto menu;
      break;

    case 5:
           cout << "\n---------- Thanks For Using The Program
----------" << endl;
      break;

    default:
      cout << "\n---------- Error No Function ----------" << endl;
      cout << "\nPress Any Key To Continue...";
      cin.ignore();
      cin.get();
      goto menu;
      break;
  }
}

void create(int value)
{
  baru = new node;
  baru -> data = value;
  baru -> next = NULL;
}

void push()
{
  if(top == NULL)
  {
    top = baru;
    counter++;

    cout << "\n<---Input Stack Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(counter == max)
```

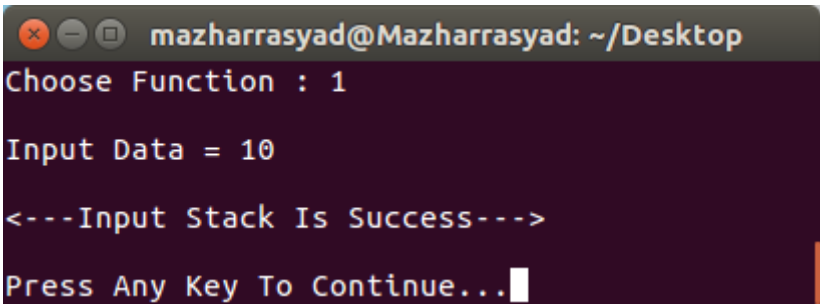
```
  {
    cout << "\n<---Stack Is Full--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    baru -> next = top;
    top = baru;
    counter++;

    cout << "\n<---Input Stack Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void pop()
{
  if(top == NULL)
  {
    cout << "\n<---Stack Is Empty--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    top = top -> next;
    counter--;

    cout << "\n<---Delete Stack Is Success--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void read()
{
  if(top == NULL)
  {
    cout << "\n<---Stack Is Empty--->" << endl;
```

```
    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else if(top -> next == NULL)
  {
    cout << "\nData Stack : " << endl;
    cout << "    |" << top -> data << "|" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    bantu = top;
    cout << "\nData Stack : " << endl;
    while(bantu != NULL)
    {
      cout << "    |" << bantu -> data << "|" << endl;
      bantu = bantu -> next;
    }
    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}

void clear()
{
  if(top == NULL)
  {
    cout << "\n<---Stack Is Empty--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
  else
  {
    top = NULL;
    counter = 0;

    cout << "\n<---Stack Is Clear--->" << endl;

