***Tugas Pendahuluan 7***

**PRAKTIKUM STRUKTUR DATA**

**“Heaps dan Priority Queue”**



Asisten :

1. Muh. Azrial Mahesa
2. Niswa Ayu Lestari

Oleh :

Nama : Nursyamsu rijal usman

Nim : 60900121070

Kelas : C

**LABORATORIUM KOMPUTER TERPADU**

**JURUSAN SISTEM INFORMASI**

**FAKULTAS SAINS DAN TEKNOLOGI**

**UNIVERSITAS ISLAM NEGERI ALAUDDIN MAKASSAR**

**2022**

|  |
| --- |
| * **Soal**   1. Apa itu Heaps dan Priority Queue  2. Pelajari Kode Program antrian C++ pada nomor 2 langkah Praktikum modul 4, kemudian kembangkan antrian menjadi antrian prioritas  3. Buat sebuah program deque dengan prioritas   * **Jawaban**   **1.** heap adalah struktur data yang berdasarkan konsep struktur data pohon. Pengertian Priority Queue. Priority. Queue. adalah. Queue. dengan. basis. HPIFO. (Highest. Priority. In. First. Out).  **2. Kode :**  #include <iostream>  #include <cstdio>  #include <cstring>  #include <cstdlib>  using namespace std;  struct node  {  string namaM;  int nim;  struct node \*link;  };  class Priority\_Queue  {  private:  node \*front;  public:  Priority\_Queue()  {  front = NULL;  }  void insert(string nama, int nimm)  {  node \*tmp, \*q;  tmp = new node;  tmp->namaM = nama;  tmp->nim = nimm;  if (front == NULL || nimm > front->nim)  {  tmp->link = front;  front = tmp;  }  else  {  q = front;  while (q->link != NULL && q->link->nim >= nimm)  q = q->link;  tmp->link = q->link;  q->link = tmp;  }  }  void del()  {  node \*tmp;  if (front == NULL)  cout << "ANTRIAN KOSONG\n";  else  {  tmp = front;  cout << "MENGELUARKAN : " << tmp->namaM << endl;  front = front->link;  free(tmp);  }  }  void display()  {  node \*ptr;  ptr = front;  if (front == NULL)  cout << "ANTRIAN KOSONG\n";  else  {  cout << "ANTRIAN :\n";  cout << "=====================" << endl;  cout << "NIM NAMA\n";  cout << "=====================" << endl;  while (ptr != NULL)  {  cout << ptr->nim << " " << ptr->namaM << endl;  ptr = ptr->link;  }  }  }  };  int main()  {  int choice, nim;  string nama;  Priority\_Queue pq;  do  {  cout << "=====================" << endl;  cout << "1.QUEUE\n";  cout << "2.DEQUE\n";  cout << "3.DISPLAY\n";  cout << "4.Quit\n";  cout << "=====================" << endl;  cout << "PILIH[1-4] : ";  cin >> choice;  switch (choice)  {  case 1:  cout << "MASUKKAN NAMA : ";  cin >> nama;  cout << "MASUKKAN NIM : ";  cin >> nim;  pq.insert(nama, nim);  break;  case 2:  pq.del();  break;  case 3:  pq.display();  break;  case 4:  break;  default:  cout << "PILIHAN TIDAK ADA\n";  }  } while (choice != 4);  return 0;  }  **Output :**    **3. Kode :**  #include <iostream>  using namespace std;  struct Node  {  int data, p;  Node \*next;  };  struct Node \*f = NULL;  struct Node \*r = NULL;  void enqueue(int d, int pr)  {  Node \*tmp, \*q;  tmp = new Node;  tmp->data = d;  tmp->p = pr;  if (r == NULL || pr > r->p)  {  tmp->next = r;  r = tmp;  }  else  {  q = r;  while (q->next != NULL && q->next->p >= pr)  q = q->next;  tmp->next = q->next;  q->next = tmp;  }  }  void display()  {  Node \*temp = r;  while (temp != NULL)  {  cout << "\tNO PASIAN : " << temp->data << " STADIUM : " << temp->p << endl;  temp = temp->next;  }  }  void dequeue()  {  Node \*temp = new Node;  if (r == NULL)  {  cout << "ANTRIAN KOSONG\n";  return;  }  else  {  temp = r;  r = r->next;  cout << "\n\t----------------------------------------";  cout << "\n\tMEMANGGIL PASIEN: " << temp->data << " \nDIPRIORITASKAN KARENA STADIUM " << temp->p;  cout << "\n\t----------------------------------------\n";  free(temp);  }  }  int main()  {  int qdata, qprior, choice, numberOfElements;  cout << "\n\t\t\_\_\_\_\_\_\_\_\_OPERATE PRIORITY DEQUEU\_\_\_\_\_\_\_\_\_\n\n";  cout << "\t1. QUEUE.\n\t2. Display.\n\t3. DEQUEU ELEMEN TERBESAR\n\t0. Exit\n\n";  do  {  cout << "\n\tPILIH[0-3] : ";  cin >> choice;  switch (choice)  {  case 1:  cout << "\n\tMASUKKAN NOMER PASIEN : ";  cin >> numberOfElements;  cout << "\n\tPRIOTAS: ";  cin >> qprior;  enqueue(numberOfElements, qprior);  break;  case 2:  cout << "\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n\n";  display();  cout << "\t\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";  cout << "\n\n";  break;  case 3:  dequeue();  break;  default:  cout << "\nEXIT \n\n";  break;  }  } while (choice != 0);  return 0;  }  **Output :** |