**LAPORAN PRAKTIKUM**

**SESI VII**

**PRAKTIKUM COMP6362 – DATA STRUCTURES**

**KELAS BC20**



Oleh :

2440059495 – Bismo Agung Tri Achmad Bramantyo

**SEMESTER GENAP 2020/2021  
BINA NUSANTARA UNIVERSITY**

**MALANG**

## **Kode Program**

|  |
| --- |
| Source Code |
| #include <stdio.h>  #include <stdlib.h>  #include <string.h>  struct movie{      char name[21];      char genre[30];      int no;      int price;      struct move \*next;      struct movie \*prev;  }\*current, \*newnode;  struct movie \*head = NULL;  struct movie \*tail = NULL;      struct movie \*before = NULL;      struct movie \*after = NULL;      struct movie \*temp = NULL;  int number = 1;  void insert(){      newnode = (struct movie\*)malloc(sizeof(struct movie));      newnode->next = NULL;      newnode->prev = NULL;        do{      printf("\nInput Movie Name [3...21]: ");      scanf("%[^\n]", &\*newnode->name); getchar();      }while(strlen(newnode->name)<3 || strlen(newnode->name)>21);      do      {          printf("\nInput Movie Genre [3...30]: ");          scanf("%[^\n]", &\*newnode->genre); getchar();      } while (strlen(newnode->genre)<3 || strlen(newnode->genre)>30);      do      {          printf("Input Movie Price [$ 10...1000]: $ ");          scanf("%d", &newnode->price); getchar();      } while (newnode->price < 10 || newnode->price > 1000);      newnode->no = number;      if (head == NULL){          head = tail = newnode;          newnode->next = newnode;          newnode->prev = newnode;      }      else{          tail->next = newnode;          newnode->prev = tail;          tail = tail->next;          tail->next = head;          head->prev = tail;      }        printf("\n---Add New Movie Succes---\n");      number = number + 1;      getchar();      system("cls");      getchar();  }  void delete\_begin(){      if (head == NULL){          printf("\nNo data can be deleted\n");      }      else if(head == tail){          current = head;          head = tail = NULL;          free(current);          number = 1;      }      else{            current = head;          temp = current;          while (temp != head->prev){              temp->no = temp->no - 1;              temp = temp->next;          }            tail->no = tail->no - 1;            head = head->next;          head->prev = tail;          tail->next= head;          number = number - 1;          free(current);      }  }  void delete\_last(){      if (head == NULL){          printf("\nNo data can be deleted\n");      }      else if(head == tail){          current = head;          head = tail = NULL;          free(current);          number = 1;      }      else{          current = tail;          tail = tail->prev;          head->prev = tail;          tail->next= head;          number = number - 1;          free(current);        }  }  void delete\_position(int key){        current = head;      int found = 0;      if (head == NULL){          printf("\nNo data\n");      }      else{          while (current != NULL){          if(current->no == key){              found = 1;              break;           }           current = current->next;          }          }        if (found == 1){          if(current == head){              delete\_begin();          }          else if(current == tail){              delete\_last();          }          else{                temp = current;              while (temp != head ){              temp->no = temp->no - 1;              temp = temp->next;              }                before = current->prev;              after = current->next;              before->next = after;              after->prev = before;                number = number - 1;              free(current);          }          printf("\n---Success---\n");      }      else{          printf("\nData Not Found\n");      }  }  void display(){      current = head;      printf("\n\t---MOVIE LIST---\n\n");      printf("-+-----+----------------------------+---------------------------+---------+-\n");      printf(" : No. :     Name of Movie          :         Genre             :   Price : \n");      printf("-+-----+----------------------------+---------------------------+---------+-\n");      if(head == NULL){      printf("-+-----+----------------------------+---------------------------+---------+-\n");      }      else{          do{          printf(" : %2d. :    %-20s    :   %-20s   :   %3d   :\n", current->no, current->name, current->genre, current->price);          current = current ->next;      }while(current != head );      printf("-+-----+----------------------------+---------------------------+---------+-\n");    }  }  void delete\_all(){      while(head != NULL){          delete\_begin();      }  }  void main(){      int menu;      int key;      int parameter = 1;      while(parameter == 1){      printf("\tEXTRAVAGANT MOVIE SHOP\n");      printf("\t=======================\n\n");      printf("1. View Movie List\n");      printf("2. Add New Movie\n");      printf("3. Take Movie\n");      printf("4. Exit\n");      printf("Input Menu [1/2/3/4]: ");      scanf("%d", &menu); fflush(stdin);      switch (menu)      {      case 1:          display();          getchar();          system("cls");          break;        case 2:          insert();          break;      case 3:          if(head == NULL){              printf("\nNo Data\n");              getchar();              system("cls");          }          else{          display();          printf("\nInput the Number of the Movie[%d...%d]: ", head->no, tail->no);          scanf("%d", &key);          delete\_position(key);          getchar();          system("cls");          }          break;      case 4:          printf("\n----Program Closed---\n");          getchar();          system("cls");          delete\_all();          parameter = 0;          break;      default :            printf("\n---Please Input The Correct Menu---\n");          getchar();          system("cls");      }    }  } |

