

# **PORTOFOLIO LAPORAN AKHIR**

Mata Kuliah Dasar sistem computer

## **Judul Proyek**

Sistem Antrian Prioritas Berbasis Assembly

## **Identitas**

- **Nama** : Muhammmad Rosyhan Anwar
- **NIM** : 2500018111
- **Kelas** : C
- **Mata Kuliah** : Dasar Sistem Komputer

## **Deskripsi Singkat Proyek**

Proyek ini merupakan aplikasi Sistem Antrian Prioritas berbasis teks yang dibuat menggunakan bahasa pemrograman Assembly dan dijalankan pada emulator EMU8086. Aplikasi dirancang untuk mensimulasikan sistem antrean dengan dua jenis layanan, yaitu Reguler dan Prioritas, di mana antrean prioritas akan dipanggil lebih dahulu.

## **Fitur Aplikasi**

1. Mengambil nomor antrean **Reguler**
2. Mengambil nomor antrean **Prioritas**
3. Memanggil antrean (mendahulukan prioritas)
4. Keluar dari aplikasi

## **Alur Program**

**START → Menu Utama → Ambil Reguler / Ambil Prioritas → Kembali ke Menu → Panggil Antrian → Prioritas → Reguler → Menu → Keluar → END**

## **Struktur Program**

### **Data Segment**

- Variabel nomor antrean reguler dan prioritas
- Pesan menu dan output teks

## **Code Segment**

- Menu utama
- Proses pengambilan antrean
- Proses pemanggilan antrean
- Keluar program

## **Source Code Inti**

```
.model small
.stack 100h

.data
menu db 13,10,"== SISTEM ANTRIAN PRIORITAS ==",13,10
        db "1. Ambil Reguler",13,10
        db "2. Ambil Prioritas",13,10
        db "3. Panggil Antrian",13,10
        db "4. Keluar",13,10
        db "Pilih: $"

msgR db 13,10,"Nomor Reguler: R$"
msgP db 13,10,"Nomor Prioritas: P$"
msgCall db 13,10,"Dipanggil: $"
msgEmpty db 13,10,"Antrian kosong.$"

regNext db 1
prioNext db 1
regCall db 1
prioCall db 1

.code
main proc
        mov ax,@data
```

```
mov ds,ax
```

```
menu_loop:
```

```
    lea dx,menu
```

```
    mov ah,09h
```

```
    int 21h
```

```
    mov ah,01h
```

```
    int 21h
```

```
    sub al,'0'
```

```
    cmp al,1
```

```
    je ambil_reg
```

```
    cmp al,2
```

```
    je ambil_prio
```

```
    cmp al,3
```

```
    je panggil
```

```
    cmp al,4
```

```
    je keluar
```

```
    jmp menu_loop
```

```
ambil_reg:
```

```
    lea dx,msgR
```

```
    mov ah,09h
```

```
    int 21h
```

```
    mov al,regNext
```

```
    add al,'0'
```

```
    mov dl,al
```

```
    mov ah,02h
```

```
    int 21h
```

```
inc regNext  
jmp menu_loop
```

ambil\_prio:

```
lea dx,msgP  
mov ah,09h  
int 21h
```

```
mov al,prioNext  
add al,'0'  
mov dl,al  
mov ah,02h  
int 21h
```

```
inc prioNext  
jmp menu_loop
```

; ===== PANGGIL ANTRIAN (FIXED) =====

panggil:

```
mov al, prioCall ; AL = nomor prioritas yang akan dipanggil  
cmp al, prioNext  
jb call_prio ; jika masih ada prioritas
```

```
mov al, regCall  
cmp al, regNext  
jb call_reg ; jika masih ada reguler
```

```
lea dx,msgEmpty  
mov ah,09h  
int 21h  
jmp menu_loop
```

```
call_prio:  
    lea dx,msgCall  
    mov ah,09h  
    int 21h
```

```
    mov dl,'P'  
    mov ah,02h  
    int 21h
```

```
    mov al,prioCall  
    add al,'0'  
    mov dl,al  
    mov ah,02h  
    int 21h
```

```
    inc prioCall  
    jmp menu_loop
```

```
call_reg:  
    lea dx,msgCall  
    mov ah,09h  
    int 21h
```

```
    mov dl,'R'  
    mov ah,02h  
    int 21h
```

```
    mov al,regCall  
    add al,'0'  
    mov dl,al  
    mov ah,02h  
    int 21h
```

```
inc regCall  
jmp menu_loop
```

