**LAPORAN PRAKTIKUM**

**SESI VI**

**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 <string.h>  #include <stdlib.h>  struct family  {      char name[20];      char choice[20];      struct family \*next;      struct family \*prev;  } \*current, \*newnode;  struct family \*head = NULL;  struct family \*tail = NULL;  int cake\_counter = 0;  int money\_counter = 0;  int holiday\_counter = 0;  char message\_cake[]=("we will give a fancy yummy cake as a gift on a family member's birthday");  char message\_money[]=("we will give amount of money as a gift on a family member's birthday");  char message\_holiday[]=("we will give a holiday trip as a gift on a family member's birthday");  int number = 1;  void add\_first(char name[20], int choice){        newnode=(struct family\*)malloc(sizeof(struct family));      newnode->next = NULL;      newnode->prev = NULL;      strcpy(newnode->name, name);      if (choice == 1){          strcpy(newnode->choice,"Fancy yummy cake");      }      else if(choice == 2){          strcpy(newnode->choice,"Holiday trip");      }      else if(choice == 3){          strcpy(newnode->choice, "Money");      }        if(head == NULL){            head = tail = newnode;      }      else{          head->prev = newnode;          newnode->next = head;          head = newnode;      }      getchar();      system("cls");  }  void add\_last(char name[20], int choice){        newnode=(struct family\*)malloc(sizeof(struct family));      newnode->next = NULL;      newnode->prev = NULL;      strcpy(newnode->name, name);      if (choice == 1){          strcpy(newnode->choice,"Fancy yummy cake");      }      else if(choice == 2){          strcpy(newnode->choice,"Holiday trip");      }      else if(choice == 3){          strcpy(newnode->choice, "Money");      }      if(head == NULL){            head = tail = newnode;      }      else{          tail->next = newnode;          newnode->prev = tail;          tail = newnode;      }      getchar();      system("cls");  }  void input(){        int choice;      int type;      char name[20];      printf("1. Add before the first data\n");      printf("2. Add after the last data\n");      printf("Input Your Choice [1/2]: ");      scanf("%d", &choice); fflush(stdin);      switch(choice){          case 1:          printf("Input family member's name: ");          scanf("%[^\n]", &name); fflush(stdin);          do{          printf("\n 1 = Fancy yummy cake    2 = Holiday trip    3 = Money\n");          printf("Select the birthday gift [1/2/3]: ");          scanf("%d", &type); getchar();          }while(type != 1 && type != 2 && type != 3);          if(type == 1){              cake\_counter = cake\_counter + 1;            }          else if(type == 2 ){              holiday\_counter = holiday\_counter + 1;            }          else if (type == 3)          {              money\_counter = money\_counter + 1;            }            add\_first(name, type);          break;          case 2:          printf("Input family member's name: ");          scanf("%[^\n]", &name); fflush(stdin);          do{          printf("\n 1 = Fancy yummy cake    2 = Holiday trip    3 = Money\n");          printf("Select the birthday gift [1/2/3]: ");          scanf("%d", &type); getchar();            }while(type != 1 && type != 2 && type != 3);          if(type == 1){              cake\_counter = cake\_counter + 1;            }          else if(type == 2){              holiday\_counter = holiday\_counter + 1;            }          else if (type == 3)          {              money\_counter = money\_counter + 1;            }            add\_last(name, type);      }  }  void delete\_first(){      if (head == NULL){          printf("\n---no data can be deleted---\n");          getchar();          system("cls");      }      else if(head == tail){          current = head;          if(strcmp(current->choice,"Fancy yummy cake")==0){              cake\_counter = cake\_counter - 1;            }          else if(strcmp(current->choice,"Holiday trip")==0){              holiday\_counter = holiday\_counter - 1;            }          else if (strcmp(current->choice,"Money")==0)          {              money\_counter = money\_counter - 1;            }          head = tail = NULL;          printf("\n\n---- %s's opinion has been deleted----", current->name);          getchar();          system("cls");          free(current);      }      else{          current = head;          if(strcmp(current->choice,"Fancy yummy cake")==0){              cake\_counter = cake\_counter - 1;            }          else if(strcmp(current->choice,"Holiday trip")==0){              holiday\_counter = holiday\_counter - 1;            }          else if (strcmp(current->choice,"Money")==0)          {              money\_counter = money\_counter - 1;            }          head = head -> next;          current->next = NULL;          head->prev = NULL;          printf("\n\n---- %s's opinion has been deleted----", current->name);          free(current);          getchar();          system("cls");      }  }  void delete\_last(){      if (head == NULL){          printf("\n---no data can be deleted---\n");          getchar();          system("cls");      }      else if(head == tail){          current = tail;            if(strcmp(current->choice,"Fancy yummy