Antrian Bank

by Kelompok 11

2 Introduction

Nama : Desy A.E.P.P.W
NIM : 23031554174

Nama : Friza Chintia P.
NIM : 23031554188

Nama : Rizka Ayu R.
NIM : 23031554224

3 Purpose

what's the purpose?

Tujuan pembuatan program simulasi antrian bank dalam proyek struktur data dan algoritma adalah memahami dan mengimplementasikan konsep antrian, menganalisis struktur data dan algoritma pengelolaannya. Project ini mengembangkan keterampilan pemrograman, menerapkan teori dalam konteks praktis, melakukan eksperimen untuk mengevaluasi kinerja sistem, dan menyusun laporan yang mencakup desain program, hasil simulasi, memberikan pengalaman dalam memecahkan masalah nyata di layanan pelanggan.

4 Struktur Data

Struktur Data yang digunakan :

QUEUE
(PRIORITY QUEUE)

```python
import heapq
import datetime

class PriorityQueue:
    def __init__(self):
        self.counter = 0  # Counter to keep track of insertion order
        self.items = []

    def enqueue(self, item, priority):
        timestamp = self.counter  # Use counter as the secondary key for maintaining order
        heapq.heappush(self.items, (priority, timestamp, self.counter, item))
        self.counter += 1

    def dequeue(self):
        return heapq.heappop(self.items)[3]

    def is_empty(self):
        return len(self.items) == 0

    def queue_view(self):
        sorted_items = sorted(self.items, key=lambda x: (x[0], x[1], x[2]))
        for i, (priority, timestamp, counter, (nama_nasabah, keperluan, time_of_enqueue)) in enumerate(sorted_items, start=1):
            print(f"{i}. {nama_nasabah} - {keperluan} (Priority: {priority}) at {time_of_enqueue}")

    def size(self):
        return len(self.items)
```

```python
# Inisialisasi prioritas
PRIORITY_HIGH = 0
PRIORITY_LOW = 1

# Inisialisasi kartu prioritas
KARTU_PRIORITAS_YES = 'ya'
KARTU_PRIORITAS_NO = 'tidak'
```

5 Algoritma

```
def search_by_name(self, name):
    results = []
    for i, (priority, (nama_nasabah, keperluan, timestamp)
        if name.lower() in nama_nasabah.lower():
            results.append((i + 1, nama_nasabah, keperluan
    return results
```

Algoritma

Algoritma yang digunakan :

SEARCH BY NAME

6 Code

```python
import heapq
import datetime
```

```python
class PriorityQueue:
    def __init__(self):
        self.counter = 0  # Counter to keep track of insertion order
        self.items = []

    def enqueue(self, item, priority):
        timestamp = self.counter  # Use counter as the secondary key for maintaining order
        heapq.heappush(self.items, (priority, timestamp, self.counter, item))
        self.counter += 1

    def dequeue(self):
        return heapq.heappop(self.items)[3]

    def is_empty(self):
        return len(self.items) == 0

    def queue_view(self):
        sorted_items = sorted(self.items, key=lambda x: (x[0], x[1], x[2]))
        for i, (priority, timestamp, counter, (nama_nasabah, keperluan, time_of_enqueue)) in enumerate(sorted_items, start=1):
            print(f"{i}. {nama_nasabah} - {keperluan} (Priority: {priority}) at {time_of_enqueue}")

    def size(self):
        return len(self.items)

    def search_by_name(self, name):
        results = []
        for i, (priority, timestamp, counter, (nama_nasabah, keperluan, time_of_enqueue)) in enumerate(self.items):
            if name.lower() in nama_nasabah.lower():
                results.append((i + 1, nama_nasabah, keperluan, priority, time_of_enqueue))
        return results
```

-Library heapq di Python digunakan untuk operasi heap queue atau priority queue, berguna untuk menyimpan elemen dengan prioritas, sehingga elemen dengan prioritas tertinggi (atau terendah) selalu dapat diakses dengan cepat.

-Datetime library dalam Python yang menyediakan kelas-kelas untuk menampilkan tanggal dan waktu.

Kode di samping mendefinisikan sebuah kelas Priority Queue yang menggunakan heap queue.

counter digunakan untuk memastikan bahwa item dengan prioritas yang sama diproses dalam urutan yang benar (urutan penyisipan). Ketika dua item memiliki prioritas yang sama, item dengan counter lebih rendah akan diambil lebih dulu karena disisipkan lebih dulu.