keluar:

```
mov ah,4Ch  
int 21h
```

```
main endp  
end main  
.model small  
.stack 100h
```

```
.data  
menu db 13,10,"==== SISTEM ANTRIAN PRIORITAS ====",13,10  
db "1. Ambil Reguler",13,10  
db "2. Ambil Prioritas",13,10  
db "3. Panggil Antrian",13,10  
db "4. Keluar",13,10  
db "Pilih: $"
```

```
msgR db 13,10,"Nomor Reguler: R$"  
msgP db 13,10,"Nomor Prioritas: P$"  
msgCall db 13,10,"Dipanggil: $"  
msgEmpty db 13,10,"Antrian kosong.$"
```

```
regNext db 1  
prioNext db 1  
regCall db 1  
prioCall db 1
```

```
.code
```

```
main proc  
    mov ax,@data  
    mov ds,ax
```

```
menu_loop:  
    lea dx,menu  
    mov ah,09h  
    int 21h
```

```
    mov ah,01h  
    int 21h  
    sub al,'0'
```

```
    cmp al,1  
    je ambil_reg  
    cmp al,2  
    je ambil_prio  
    cmp al,3  
    je panggil  
    cmp al,4  
    je keluar  
    jmp menu_loop
```

```
ambil_reg:  
    lea dx,msgR  
    mov ah,09h  
    int 21h
```

```
    mov al,regNext  
    add al,'0'  
    mov dl,al  
    mov ah,02h
```

```
int 21h
```

```
inc regNext  
jmp menu_loop
```

```
ambil_prio:
```

```
lea dx,msgP  
mov ah,09h  
int 21h
```

```
mov al,prioNext  
add al,'0'  
mov dl,al  
mov ah,02h  
int 21h
```

```
inc prioNext  
jmp menu_loop
```

```
; ===== PANGGIL ANTRIAN (FIXED) =====
```

```
panggil:
```

```
mov al, prioCall ; AL = nomor prioritas yang akan dipanggil  
cmp al, prioNext  
jb call_prio ; jika masih ada prioritas
```

```
mov al, regCall  
cmp al, regNext  
jb call_reg ; jika masih ada reguler
```

```
lea dx,msgEmpty  
mov ah,09h  
int 21h
```

```
jmp menu_loop
```

```
call_prio:
```

```
lea dx,msgCall
```

```
mov ah,09h
```

```
int 21h
```

```
mov dl,'P'
```

```
mov ah,02h
```

```
int 21h
```

```
mov al,prioCall
```

```
add al,'0'
```

```
mov dl,al
```

```
mov ah,02h
```

```
int 21h
```

```
inc prioCall
```

```
jmp menu_loop
```

```
call_reg:
```

```
lea dx,msgCall
```

```
mov ah,09h
```

```
int 21h
```

```
mov dl,'R'
```

```
mov ah,02h
```

```
int 21h
```

```
mov al,regCall
```

```
add al,'0'
```

```
mov dl,al
```

```
mov ah,02h
```

```
int 21h
```

```
inc regCall
```

```
jmp menu_loop
```

keluar:

```
mov ah,4Ch
```

```
int 21h
```

```
main endp
```

```
end main
```

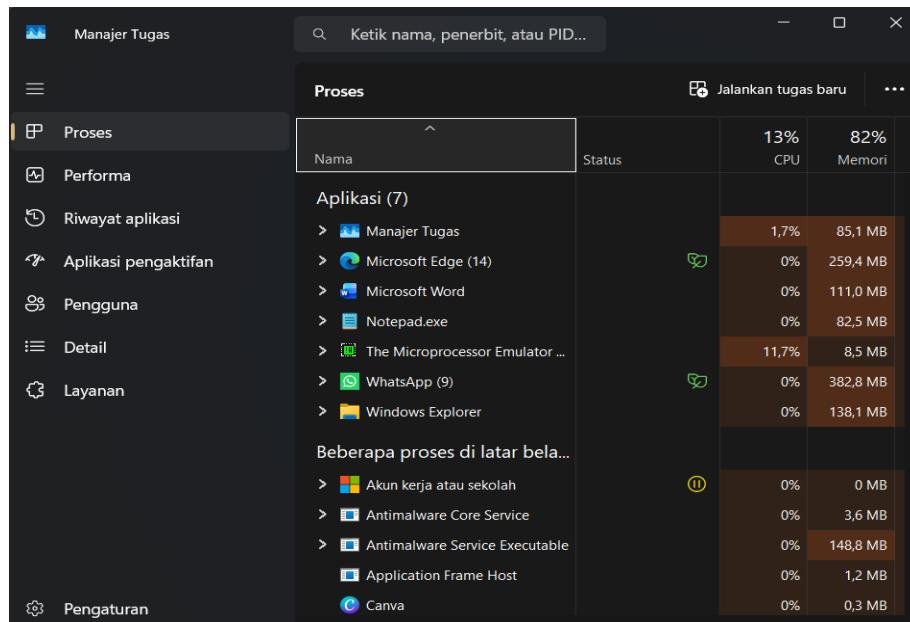
## HASIL ANALISIS KINERJA APLIKASI

### 1. Tujuan Pengujian

Pengujian dilakukan untuk mengetahui penggunaan sumber daya komputer saat aplikasi dijalankan menggunakan Task Manager dan tool monitoring sistem.