cake")==0){              cake\_counter = cake\_counter - 1;            }          else if(strcmp(current->choice,"Holiday trip")==0){              holiday\_counter = holiday\_counter - 1;            }          else if (strcmp(current->choice,"Money")==0)          {              money\_counter = money\_counter - 1;            }          head = tail = NULL;            printf("\n\n---- %s's opinion has been deleted----", current->name);          getchar();          system("cls");          free(current);      }      else{          current = tail;          if(strcmp(current->choice,"Fancy yummy cake")==0){              cake\_counter = cake\_counter - 1;            }          else if(strcmp(current->choice,"Holiday trip")==0){              holiday\_counter = holiday\_counter - 1;            }          else if (strcmp(current->choice,"Money")==0)          {              money\_counter = money\_counter - 1;            }          tail = tail->prev;          current->prev = NULL;          tail->next = NULL;          printf("\n\n---- %s's opinion has been deleted----", current->name);          free(current);          getchar();          system("cls");      }  }  void delete\_data(){      int choice;      printf("\n1= Remove the first data\n");      printf("2= Remove the last data\n");      do{      printf("\nInput your choice [1/2]: ");      scanf("%d", &choice); getchar();      }while(choice != 1 && choice != 2);      switch(choice){          case 1:          delete\_first();          break;          case 2:          delete\_last();          break;      }  }  void delete\_all(){        while(head != NULL){          current = head;          head = head->next;          free(current);      }   cake\_counter = 0;   money\_counter = 0;   holiday\_counter = 0;     printf("\n---Success---\n");   getchar();   system("cls");  }  void display(){      current = head;      if (head == NULL){          printf("\t--- Determining a birthday gift ---\n\n");          printf("==============================================================\n");          printf("| No.  | Family member's name   |       Birthday gift        |\n");          printf("==============================================================\n");          printf(":                                                            :\n");          printf(":                     --- no data here ---                   :\n");          printf(":                                                            :\n");          printf("==============================================================\n\n\n");      }      else{          printf("\t--- Determining a birthday gift ---\n\n");          printf("==============================================================\n");          printf("| No.  | Family member's name   |       Birthday gift        |\n");          printf("==============================================================");          do{              printf("\n: %2d. : %-12s            : %-20s       :", number, current->name, current->choice);                if(current==head){                  printf("<-- Head ");              }              if(current==tail){                  printf("<-- Tail\n");              }              current = current->next;              number++;          }while(current != NULL);          printf("==============================================================\n\n\n");          number = 1;      }  }  int main(){      int menu;      int parameter = 1;      while(parameter == 1){        display();      printf("1. Add Polling Data\n");      printf("2. Delete a Polling Data\n");      printf("3. Show the Polling Result\n");      printf("4. Delete list\n");      printf("5. Exit\n\n");      do{      printf(">> Input Your Choice [1/2/3/4/5]: ");      scanf("%d", &menu); getchar();      }while(menu != 1 && menu != 2 && menu != 3 && menu != 4 && menu != 5);      switch(menu){            case 1:              input();          break;          case 2:              delete\_data();          break;          case 3:              printf("\nThe Polling Result so far are, \n");              printf("Fancy yummy cake : %d , Holiday trip: %d , Money = %d \n", cake\_counter, holiday\_counter, money\_counter);                if (cake\_counter > holiday\_counter && cake\_counter > money\_counter){                  printf("\nBased on this polling, \n%s\n", message\_cake);              }              else if (holiday\_counter > cake\_counter && holiday\_counter > money\_counter){                  printf("\nBased on this polling, \n%s\n", message\_holiday);              }              else if (money\_counter > holiday\_counter && money\_counter > cake\_counter){                  printf("\nBased on this polling, \n%s\n", message\_money);              }              else{                  printf("\ninsert or remove a polling data.\n");              }              getchar();              system("cls");          break;            case 4:              if (head == NULL){                  printf("\n---no data can be deleted---\n");                  getchar();                  system("cls");              }              else{                  delete\_all();              }            break;            case 5:              printf("\n-----Thank You , Program closed----\n");              parameter = 0;          break;      }      }  } |

