|  |  |
| --- | --- |
| Nama | : La Ode Muhammad Gazali |
| Kelas | : 2KS2 |
| NIM | : 222212696 |

**MODUL 7 PRAKTIKUM STRUKTUR DATA**

* **Hasil modifikasi versi Linked list**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

// Node

typedef struct node

{

char nama[20];

int alpro;

int kalkulus;

struct node \*next;

} mhs;

int count = 0;

// Function to Create A New Node

mhs \*newmhs(char a[], int alp, int kal)

{

mhs \*temp = (mhs \*)malloc(sizeof(mhs));

strcpy(temp->nama, a);

temp->alpro = alp;

temp->kalkulus = kal;

temp->next = NULL;

return temp;

}

// menghapus pendaftar

void dequeue(mhs \*\*head)

{

if ((\*head) != NULL)

{

mhs \*temp = \*head;

(\*head) = (\*head)->next;

free(temp);

}

}

// Function to push according to priority

void enqueue(mhs \*\*head, char n[], int alp, int kal)

{

mhs \*temp = newmhs(n, alp, kal);

if ((\*head) == NULL)

{

(\*head) = temp;

}

else if (((\*head)->alpro < alp) || (alp == (\*head)->alpro && kal > (\*head)->kalkulus))

{

temp->next = \*head;

(\*head) = temp;

}

else

{

mhs \*start = (\*head);

while (start->next != NULL && start->next->alpro > alp)

{

start = start->next;

}

if (start->next != NULL)

{

while (start->next != NULL && start->next->alpro == alp && start->next->kalkulus > kal)

{

start = start->next;

}

}

temp->next = start->next;

start->next = temp;

}

if (count >= 5)

{

mhs \*current = \*head;

while (current->next->next != NULL)

{

current = current->next;

}

dequeue(&(current->next));

}

else

{

count++;

}

}

void display(mhs \*head)

{

if (count == 0)

{

printf("Belum ada yang daftar\n");

}

else

{

printf("Daftar urutan dari nilai terbaik:\n");

for (int i = count; i > 0; i--)

{

printf("Nama: %s | Alpro: %d | Kalkulus: %d\n", head->nama, head->alpro, head->kalkulus);

head = head->next;

}

}

}

int main()

{

mhs \*wakil = NULL;

enqueue(&wakil, "Eko", 50, 20);

enqueue(&wakil, "Budi", 50, 20);

enqueue(&wakil, "bambang", 60, 20);

enqueue(&wakil, "Eka", 60, 20);

enqueue(&wakil, "wawo", 60, 20);

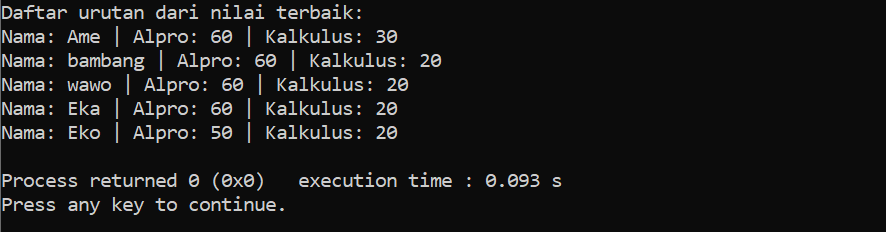
enqueue(&wakil, "Ame", 60, 30);

display(wakil);

return 0;

}

**Output:**



* **Hasil modifikasi dengan Array**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

typedef struct

{

char nama[20];

int alpro;

int kalkulus;

} Mahasiswa;

int count = 0;

void enqueue(Mahasiswa arr[], char nama[], int alpro, int kalkulus)

{

if (count < 5)

{

strcpy(arr[count].nama, nama);

arr[count].alpro = alpro;

arr[count].kalkulus = kalkulus;

(count)++;

}

else

{

int minIndex = 0;

for (int i = 1; i < 5; i++)

{

if (arr[i].alpro < arr[minIndex].alpro ||

(arr[i].alpro == arr[minIndex].alpro && arr[i].kalkulus < arr[minIndex].kalkulus))

{

minIndex = i;

}

}

if (alpro > arr[minIndex].alpro || (alpro == arr[minIndex].alpro && kalkulus > arr[minIndex].kalkulus))

{

strcpy(arr[minIndex].nama, nama);

arr[minIndex].alpro = alpro;

arr[minIndex].kalkulus = kalkulus;

}

}

}

void tampilkanMahasiswa(Mahasiswa arr[], int count)

{

if (count == 0)

{

printf("Belum ada yang daftar\n");

}

else

{

printf("Daftar urutan dari nilai terbaik:\n");

for (int i = 0; i < count; i++)

{

printf("Nama: %s | Alpro: %d | Kalkulus: %d\n", arr[i].nama, arr[i].alpro, arr[i].kalkulus);

}

}

}

int main()

{

Mahasiswa mahasiswa[5];

enqueue(mahasiswa, "Budi", 50, 20);

enqueue(mahasiswa, "Eko", 50, 20);

enqueue(mahasiswa, "bambang", 60, 20);

enqueue(mahasiswa, "Eka", 60, 20);

enqueue(mahasiswa, "wawo", 60, 20);

enqueue(mahasiswa, "Ame", 60, 30);

tampilkanMahasiswa(mahasiswa, count);

return 0;

}

**Output:**

