

## **Anggota Kelompok**

1. Hyun Jung (00000088991)
2. Jeremy Marvel (00000088987)
3. Yua Pedroza (00000089118)
4. Marcello Zefanya Albertz Zega (00000089382)

## **Pembagian Tugas**

### **1. Jeremy Marvel**

Menganalisa website yang digunakan sebagai resource, menentukan fitur-fitur yang diimplementasikan, menentukan tipe-tipe data dari resource. Menyusun dan merancang program, menyusun fungsi mengurutkan hotel berdasarkan tipe, urutkan hotel berdasarkan harga.

### **2. Marcello Zefanya Albertz Zega**

Menganalisa website yang digunakan sebagai resource, menyusun dan merancang program, menyusun source code halaman utama, menyusun fungsi-fungsi yang ada pada lihat hotel, Menyusun fungsi lihat hotel berdasarkan bintang.

### **3. Hyun Jung**

Menganalisa website yang digunakan sebagai resource, menyusun dan merancang program, menyusun fungsi lihat tiket, menyusun fungsi mengurutkan hotel berdasarkan harga, menyusun fungsi batalkan pesan.

### **4. Yua Pedroza**

Menganalisa website yang digunakan sebagai resource, menyusun dan merancang program, menyusun fungsi push, menyusun fungsi pesan tiket, menyusun fungsi urutkan hotel berdasarkan rating, dan menyusun struct pada source code.

## **I. Observasi**

Kelompok kami memilih Traveloka (online travel agent) sebagai studi kasus. Berikut adalah beberapa lampiran hasil observasi kami :

1. Proses bisnis yang ingin kami implementasikan adalah pemesanan hotel. Pada laman ini, terdapat beberapa elemen. Elemen pertama, adalah lokasi/daerah hotel yang dipilih. Elemen kedua adalah tanggal check in, dan durasi menginap. Elemen ketiga adalah jumlah tamu dan kamar yang ingin dipesan.

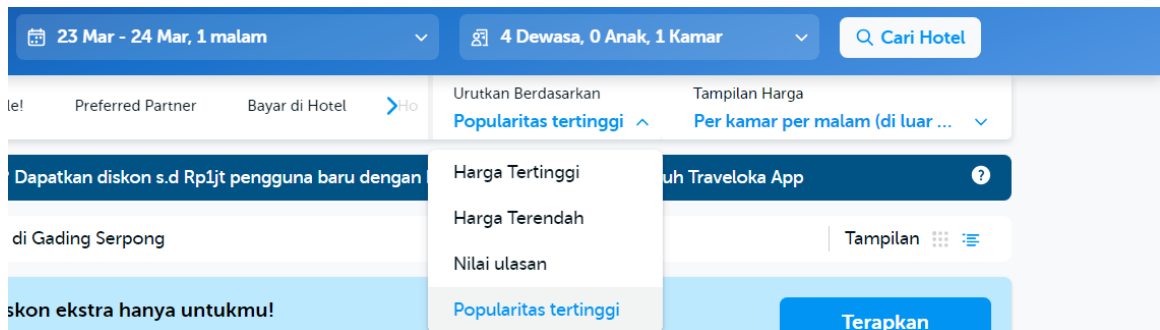
The screenshot shows the Traveloka website's search interface. At the top, there's a navigation bar with links like 'Hotel', 'Tiket Pesawat', 'Tiket Kereta Api', etc. Below this, a large blue banner features promotional text and images. The main search area is a white box titled 'Hotel yang Terakhir Dilihat'. It contains a search bar with 'Gading Serpong' entered. Below the search bar, there are fields for 'Check-in' (Sab, 23 Mar 2024), 'Durasi' (1 malam), and 'Check-out' (Min, 24 Mar 2024). A dropdown menu shows '4 Dewasa, 0 Anak, 1 Kamar'. There are buttons for 'Cari Hotel' and 'Bayar di Hotel'. A 'Perlu Bantuan?' button is at the bottom right.