    cout << "\nPress Any Key To Continue...";
    cin.ignore();
    cin.get();
  }
}
```

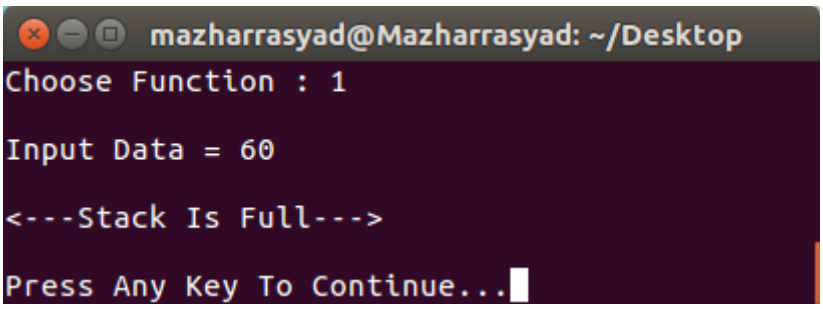
Berikut penjelasan dari program stack diatas :



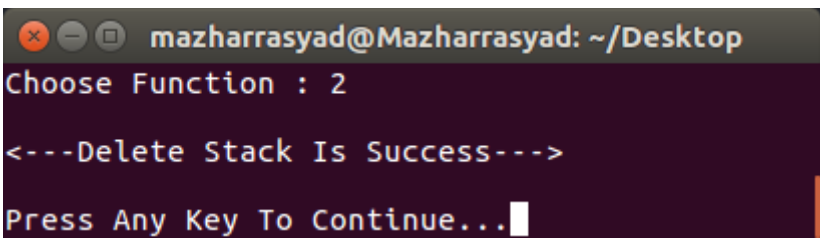
Tampilan diatas merupakan menu utama stack.



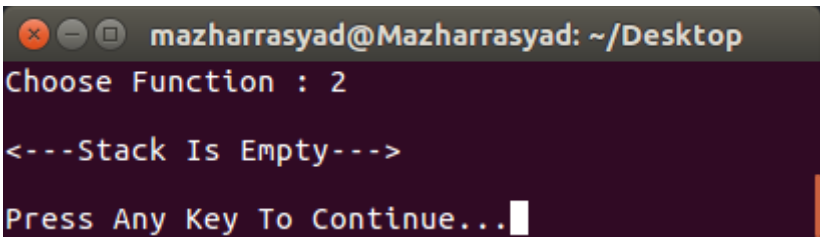
Tampilan diatas merupakan fungsi push data.



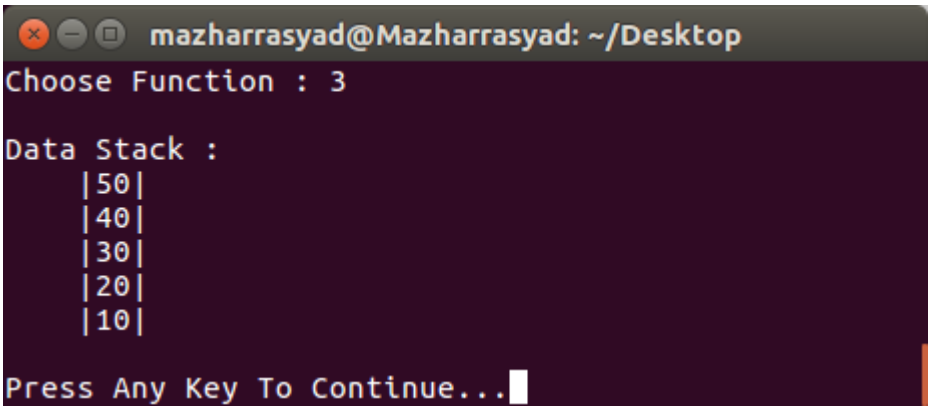
Tampilan diatas merupakan fungsi push data jika stack penuh.



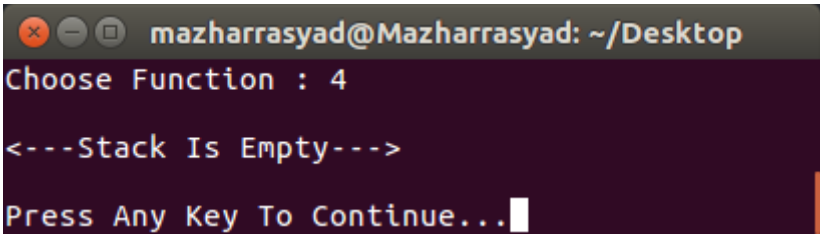
Tampilan diatas merupakan fungsi pop data.



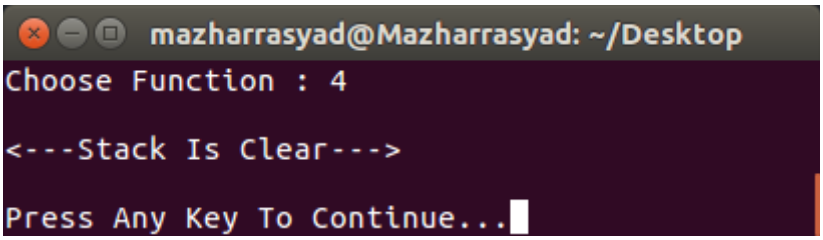
Tampilan diatas merupakan fungsi pop data jika stack kosong.



Tampilan diatas merupakan fungsi read data stack.



Tampilan diatas merupakan fungsi read data jika stack kosong.



Tampilan diatas merupakan fungsi clear data stack.

Kesimpulan

Double linked list merupakan kumpulan data yang saling terhubung satu sama lain dengan menggunakan 2 alamat pointer sehingga setiap node memiliki alamat selanjutnya untuk node berikutnya dan alamat sebelumnya untuk node sebelumnya secara berurutan.

Stack merupakan kumpulan data yang ditumpuk dari bawah ke atas secara berurutan sehingga jika ingin mengambil data yang paling bawah maka otomatis data yang paling atas akan diambil terlebih dahulu, Jadi data yang terakhir masuk akan keluar duluan.