**LAPORAN RESMI 7**

**PRAKTIKUM ALGORITMA DAN STRUKTUR DATA**

QUEUE



Nama : Aqilah Akmalia Dewi

Kelas : 1 D4 IT B

NRP : 3120600046

PROGRAM STUDI D4 TEKNIK INFORMATIKA

POLITEKNIK ELEKTRONIKA NEGERI SURABAYA

2021

**DAFTAR ISI**

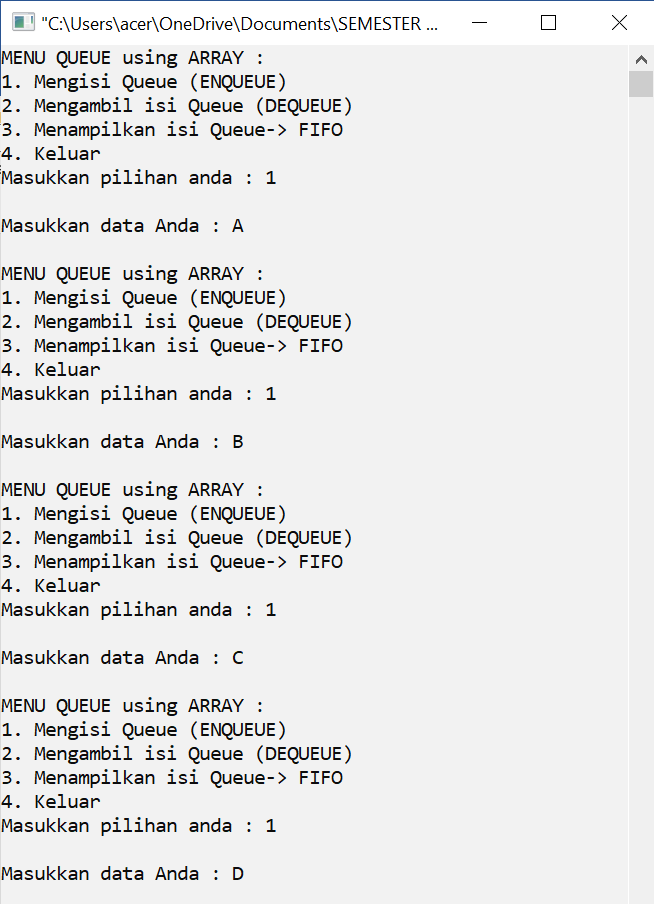
[1. Menu Queue Using Array 3](#_Toc68644435)

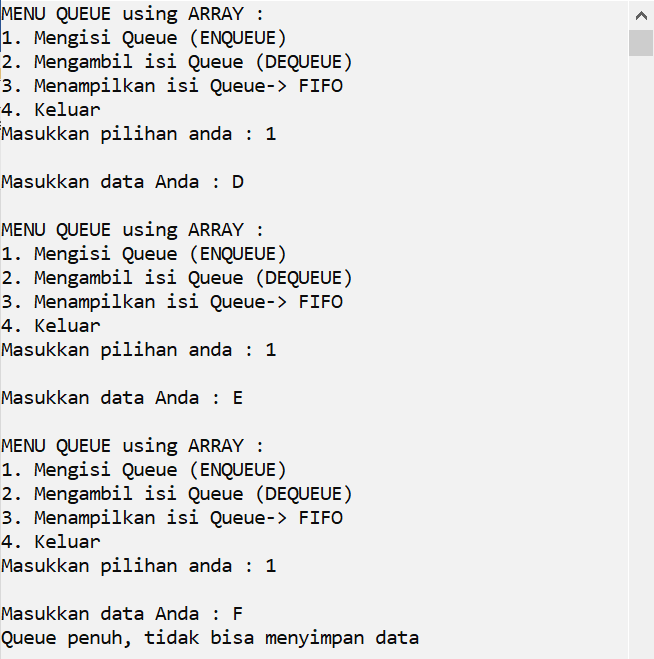
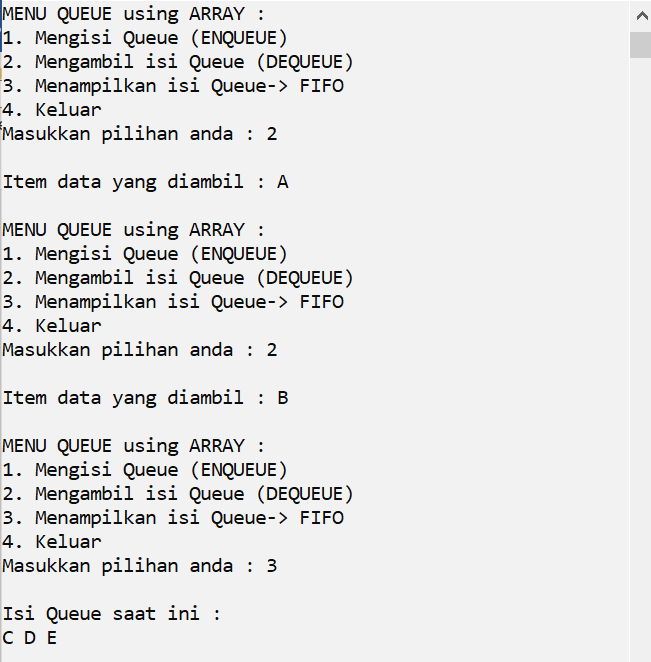
[2. Menu Queue Using Linked List 6](#_Toc68644436)