Pada bagian ini, data yang akan kami gunakan adalah daerah/lokasi hotel yang bertipe data character (char), tanggal check in dan durasi menginap bertipe data integer (int). Pada program yang akan kami buat, pemilihan lokasi/daerah hotel disederhanakan hanya di daerah Gading Serpong.

2. Pada laman pencarian hotel, laman tersebut akan menampilkan data-data hotel yang ada. Data yang ditampilkan adalah nama hotel, jenis penginapan, bintang hotel, rating, dan harga hotel.

The screenshot shows the Traveloka search results page for hotels in Gading Serpong. The page has a blue header with the Traveloka logo and navigation links. Below the header, there's a search bar with 'Gading Serpong (578 properties)' and filters for '23 Mar - 24 Mar, 1 malam' and '4 Dewasa, 0 Anak, 1 Kamar'. The main content area displays a list of hotel results. Each result includes a hotel name, a rating, a price, and a 'Pilih Kamar' button. On the left side, there are filters for 'Ramadan Sale!', 'Hotel', 'Kamar Renang', 'Kisaran Harga', 'Kebijakan Pesanan', and 'Bintang'. A 'Perlu Bantuan?' button is at the bottom right.

Terdapat beberapa fitur untuk membantu mempermudah pencarian. Fitur tersebut adalah mengurutkan hotel berdasarkan harga tertinggi, harga terendah, nilai ulasan, dan popularitas tertinggi.



Fitur mengurutkan hotel berdasarkan harga tertinggi akan menampilkan data hotel berdasarkan harga tertinggi ke terendah. Fitur mengurutkan harga terendah ke tertinggi akan mengurutkan data hotel dari harga terendah ke harga tertinggi. Fitur mengurutkan hotel berdasarkan nilai ulasan akan mengurutkan data hotel dari ulasan tertinggi ke terendah.

Pada bagian ini, data yang akan kami gunakan adalah nama hotel yang bertipe data character (char), harga hotel, nilai ulasan, dan rating yang memiliki tipe data float. Kami juga akan mengimplementasikan fitur mengurutkan data hotel berdasarkan bintang, rating, tipe penginapan, dan harga hotel.

Kelompok kami mengembangkan aplikasi untuk memesan hotel di daerah Gading Serpong. Aplikasi ini akan memiliki lima menu utama, yaitu pesan tiket, lihat tiket, batalkan pesanan, lihat hotel, dan keluar/berhenti.

Pada menu pertama, yaitu pesan tiket, pengguna akan diminta untuk memasukkan tanggal check-in, durasi menginap, jumlah tamu, dan jumlah kamar. Setelah itu, akan ditampilkan daftar hotel yang tersedia, dimana pengguna diminta untuk memilih hotel sesuai dengan nomor yang ditampilkan.

Menu kedua, yaitu tampilkan tiket akan menampilkan tiket yang sudah dipesan oleh pengguna. Menu ketiga, yaitu menu batalkan pesanan, pengguna dapat membatalkan pesanan yang sudah dibuat sebelumnya. Menu keempat, yaitu lihat hotel, pengguna dapat melihat daftar hotel berdasarkan harga tertinggi, harga terendah, rating tertinggi, rating terendah, bintang hotel, dan tipe penginapan. Menu terakhir, menu keluar/berhenti digunakan untuk keluar dari program.