## **Deskripsi Program**

1. Library

#include <stdio.h>

#include <string.h>

#include <stdlib.h>

1. Struct

struct family

{

    char name[20];

    char choice[20];

    struct family \*next;

    struct family \*prev;

} \*current, \*newnode;

Di struct ini saya menggunakan 2 variabel dan 2 pointer utama

Name = menyimpan nama

Choice = menyimpan pilihan kado

\*next = pointer ke node berikutnya

\*previous = Pointer ke node sebelumnya

1. Variabel universal

struct family \*head = NULL;

struct family \*tail = NULL;

int cake\_counter = 0;

int money\_counter = 0;

int holiday\_counter = 0;

Disini ada pointer untuk membantu yaitu head dan tail

\*head = data pertama

\*tail = data terakhir

Serta variabel untuk menampung jumlah data tiap hadiah.

1. Fungsi
2. Input awal

void add\_first(char name[20], int choice){

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

    newnode->next = NULL;

    newnode->prev = NULL;

    strcpy(newnode->name, name);

    if (choice == 1){

        strcpy(newnode->choice,"Fancy yummy cake");

    }

    else if(choice == 2){

        strcpy(newnode->choice,"Holiday trip");

    }

    else if(choice == 3){

        strcpy(newnode->choice, "Money");

    }

    if(head == NULL){

        head = tail = newnode;

    }

    else{

        head->prev = newnode;

        newnode->next = head;

        head = newnode;

    }

    getchar();

    system("cls");

}

Dalam void ini ada 2 passing parameter , yaitu name dan choice,

Alokasi memori akan dilakukan terlebih dahulu, lalu pointer next dan previous akan di NULL kan terlebih dahulu. Kemudian program akan mengecek int choice, apabila user memilih satu maka akan dituliskan string Fancy yummy cake, apabila 2, maka string Holiday trip yang akan dituliskan, selain itu maka Money yang dituliskan.

Kemudian ketika tidak ada data maka pointer head dan tail akan menunjuk ke node baru, jika ada data maka pointer previous dari head akan menunjuk node baru, kemudian pointer next dari newnode akan menunjuk ke head, lalu pointer head akan berpindah ke newnode.

1. Input akhir

void add\_last(char name[20], int choice){

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

    newnode->next = NULL;

    newnode->prev = NULL;

    strcpy(newnode->name, name);

    if (choice == 1){

        strcpy(newnode->choice,"Fancy yummy cake");

    }

    else if(choice == 2){

        strcpy(newnode->choice,"Holiday trip");

    }

    else if(choice == 3){

        strcpy(newnode->choice, "Money");

    }

    if(head == NULL){

        head = tail = newnode;

    }

    else{

        tail->next = newnode;

        newnode->prev = tail;

        tail = newnode;

    }

    getchar();

    system("cls");

}

Kurang lebih sama dengan insert di awal, yang membedakan hanya pada penempatan node baru yaitu setelah tail, pointer next node tail akan menunjuk node baru, pointer prev node baru akan menunjuk tail , kemudian pointer tail menunjuk node baru.