7 Code

```
# Deklarasi Variabel
layanan = 0
keperluan = 0
user = 0
pengerjaan = 0
jenisPegawai = 0
namaNasabah = ""

# Deklarasi pelayanan di Customer Service
keperluanCS_1 = "Pembukaan rekening bank"
keperluanCS_2 = "Pelayanan masalah pada rekening nasabah"
keperluanCS_3 = "Informasi saldo dan mutasi nasabah"
keperluanCS_4 = "Administrasi buku cek dan buku tabungan"

# Deklarasi pelayanan di Teller
keperluanTeller_1 = "Penyetoran tabungan dan deposito"
keperluanTeller_2 = "Pencatatan tabungan dan deposito"
keperluanTeller_3 = "Pencatatan transaksi buku tabungan"
```

Kode di samping mendeklarasikan beberapa variabel yang akan digunakan untuk mengelola layanan bank, termasuk jenis keperluan nasabah di Customer Service dan Teller

```
queueAntrianCS = PriorityQueue()  # Menggunakan Priority Queue
queueAntrianTeller = PriorityQueue()  # Menggunakan Priority Queue
```

Kode di samping mendeklarasikan dua antrian prioritas (priority queues) untuk mengelola antrian nasabah di bank, satu untuk Customer Service dan satu lagi untuk Teller.

8 Code

```python
# Buka User Interface Program
print("===== Aplikasi Antrian Bank =====")

while True:
    # Masukkan user
    print("Pilih jenis user : ")
    print("1. Nasabah Bank")
    print("2. Pegawai Bank")
    print("3. Keluar")
    user = int(input())

    if user == 1:
        while True:
            print()
            print("Pilih layanan : ")
            print("1. Customer Service")
            print("2. Teller")
            print("3. Cek Antrian")
            print("4. Cari Nama di Antrian")
            print("5. Kembali ke Menu")
            layanan = int(input())
```

- kode ini merupakan bagian dari sistem interaktif yang menavigasikan pengguna melalui serangkaian menu untuk memilih jenis pengguna dan layanan yang dibutuhkan.
- Menu utama memberikan pilihan untuk memilih apakah pengguna adalah nasabah bank atau pegawai bank, atau untuk keluar dari program.
- Jika pengguna adalah nasabah bank, mereka diberikan pilihan layanan yang spesifik, seperti layanan Customer Service, Teller, mengecek antrian, mencari nama di antrian, atau kembali ke menu utama.

9 Code

```python
if layanan == 1:
    while True:
        print()
        print("Pilih keperluan : ")
        print("1. " + keperluanCS_1)
        print("2. " + keperluanCS_2)
        print("3. " + keperluanCS_3)
        print("4. " + keperluanCS_4)
        print("5. Kembali")
        keperluan = int(input())

        if keperluan >= 1 and keperluan <= 4:
            print()
            namaNasabah = input("Masukkan nama Anda : ")
            print("Apakah Anda memiliki kartu prioritas? (ya/tidak): ")
            kartu_prioritas = input().strip().lower()

            priority = PRIORITY_HIGH if kartu_prioritas == KARTU_PRIORITAS_YES else PRIORITY_LOW

            timestamp = datetime.datetime.now().strftime("%Y-%m-%d , %H:%M:%S")

            if keperluan == 1:
                queueAntrianCS.enqueue((namaNasabah, keperluanCS_1, timestamp), priority)
            elif keperluan == 2:
                queueAntrianCS.enqueue((namaNasabah, keperluanCS_2, timestamp), priority)
            elif keperluan == 3:
                queueAntrianCS.enqueue((namaNasabah, keperluanCS_3, timestamp), priority)
            elif keperluan == 4:
                queueAntrianCS.enqueue((namaNasabah, keperluanCS_4, timestamp), priority)

            print()
            print("== Customer Service ==")
            print("Nomor antrian anda : ", queueAntrianCS.size())
            print("Dengan keperluan : ", end="")
            if keperluan == 1:
                print(keperluanCS_1)
            elif keperluan == 2:
                print(keperluanCS_2)
            elif keperluan == 3:
                print(keperluanCS_3)
            elif keperluan == 4:
                print(keperluanCS_4)
            break
        elif keperluan == 5:
            break
        else:
            print("Nomor tidak valid!")
```

Kode di samping adalah membuat antrian layanan CS. Pengguna diminta memilih tujuan layanan dan memasukkan nama mereka serta apakah mereka memiliki kartu prioritas. Informasi pelanggan disimpan dalam antrian, dengan pemberian prioritas kepada yang memiliki kartu prioritas. Setelah mendaftar, pengguna diberi nomor antrian dan tujuan layanan mereka. Mereka dapat keluar dari antrian kapan saja dengan memilih opsi "Kembali".