## II. File Processing

- **File processing “Read”**

Berikut adalah source code yang mengimplementasikan file processing dengan mode “r” atau read.

```
void lihatTiket() {
    FILE *file = fopen(FILE_PESANAN, "r");
    if (file == NULL) {
        printf("Gagal membuka file pesanan.\n");
        return;
    }

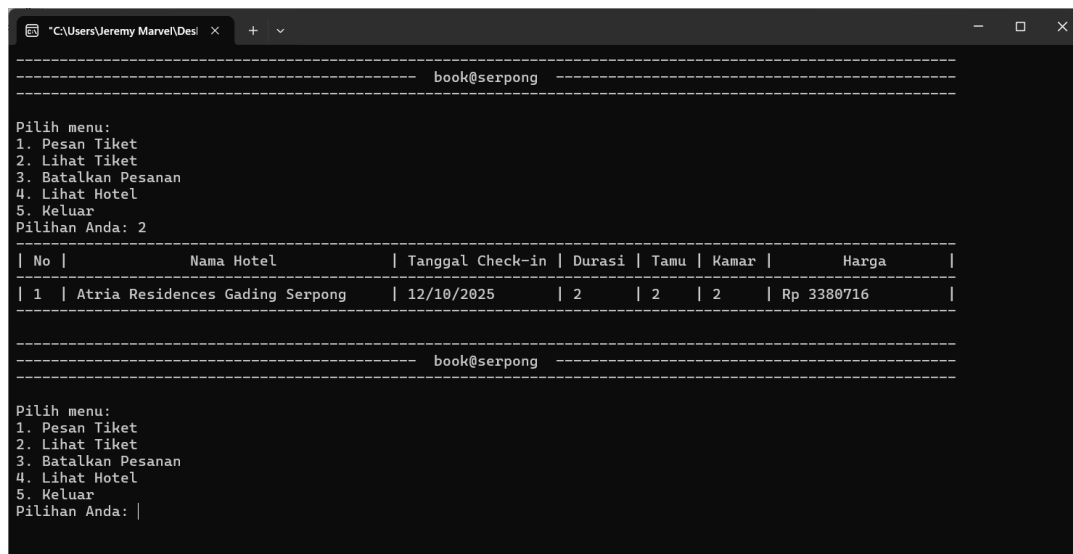
    printf("-----\n");
    printf("| No | Nama Hotel | Tanggal Check-in | Durasi | Tamu | Kamar | Harga | \n");
    printf("-----\n");

    int nomor_pesanan = 1;
    struct Tiket pesanan;
    while (fscanf(file, "%[^#]#%[^#]#%d#%d#%d#%s\n", pesanan.hotel.nama_hotel, pesanan.tanggal_checkin,
        &pesanan.durasi, &pesanan.jumlah_tamu, &pesanan.jumlah_kamar, pesanan.hotel.harga) != EOF) {
        // Calculate the modified price
        long int hargaNumerik = konversiHarga(pesanan.hotel.harga);
        hargaNumerik *= pesanan.jumlah_kamar * pesanan.durasi;

        printf("| %-2d | %-35s | %-16s | %-6d | %-4d | %-5d | Rp %-15ld | \n", nomor_pesanan, pesanan.hotel.nama_hotel, pesanan.tanggal_checkin,
            pesanan.durasi, pesanan.jumlah_tamu, pesanan.jumlah_kamar, hargaNumerik);
        nomor_pesanan++;
    }

    printf("-----\n\n");
    fclose(file);
}
```