1. Input utama

void input(){

    int choice;

    int type;

    char name[20];

    printf("1. Add before the first data\n");

    printf("2. Add after the last data\n");

    printf("Input Your Choice [1/2]: ");

    scanf("%d", &choice); fflush(stdin);

    switch(choice){

        case 1:

        printf("Input family member's name: ");

        scanf("%[^\n]", &name); fflush(stdin);

        do{

        printf("\n 1 = Fancy yummy cake    2 = Holiday trip    3 = Money\n");

        printf("Select the birthday gift [1/2/3]: ");

        scanf("%d", &type); getchar();

        }while(type != 1 && type != 2 && type != 3);

        if(type == 1){

            cake\_counter = cake\_counter + 1;

        }

        else if(type == 2 ){

            holiday\_counter = holiday\_counter + 1;

        }

        else if (type == 3)

        {

            money\_counter = money\_counter + 1;

        }

        add\_first(name, type);

        break;

        case 2:

        printf("Input family member's name: ");

        scanf("%[^\n]", &name); fflush(stdin);

        do{

        printf("\n 1 = Fancy yummy cake    2 = Holiday trip    3 = Money\n");

        printf("Select the birthday gift [1/2/3]: ");

        scanf("%d", &type); getchar();

        }while(type != 1 && type != 2 && type != 3);

        if(type == 1){

            cake\_counter = cake\_counter + 1;

        }

        else if(type == 2){

            holiday\_counter = holiday\_counter + 1;

        }

        else if (type == 3)

        {

            money\_counter = money\_counter + 1;

        }

        add\_last(name, type);

    }

}

Dalam fungsi ini terdapat 3 variabel yaitu choice (untuk memilih menambah dari depan atau belakang) , name (nama) dan type (jenis hadiah).

Pertama user akan diminta untuk memilih menambah data dari depan atau belakang.

Apabila user memilih 1 maka

User akan diminta menginput nama anggota keluarga

Kemudian user diminta untuk memilih hadiah ulang tahun, hanya 1 2 3 selain itu program akan meminta user untuk menginput kembali,

Lalu apabila input 1 maka cake counter akan ditambah 1

Apabila 2 maka holiday counter yang akan ditambah 1

Apabila 3 maka money counter yang akan ditambah 1

Kemudian fungsi add\_first di rekursi

Apabila user memilih 2

User akan diminta menginput nama anggota keluarga

Kemudian user diminta untuk memilih hadiah ulang tahun, hanya 1 2 3 selain itu program akan meminta user untuk menginput kembali,

Lalu apabila input 1 maka cake counter akan ditambah 1

Apabila 2 maka holiday counter yang akan ditambah 1

Apabila 3 maka money counter yang akan ditambah 1

Kemudian fungsi add\_last di rekursi

1. Delete awal

void delete\_first(){

    if (head == NULL){

        printf("\n---no data can be deleted---\n");

        getchar();

        system("cls");

    }

    else if(head == tail){

        current = head;

        if(strcmp(current->choice,"Fancy yummy cake")==0){

            cake\_counter = cake\_counter - 1;

        }

        else if(strcmp(current->choice,"Holiday trip")==0){

            holiday\_counter = holiday\_counter - 1;

        }

        else if (strcmp(current->choice,"Money")==0)

        {

            money\_counter = money\_counter - 1;

        }

        head = tail = NULL;

        printf("\n\n---- %s's opinion has been deleted----", current->name);

        getchar();

        system("cls");

        free(current);

    }

    else{

        current = head;

        if(strcmp(current->choice,"Fancy yummy cake")==0){

            cake\_counter = cake\_counter - 1;

        }

        else if(strcmp(current->choice,"Holiday trip")==0){

            holiday\_counter = holiday\_counter - 1;

        }

        else if (strcmp(current->choice,"Money")==0)

        {

            money\_counter = money\_counter - 1;

        }

        head = head -> next;

        current->next = NULL;

        head->prev = NULL;

        printf("\n\n---- %s's opinion has been deleted----", current->name);

        free(current);

        getchar();

        system("cls");

    }

}

Apabila data kosong maka program akan menampilkan pesan yang menyatakan bahwa tidak ada data untuk dilakukan operasi.