## **Deskripsi Program**

1. Library

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

1. Struct

struct movie{

    char name[21];

    char genre[30];

    int no;

    int price;

    struct move \*next;

    struct movie \*prev;

}\*current, \*newnode;

struct movie \*head = NULL;

struct movie \*tail = NULL;

    struct movie \*before = NULL;

    struct movie \*after = NULL;

    struct movie \*temp = NULL;

pada struct ini terdapat 4 variabel data, name untuk nama, genre untuk genre film, no untuk nomor urut film, price untuk harga film, kemudian terdapat 2 pointer utama dan 7 pointer bantu.

1. Variabel Universal

int number = 1;

variabel number untuk nomor urut film.

1. Fungsi
2. Insert

void insert(){

    newnode = (struct movie\*)malloc(sizeof(struct movie));

    newnode->next = NULL;

    newnode->prev = NULL;

    do{

    printf("\nInput Movie Name [3...21]: ");

    scanf("%[^\n]", &\*newnode->name); getchar();

    }while(strlen(newnode->name)<3 || strlen(newnode->name)>21);

    do

    {

        printf("\nInput Movie Genre [3...30]: ");

        scanf("%[^\n]", &\*newnode->genre); getchar();

    } while (strlen(newnode->genre)<3 || strlen(newnode->genre)>30);

    do

    {

        printf("Input Movie Price [$ 10...1000]: $ ");

        scanf("%d", &newnode->price); getchar();

    } while (newnode->price < 10 || newnode->price > 1000);

    newnode->no = number;

    if (head == NULL){

        head = tail = newnode;

        newnode->next = newnode;

        newnode->prev = newnode;

    }

    else{

        tail->next = newnode;

        newnode->prev = tail;

        tail = tail->next;

        tail->next = head;

        head->prev = tail;

    }

    printf("\n---Add New Movie Succes---\n");

    number = number + 1;

    getchar();

    system("cls");

    getchar();

}

Pada fungsi ini, akan terjadi alokasi memori terlebih dulu untuk newnode, kemudian pointer next dan prev dari newnode di NULL kan, lalu user diminta untuk menginput nama film yang dimana minimal panjang kalimat harus 3 sampai 21 (space included), kemudian user diminta untuk menginput genre dimana minimal panjang kalimat adalah 3 sampai 30 (space included), setelah itu user diminta menginput harga film minimal 10 maksimal 1000. Semua data itu akan langsung disimpan ke node baru. Ketika data kosong, maka pointer head, tail akan menunjuk ke node baru, pointer next dan prev akan menujuk ke newnode sendiri. Apabila data sudah ada, maka pointer next dari node tail akan mengarah ke node baru, kemudian pointer prev dari node baru akan mengarah ke tail, lalu pointer tail akan berpindah ke node baru, lalu pointer next dari tail akan mengarah ke head, dan prev head akan mengarah ke tail. Int number + 1.

1. Delete awal

void delete\_begin(){

    if (head == NULL){

        printf("\nNo data can be deleted\n");

    }

    else if(head == tail){

        current = head;

        head = tail = NULL;

        free(current);

        number = 1;

    }

    else{

        current = head;

        temp = current;

        while (temp != head->prev){

            temp->no = temp->no - 1;

            temp = temp->next;

        }

        tail->no = tail->no - 1;

        head = head->next;

        head->prev = tail;

        tail->next= head;

        number = number - 1;

        free(current);

    }

}

Apabila data kosong maka akan muncul error message. Apabila data hanya 1 maka pointer tail dan head akan di NULL kan lalu current di free, number dikembalikan nilai nya menjadi 1. Apabila data lebih dari 1 maka current akan menujuk ke head, lalu temp akan menunjuk ke current, data no pada setiap node akan di -1 (kecuali tail), lalu data no di tail dikurangi 1. Pointer head akan bergeser ke 1 data setelah nya, head prev akan menunjuk ke tail, dan tail next akan menunjuk ke head, current dihapus.