# 1. Menu Queue Using Array

1. Listing Program

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  #define MAX 5  typedef struct{  char item[MAX];  int front;  int rear;  int count;  }queue;  void inisialisasi(queue \*q);  int kosong(queue \*q);  int penuh(queue \*q);  void enqueue(char x, queue \*q);  char dequeue(queue \*q);  void tampil(queue \*q);  int main(){  queue antrian;  char kar;  int pil;  inisialisasi(&antrian);  do{  printf("MENU QUEUE using ARRAY :\n");  printf("1. Mengisi Queue (ENQUEUE)\n");  printf("2. Mengambil isi Queue (DEQUEUE)\n");  printf("3. Menampilkan isi Queue-> FIFO\n");  printf("4. Keluar\n");  printf("Masukkan pilihan anda : ");  scanf("%d",&pil);  switch(pil)  {  case 1:  fflush(stdin);  puts("");  printf("Masukkan data Anda : ");  scanf("%c", &kar);  enqueue(kar, &antrian);  fflush(stdin);  break;  case 2:  fflush(stdin);  puts("");  printf("Item data yang diambil : %c", dequeue(&antrian));  puts("");  fflush(stdin);  break;  case 3:  puts("");  tampil(&antrian);  break;  default:  exit(0);  }  puts("");  }while(pil!=4);  return 0;  }  void inisialisasi(queue \*q){  q->count = 0;  q->front = 0;  q->rear = 0;  }  int kosong(queue \*q){  if(q->count == 0)  return 1;  else  return 0;  }  int penuh(queue \*q){  if(q->count == MAX)  return 1;  else  return 0;  }  void enqueue(char x, queue \*q){  if(penuh(q))  printf("Queue penuh, tidak bisa menyimpan data\n");  else{  q->item[q->rear] = x;  q->rear = (q->rear + 1) % MAX;  (q->count)++;  }  }  char dequeue(queue \*q){  char ambil;  if(kosong(q))  printf("Queue kosong, tidak bisa mengambil data\n");  else{  ambil = q->item[q->front];  q->front = (q->front + 1) % MAX;  (q->count)--;  }  return(ambil);  }  void tampil(queue \*q){  int x,y;  char hasil;  x = q->front;  y = q->count;  printf("Isi Queue saat ini : \n");  if(kosong(q)){  printf("Queue Kosong");  }else{  while(y!=0){  hasil=q->item[x];  x = (x+1) % MAX;  y--;  printf("%c ",hasil);  }  puts("");  }  } |