Apabila data hanya ada 1 maka pointer current akan menunjuk pada node head, lalu dicek string pada data choice di node tersebut, apabila sama dengan string yang sudah ditentukan maka penghitung hadiah yang sesuai akan dikurangi jumlah nya, kemudian head dan tail akan di NULL kan lalu program mencetak data nama pada node current bahwa data tersebut akan dihapus, lalu current dihapus.

Apabila data lebih dari 1

Maka pointer current akan menunjuk ke head, data akan dicek dan disesuaikan , kemudian head bergeser 1 node setelahnya, kemudian pointer next current akan di null kan , begitu juga dengan pointer previous head, lalu pesan dengan data nama di node tersebut akan dicetak, lalu node (current) dihapus.

1. Delete akhir

void delete\_last(){

    if (head == NULL){

        printf("\n---no data can be deleted---\n");

        getchar();

        system("cls");

    }

    else if(head == tail){

        current = tail;

        if(strcmp(current->choice,"Fancy yummy cake")==0){

            cake\_counter = cake\_counter - 1;

        }

        else if(strcmp(current->choice,"Holiday trip")==0){

            holiday\_counter = holiday\_counter - 1;

        }

        else if (strcmp(current->choice,"Money")==0)

        {

            money\_counter = money\_counter - 1;

        }

        head = tail = NULL;

        printf("\n\n---- %s's opinion has been deleted----", current->name);

        getchar();

        system("cls");

        free(current);

    }

    else{

        current = tail;

        if(strcmp(current->choice,"Fancy yummy cake")==0){

            cake\_counter = cake\_counter - 1;

        }

        else if(strcmp(current->choice,"Holiday trip")==0){

            holiday\_counter = holiday\_counter - 1;

        }

        else if (strcmp(current->choice,"Money")==0)

        {

            money\_counter = money\_counter - 1;

        }

        tail = tail->prev;

        current->prev = NULL;

        tail->next = NULL;

        printf("\n\n---- %s's opinion has been deleted----", current->name);

        free(current);

        getchar();

        system("cls");

    }

}

Kurang lebih sama seperti delete\_first, yang membedakan hanya pointer current menunjuk pada tail, pointer tail akan bergeser 1 node sebelumnya, kemudian pointer previous dari node current akan di NULL kan, kemudian pointer next dari tail akan di NULL kan juga, lalu pesan akan muncul dan current di hapus.

1. Delete utama

void delete\_data(){

    int choice;

    printf("\n1= Remove the first data\n");

    printf("2= Remove the last data\n");

    do{

    printf("\nInput your choice [1/2]: ");

    scanf("%d", &choice); getchar();

    }while(choice != 1 && choice != 2);

    switch(choice){

        case 1:

        delete\_first();

        break;

        case 2:

        delete\_last();

        break;

    }

}

Pada fungsi ini hanya ada 1 variabel yaitu choice untuk memilih delete awal / akhir, apabila user menginput 1 maka fungsi delete\_first akan di rekursi , apabila user memilih 2 maka fungsi delete\_last akan direkursi, selain itu user akan diminta menginput lagi.

1. Delete semua data

void delete\_all(){

    while(head != NULL){

        current = head;

        head = head->next;

        free(current);

    }

 cake\_counter = 0;

 money\_counter = 0;

 holiday\_counter = 0;

 printf("\n---Success---\n");

 getchar();

 system("cls");

}

Selama headtidak sama dengan NULL / dta kosong maka, pointer current akan menunjuk ke head, lalu pointer head akan bergeser 1 node setelahnya, kemudian current akan di hapus, begitu terus sampai data habis, semua penghitung jenis hadiah ulang tahun akan di 0 kan.