10 Code

```python
elif layanan == 2:
    while True:
        print()
        print("Pilih keperluan : ")
        print("1. " + keperluanTeller_1)
        print("2. " + keperluanTeller_2)
        print("3. " + keperluanTeller_3)
        print("4. Kembali")

        keperluan = int(input())

        if keperluan >= 1 and keperluan <= 3:
            print()
            namaNasabah = input("Masukkan nama Anda : ")
            print("Apakah Anda memiliki kartu prioritas? (ya/tidak): ")
            kartu_prioritas = input().strip().lower()

            priority = PRIORITY_HIGH if kartu_prioritas == KARTU_PRIORITAS_YES else PRIORITY_LOW
            timestamp = datetime.datetime.now().strftime("%Y-%m-%d , %H:%M:%S")

            if keperluan == 1:
                queueAntrianTeller.enqueue((namaNasabah, keperluanTeller_1, timestamp), priority)
            elif keperluan == 2:
                queueAntrianTeller.enqueue((namaNasabah, keperluanTeller_2, timestamp), priority)
            elif keperluan == 3:
                queueAntrianTeller.enqueue((namaNasabah, keperluanTeller_3, timestamp), priority)

            print("== Teller ==")
            print("Nomor antrian anda : ", queueAntrianTeller.size())
            print("Dengan keperluan : ", end="")
            if keperluan == 1:
                print(keperluanTeller_1)
            elif keperluan == 2:
                print(keperluanTeller_2)
            elif keperluan == 3:
                print(keperluanTeller_3)
            break
        elif keperluan == 4:
            break
        else:
            print("Nomor tidak valid!")
```

Kode di samping adalah membuat antrian layanan Teller. Pengguna diminta memilih tujuan layanan dan memasukkan nama mereka serta apakah mereka memiliki kartu prioritas. Informasi pelanggan disimpan dalam antrian, dengan pemberian prioritas kepada yang memiliki kartu prioritas. Setelah mendaftar, pengguna diberi nomor antrian dan tujuan layanan mereka. Mereka dapat keluar dari antrian kapan saja dengan memilih opsi "Kembali".

11 Code

```python
elif layanan == 3:
    print()
    print("===== ANTRIAN CUSTOMER SERVICE =====")
    queueAntrianCS.queue_view()
    print("=====      ANTRIAN TELLER      =====")
    queueAntrianTeller.queue_view()

elif layanan == 4:
    print()
    print("Masukkan nama nasabah yang ingin dicari: ")
    nama_cari = input().strip()

    print("===== HASIL PENCARIAN CUSTOMER SERVICE =====")
    hasil_cs = queueAntrianCS.search_by_name(nama_cari)
    if hasil_cs:
        for result in hasil_cs:
            print(f"{result[0]}. {result[1]} - {result[2]} (Priority: {result[3]}) at {result[4]}")
    else:
        print("Nama tidak ditemukan di antrian Customer Service.")

    print("===== HASIL PENCARIAN TELLER =====")
    hasil_teller = queueAntrianTeller.search_by_name(nama_cari)
    if hasil_teller:
        for result in hasil_teller:
            print(f"{result[0]}. {result[1]} - {result[2]} (Priority: {result[3]}) at {result[4]}")
    else:
        print("Nama tidak ditemukan di antrian Teller.")

elif layanan == 5:
    break
else:
    print("Nomor tidak valid!")
```

- Jika **layanan** sama dengan 3, maka program akan menampilkan tampilan antrian untuk layanan pelanggan dan layanan teller.
- Jika **layanan** sama dengan 4, program meminta pengguna untuk memasukkan nama nasabah yang ingin dicari. Kemudian, program mencari nama tersebut di antrian layanan pelanggan dan layanan teller, dan menampilkan hasil pencarian tersebut.
- Jika **layanan** sama dengan 5, pengguna dapat keluar dari program.
- Jika input tidak valid, program akan memberikan pesan kesalahan.