### 2. Analisis Task Manager

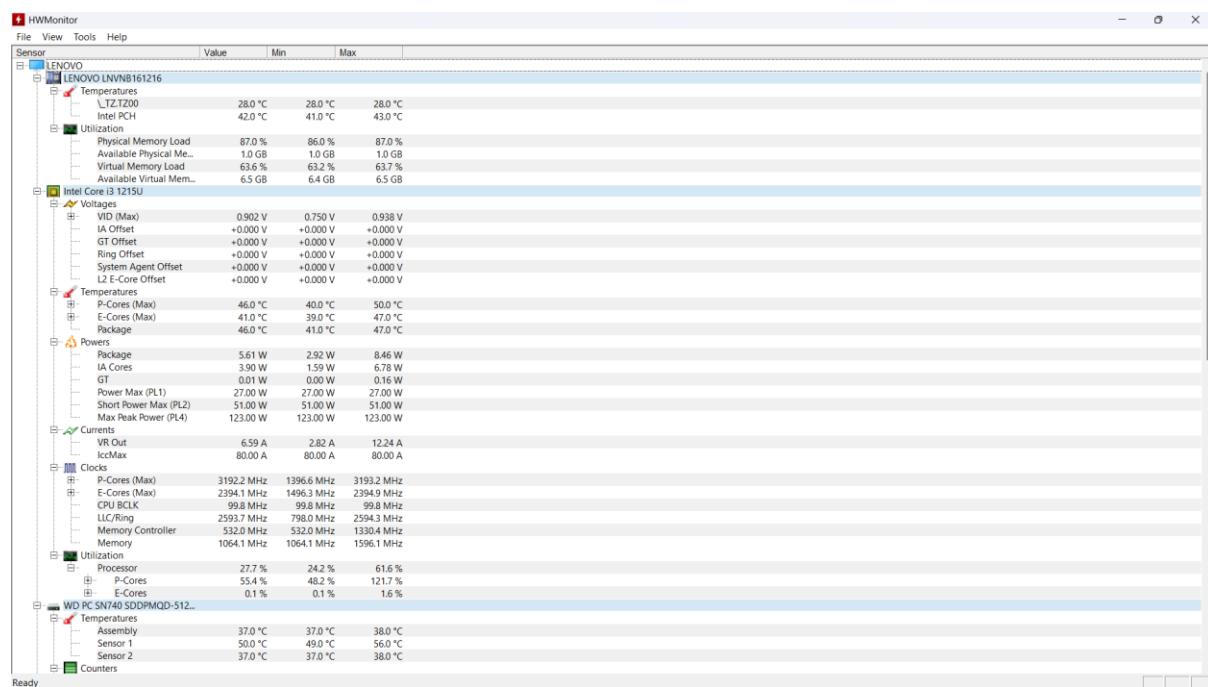
Tampilan Task Manager saat aplikasi Sistem Antrian Prioritas dijalankan menggunakan EMU 8086.



- Aplikasi dijalankan menggunakan EMU 8086 pada sistem operasi Windows.
- Penggunaan CPU sekitar 11–12% saat aplikasi aktif.
- Penggunaan memori sekitar 8,5 MB RAM.
- Tidak terjadi lonjakan penggunaan CPU maupun memori selama program berjalan.
- Aplikasi berjalan stabil tanpa mempengaruhi proses lain.
- Menunjukkan bahwa aplikasi Assembly x86 ringan dan efisien dalam penggunaan sumber daya.

### 3. Analisis Tool Monitoring

Hasil pemantauan kinerja dan suhu perangkat menggunakan aplikasi HWMonitor.