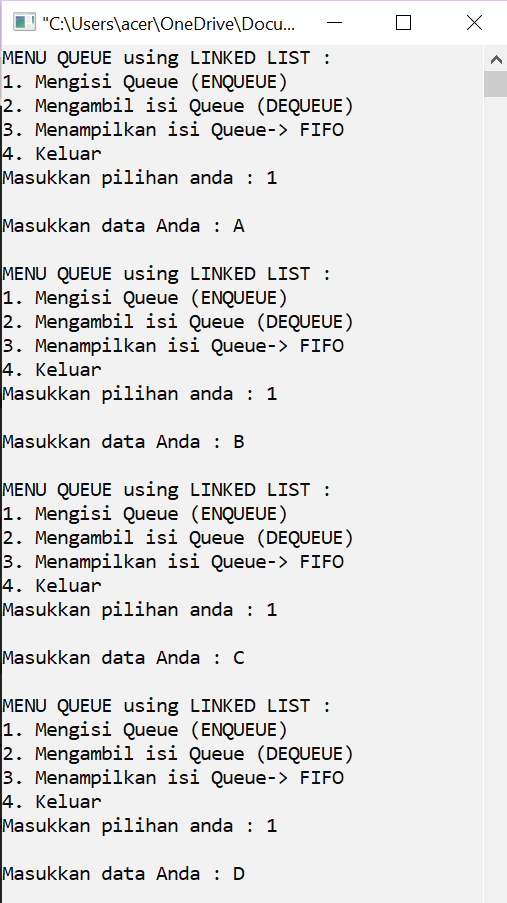
1. Output



# Menu Queue Using Linked List

1. Listing Program

|  |
| --- |
| #include <stdio.h>  #include <stdlib.h>  typedef struct simpul Node;  typedef struct{  Node \*front;  Node \*rear;  }queue;  typedef struct simpul{  char data;  Node \*next;  };  void inisialisasi(queue \*q);  int kosong(queue \*q);  int penuh(queue \*q);  void enqueue(char x, queue \*q);  char dequeue(queue \*q);  void tampil(queue \*q);  int main(){  queue antrian;  char kar;  int pil;  inisialisasi(&antrian);  do{  printf("MENU QUEUE using LINKED LIST :\n");  printf("1. Mengisi Queue (ENQUEUE)\n");  printf("2. Mengambil isi Queue (DEQUEUE)\n");  printf("3. Menampilkan isi Queue-> FIFO\n");  printf("4. Keluar\n");  printf("Masukkan pilihan anda : ");  scanf("%d",&pil);  switch(pil)  {  case 1:  fflush(stdin);  puts("");  printf("Masukkan data Anda : ");  scanf("%c", &kar);  enqueue(kar, &antrian);  fflush(stdin);  break;  case 2:  fflush(stdin);  puts("");  printf("Item data yang diambil : %c", dequeue(&antrian));  puts("");  fflush(stdin);  break;  case 3:  puts("");  tampil(&antrian);  break;  default:  exit(0);  }  puts("");  }while(pil!=4);  return 0;  }  void inisialisasi(queue \*q){  q->front = NULL;  q->rear = NULL;  }  void enqueue(char x, queue \*q){  Node \*p;  p = (Node \*)malloc(sizeof(Node));  if(p==NULL){  puts("alokasi gagal");  }else{  p->data = x;  p->next = NULL;  }  if(q->front==NULL){  q->front = p;  q->rear = p;  }else{  q->rear->next = p;  q->rear = p;  }  }  char dequeue(queue \*q){  Node \*hapus;  char ambil;  ambil = q->front->data;  hapus = q->front;  q->front = hapus->next;//hubungkan q->front dengan hapus->next jika data banyak  //jika queue data tunggal  if(q->front == NULL){  q->rear = NULL;  }  free(hapus);  hapus = NULL;  return ambil;  }  void tampil(queue \*q){  Node\* baca;  printf("Isi Queue saat ini : \n");  baca = q->front;  while(baca != NULL){  printf("%c ", baca->data);  baca = baca->next;  }  puts("");  } |

1. Output