Int number – 1;

1. Delete akhir

void delete\_last(){

    if (head == NULL){

        printf("\nNo data can be deleted\n");

    }

    else if(head == tail){

        current = head;

        head = tail = NULL;

        free(current);

        number = 1;

    }

    else{

        current = tail;

        tail = tail->prev;

        head->prev = tail;

        tail->next= head;

        number = number - 1;

        free(current);

    }

}

Sama seperti delete awal, yang membedakan, pointer current akan menunjuk ke tail, lalu tail akan berpindah 1 data sebelum nya, kemudian head prev akan menunjuk ke tail yang baru, dan tail next (baru) akan menunjuk ke head, lalu current akan di free, int number – 1;

1. Delete at the given position

void delete\_position(int key){

    current = head;

    int found = 0;

    if (head == NULL){

        printf("\nNo data\n");

    }

    else{

        while (current != NULL){

        if(current->no == key){

            found = 1;

            break;

         }

         current = current->next;

        }

        }

    if (found == 1){

        if(current == head){

            delete\_begin();

        }

        else if(current == tail){

            delete\_last();

        }

        else{

            temp = current;

            while (temp != head ){

            temp->no = temp->no - 1;

            temp = temp->next;

            }

            before = current->prev;

            after = current->next;

            before->next = after;

            after->prev = before;

            number = number - 1;

            free(current);

        }

        printf("\n---Success---\n");

    }

    else{

        printf("\nData Not Found\n");

    }

}

Jika tidak ada data maka akan program akan mencetak error message, selain itu program akan mencari data dengan key, selama data masih ada cari data, apabila data sudah ketemu, hentikan perulangan(break), kemudian int found akan berubah menjadi 1, lalu apabila dta berada pada head, maka fungsi delete\_first akan direkursi, apabila data berada di tail maka fungsi delete\_last akan direkursi, selain itu maka data no pada setiap node setelah nya akan dikurangi 1, lalu node sebelum current, akan menunjuk ke node after current, dan prev current akan menunjuk before, , number akan di – 1, lalu current akan di free

1. Display

void display(){

    current = head;

    printf("\n\t---MOVIE LIST---\n\n");

    printf("-+-----+----------------------------+---------------------------+---------+-\n");

    printf(" : No. :     Name of Movie          :         Genre             :   Price : \n");

    printf("-+-----+----------------------------+---------------------------+---------+-\n");

    if(head == NULL){

    printf("-+-----+----------------------------+---------------------------+---------+-\n");

    }

    else{

        do{

        printf(" : %2d. :    %-20s    :   %-20s   :   %3d   :\n", current->no, current->name, current->genre, current->price);

        current = current ->next;

    }while(current != head );

    printf("-+-----+----------------------------+---------------------------+---------+-\n");

  }

}

Program akan mencetak header, apabila data kosong maka program akan langsung mnecetak footer, apabila ada data, maka selama data tidak menunjuk ke head lagi, data akan di print.

1. Delete All

void delete\_all(){

    while(head != NULL){

        delete\_begin();

    }

}

Selama data masih ada (head masih ada) maka node akan terus dihapus dari depan.

1. Main function
2. void main(){
3. int menu;
4. int key;
5. int parameter = 1;
6. while(parameter == 1){
7. printf("\tEXTRAVAGANT MOVIE SHOP\n");
8. printf("\t=======================\n\n");
9. printf("1. View Movie List\n");
10. printf("2. Add New Movie\n");
11. printf("3. Take Movie\n");
12. printf("4. Exit\n");
13. printf("Input Menu [1/2/3/4]: ");
14. scanf("%d", &menu); fflush(stdin);
15. switch (menu)
16. {
17. case 1:
18. display();
19. getchar();
20. system("cls");
21. break;
23. case 2:
24. insert();
25. break;
26. case 3:
27. if(head == NULL){
28. printf("\nNo Data\n");
29. getchar();
30. system("cls");
31. }
32. else{
33. display();
34. printf("\nInput the Number of the Movie[%d...%d]: ", head->no, tail->no);
35. scanf("%d", &key);
36. delete\_position(key);
37. getchar();
38. system("cls");
39. }
40. break;
41. case 4:
42. printf("\n----Program Closed---\n");
43. getchar();
44. system("cls");
45. delete\_all();
46. parameter = 0;
47. break;
48. default :
50. printf("\n---Please Input The Correct Menu---\n");
51. getchar();
52. system("cls");
53. }
54. }
55. }

# variabel

1. Menu = untuk switch
2. Key = untuk parameter fungsi delete
3. Parameter = untuk flag

Case 1: fungsi display akan direkursi

Case 2: fungsi insert akan direkursi

Case 3: apabila tidak ada data maka program akan mencetak error message, apabila ada maka program akan mencetak list dengan fungsi display, lalu user menginput nomor film yang ingin dihapus (key), kemudian fungsi delete\_position akan direkursi dengan passing parameter key.

Case 4: program akan berhenti berjalan

Default : ketika user menginput angka selain 1 2 3 4.

## **Bukti Presentasi**