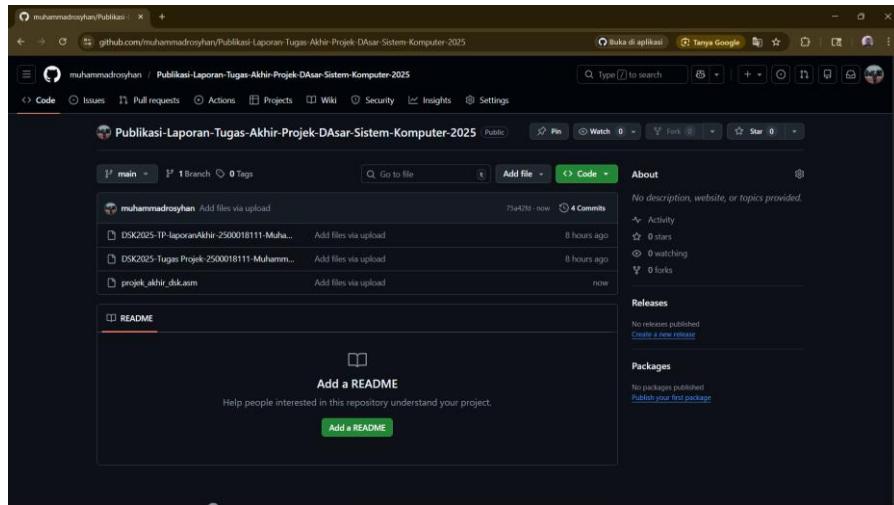
- Pengujian dilakukan saat aplikasi Sistem Antrian Prioritas berjalan di EMU 8086.
- Suhu CPU **stabil** di kisaran **41–46°C**, masih dalam batas normal.
- Daya CPU **rendah**, rata-rata sekitar **5–8 Watt**, menunjukkan beban ringan.
- Utilisasi prosesor sekitar 25–30%, tidak membebani sistem secara berlebihan.
- Memori fisik terpakai tinggi karena aplikasi lain berjalan bersamaan, namun aplikasi tetap stabil.
- Tidak terjadi **overheating** maupun lonjakan daya selama pengujian.
- Aplikasi tergolong **efisien dan aman dijalankan dalam jangka waktu lama**.

#### 4. Kesimpulan Kinerja

Aplikasi Sistem Antrian Prioritas tergolong ringan, efisien, dan stabil, serta tidak membebani sistem komputer. Sangat sesuai sebagai media pembelajaran kinerja komputer berbasis Assembly.

### UNGGAHAN GITHUB & YOUTUBE

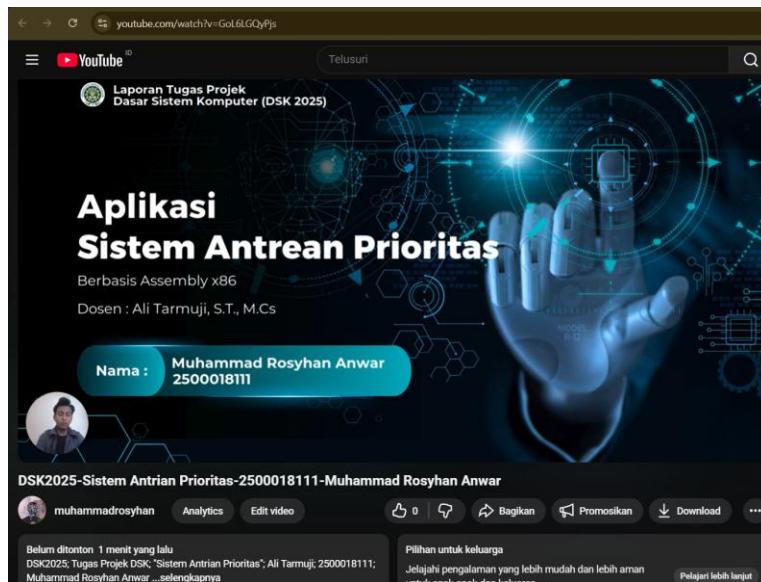
#### GitHub:



Link Github:

<https://github.com/muhammadrosyhan/Publikasi-Laporan-Tugas-Akhir-Projek-DAsar-Sistem-Komputer-2025.git>

## YouTube:



## Link Youtube:

<https://youtu.be/GoL6LGQvPjs>

## ANALISIS PENGERJAAN PROYEK

### 1. Tinjauan Waktu

Proyek diselesaikan sesuai jadwal dengan tahapan perancangan, pengkodean, dan pengujian.

### 2. Ketercapaian Spesifikasi

Seluruh fitur utama berhasil diimplementasikan dan berjalan sesuai rancangan.

### 3. Biaya

Tidak memerlukan biaya tambahan karena menggunakan software emulator dan perangkat pribadi.

### 4. Kendala

- Pemahaman logika Assembly
- Debugging alur pemanggilan antrean

### 5. Tantangan Pengembangan

Pengembangan lanjutan dapat mencakup:

- Penyimpanan data permanen
- Nomor antrean multi-digit
- Tampilan lebih interaktif