1. Display

void display(){

    current = head;

    if (head == NULL){

        printf("\t--- Determining a birthday gift ---\n\n");

        printf("==============================================================\n");

        printf("| No.  | Family member's name   |       Birthday gift        |\n");

        printf("==============================================================\n");

        printf(":                                                            :\n");

        printf(":                     --- no data here ---                   :\n");

        printf(":                                                            :\n");

        printf("==============================================================\n\n\n");

    }

    else{

        printf("\t--- Determining a birthday gift ---\n\n");

        printf("==============================================================\n");

        printf("| No.  | Family member's name   |       Birthday gift        |\n");

        printf("==============================================================");

        do{

            printf("\n: %2d. : %-12s            : %-20s       :", number, current->name, current->choice);

            if(current==head){

                printf("<-- Head ");

            }

            if(current==tail){

                printf("<-- Tail\n");

            }

            current = current->next;

            number++;

        }while(current != NULL);

        printf("==============================================================\n\n\n");

        number = 1;

    }

}

Di fungsi ini pointer current akan menujuk ke head.

Apabila head kosong / tidak ada data maka program akan menampilkan tabel kosong tanpa data.

Namun apabila ada, maka program akan mencetak data tersebut sampai current = NULL / tidak ada data lagi yang belum dicetak, int number akan ditambah 1 terus, setelah data habis maka perulangan akan berhenti lali int number dikembalikan nilai nya menjadi 1 (agar konstan nilainya)

1. Main Function

int main(){

    int menu;

    int parameter = 1;

    while(parameter == 1){

    display();

    printf("1. Add Polling Data\n");

    printf("2. Delete a Polling Data\n");

    printf("3. Show the Polling Result\n");

    printf("4. Delete list\n");

    printf("5. Exit\n\n");

    do{

    printf(">> Input Your Choice [1/2/3/4/5]: ");

    scanf("%d", &menu); getchar();

    }while(menu != 1 && menu != 2 && menu != 3 && menu != 4 && menu != 5);

    switch(menu){

        case 1:

            input();

        break;

        case 2:

            delete\_data();

        break;

        case 3:

            printf("\nThe Polling Result so far are, \n");

            printf("Fancy yummy cake : %d , Holiday trip: %d , Money = %d \n", cake\_counter, holiday\_counter, money\_counter);

            if (cake\_counter > holiday\_counter && cake\_counter > money\_counter){

                printf("\nBased on this polling, \n%s\n", message\_cake);

            }

            else if (holiday\_counter > cake\_counter && holiday\_counter > money\_counter){

                printf("\nBased on this polling, \n%s\n", message\_holiday);

            }

            else if (money\_counter > holiday\_counter && money\_counter > cake\_counter){

                printf("\nBased on this polling, \n%s\n", message\_money);

            }

            else{

                printf("\ninsert or remove a polling data.\n");

            }

            getchar();

            system("cls");

        break;

        case 4:

            if (head == NULL){

                printf("\n---no data can be deleted---\n");

                getchar();

                system("cls");

            }

            else{

                delete\_all();

            }

        break;

        case 5:

            printf("\n-----Thank You , Program closed----\n");

            parameter = 0;

        break;

    }

    }

}

Untuk main function ada 2 variabel yaitu menu dan parameter, menu untuk melakukan switch case dan parameter untuk perulangan program. User akan ditampilkan tabel dengan rekursi fungsi display(), kemudian user akan diminta menginput menu, apabila user menginput angka selain yang telah disediakan maka user akan diminta menginput lagi.

Apabila user memilih

1 = maka fungsi input() akan direkursi

2 = maka fungsi delete\_data() akan direkursi

3 = maka program akan menampilkan hasil poling, lalu mengecek hadiah mana yang paling banyak, dan message sesuai hadiah akan ditampilkan, apabila dat 0 atau berjumlah sama, maka program akan menampilkan pesan yang menyuruh user untuk menginput atau menghapus data.

4= maka fungsi delete\_all() akan direkursi

5= program tertutup.

## **Bukti Presentasi**