12 Code

```python
elif user == 2:
    while True:
        print()
        print("===== ANTRIAN CUSTOMER SERVICE =====")
        queueAntrianCS.queue_view()
        print("=====      ANTRIAN TELLER      =====")
        queueAntrianTeller.queue_view()

        print()
        print("Pilh jenis pegawai : ")
        print("1. Customer Service")
        print("2. Teller")
        print("3. Kembali ke Menu Utama")

        jenisPegawai = int(input())

        if jenisPegawai == 1:
            while True:
                print()
                print("Pilih jenis pengerjaan : ")
                print("1. Layani")
                print("2. Hapus")
                print("3. Kembali ke menu pegawai")

                pengerjaan = int(input())

                if pengerjaan == 1 or pengerjaan == 2:
                    queueAntrianCS.dequeue()
                    print()
                elif pengerjaan == 3:
                    break
                else:
                    print("Nomor tidak valid!")
```

- Pertama, program menampilkan pengguna untuk memilih jenis pegawai: Customer Service atau Teller, atau kembali ke Menu Utama.
- Jika pengguna memilih Customer Service (jenisPegawai == 1), program akan memungkinkan pengguna untuk memilih jenis pengerjaan: Layani atau Hapus. Jika pengguna memilih untuk melayani (pengerjaan == 1) atau menghapus (pengerjaan == 2), program akan menghapus pelanggan pertama dari antrian layanan pelanggan.
- Jika pengguna memilih untuk kembali ke menu pegawai (pengerjaan == 3), mereka akan kembali ke menu sebelumnya.
- Jika input tidak valid, program akan memberikan pesan kesalahan.

13 Code

```python
elif jenisPegawai == 2:
    while True:
        print()
        print("Pilih jenis pengerjaan : ")
        print("1. Layani")
        print("2. Hapus")
        print("3. Kembali ke menu pegawai")

        pengerjaan = int(input())

        if pengerjaan == 1 or pengerjaan == 2:
            queueAntrianTeller.dequeue()
            print()
            print("=====      ANTRIAN TELLER      =====")
            queueAntrianTeller.queue_view()
            break
        elif pengerjaan == 3:
            break
        else:
            print("Nomor tidak valid!")


elif jenisPegawai == 3:
    break
else:
    print("Nomor tidak valid!")
```

Kode di samping adalah bagian dari program antrian untuk pegawai dengan `jenisPegawai` == 2 (Teller). Dalam loop, program menampilkan menu pilihan: "Layani", "Hapus", atau "Kembali ke menu pegawai". Jika pengguna memilih opsi 1 atau 2, metode `dequeue()` dipanggil untuk mengeluarkan elemen dari `queueAntrianTeller`, kemudian menampilkan antrian terbaru dengan `queue_view()`, dan keluar dari loop. Pilihan opsi 3 hanya keluar dari loop. Jika input tidak valid, program akan menampilkan pesan kesalahan.

14 Code

```
if user == 3:
    break
else:
    print("Nomor tidak valid!")
```

Berdasarkan kode disamping, jika memilih user == 3 maka program akan berhenti dan keluar dari program, jika input > 3 maka output yang ditampilkan adalah 'Nomor tidak valid'

15 Code

```
queueAntrianCS = PriorityQueue()  # Menggunakan Priority Queue
queueAntrianTeller = PriorityQueue()  # Menggunakan Priority Queue
```

```
    print("===== ANTRIAN CUSTOMER SERVICE =====")
    queueAntrianCS.queue_view()
    print("=====      ANTRIAN TELLER      =====")
    queueAntrianTeller.queue_view()
```

Contoh output

```
===== ANTRIAN CUSTOMER SERVICE =====
1. dini - Pembukaan rekening bank (Priority: 0) at 202
4-05-28 , 10:10:09
2. friza - Pembukaan rekening bank (Priority: 1) at 20
24-05-28 , 10:09:24
=====      ANTRIAN TELLER      =====
1. lala - Pencatatan transaksi buku tabungan (Priority
: 0) at 2024-05-28 , 10:10:26
2. chintia - Pencatatan tabungan dan deposito (Priorit
y: 1) at 2024-05-28 , 10:09:51
```

Kode di samping mendeklarasikan dua antrian prioritas (priority queues) untuk mengelola antrian nasabah di bank, satu untuk Customer Service dan satu lagi untuk Teller.

Setiap kali nasabah mendaftar untuk layanan, data mereka (nama, keperluan, prioritas, dan datetime) dimasukkan ke dalam antrian yang sesuai. Nasabah dapat melihat antriannya, serta pegawai bank dapat melihat antrian dan melayani nasabah sesuai dengan prioritasnya.

Contribution

Desy Amelia Eka Putri P.W

1 Membuat Class Priority Queue (Enqueue, Dequeue, Is_empty, queue_view, size) dan Searching

Friza Chintia Putri

2 Membuat Code bagian User 2, yaitu Pegawai Bank

Rizka Ayu Ramadhani

3 Membuat Code bagian User 1, yaitu Nasabah Bank

17 Closing

TERIMA KASIH ATAS PERHATIANNYA