Tampilan setelah source code dijalankan.



```
book@serpong

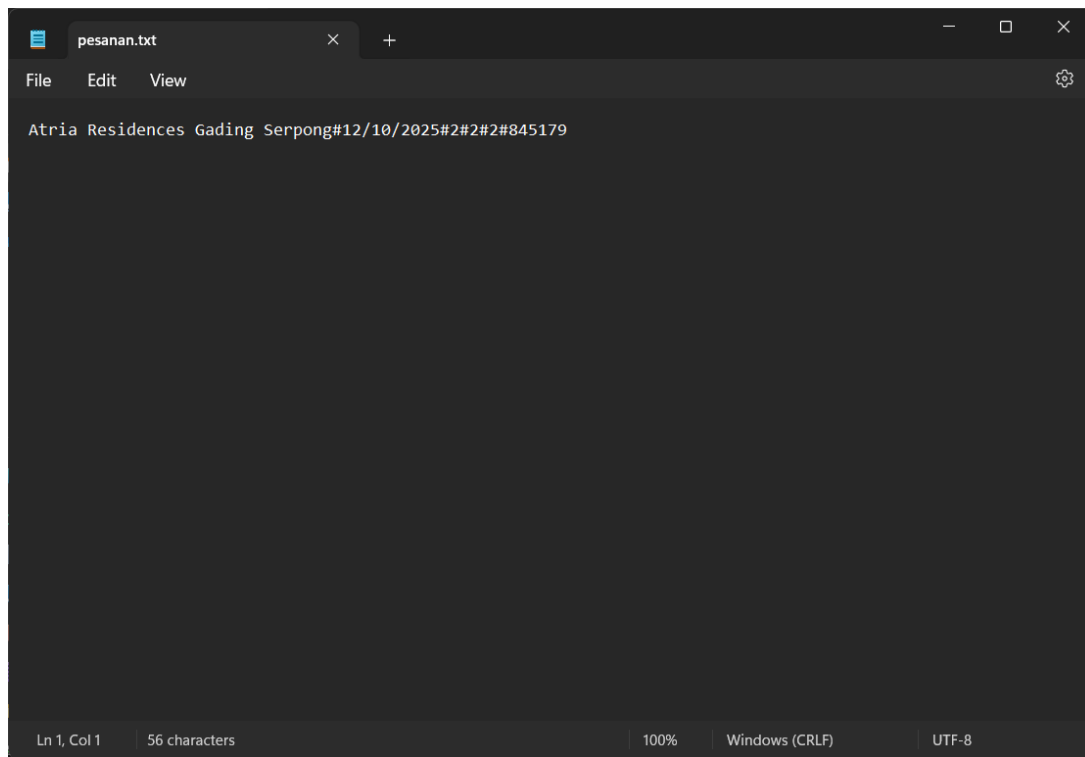
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 2

-----
| No | Nama Hotel | Tanggal Check-in | Durasi | Tamu | Kamar | Harga |
-----
| 1 | Atria Residences Gading Serpong | 12/10/2025 | 2 | 2 | 2 | Rp 3380716 |
-----

book@serpong

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: |
```

Isi file “pesanan.txt” yang dibaca oleh program.



A screenshot of a text editor window with a dark theme. The title bar shows a single tab labeled "pesanan.txt" with a close button (X) and a plus sign (+) for new tabs. The menu bar includes "File", "Edit", and "View", with a settings gear icon on the right. The main text area contains the string "Atria Residences Gading Serpong#12/10/2025#2#2#845179". The status bar at the bottom displays "Ln 1, Col 1", "56 characters", "100%", "Windows (CRLF)", and "UTF-8".

```
Atria Residences Gading Serpong#12/10/2025#2#2#845179
```

- **File processing “Write”**

Berikut adalah source code yang mengimplementasikan file processing dengan mode “w” atau write.

```
void batalkanPesan() {
    FILE *file = fopen(FILE_PESANAN, "r");
    if (file == NULL) {
        printf("Gagal membuka file pesanan.\n");
        return;
    }

    struct Tiket pesanan[MAX_TIKET];
    int jumlah_pesanan = 0;

    while (fscanf(file, "%[^#]#[^#]#%d#%d#%d#%s\n", pesanan[jumlah_pesanan].hotel.nama_hotel,
        pesanan[jumlah_pesanan].tanggal_checkin, &pesanan[jumlah_pesanan].durasi,
        &pesanan[jumlah_pesanan].jumlah_tamu, &pesanan[jumlah_pesanan].jumlah_kamar,
        pesanan[jumlah_pesanan].hotel.harga) != EOF) {
        jumlah_pesanan++;
    }
    fclose(file);

    if (jumlah_pesanan > 0) {
        printf("Pesanan terakhir akan dibatalkan.\n");
        jumlah_pesanan--;
        file = fopen(FILE_PESANAN, "w");
        if (file == NULL) {
            printf("Gagal membuka file pesanan.\n");
            return;
        }
        for (int i = 0; i < jumlah_pesanan; i++) {
            fprintf(file, "%s#%s#%d#%d#%d#%s\n", pesanan[i].hotel.nama_hotel, pesanan[i].tanggal_checkin,
                pesanan[i].durasi, pesanan[i].jumlah_tamu, pesanan[i].jumlah_kamar,
                pesanan[i].hotel.harga);
        }
        fclose(file);

        printf("Pesanan terakhir berhasil dibatalkan.\n");
    } else {
        printf("Tidak ada pesanan yang bisa dibatalkan.\n");
    }
}
```

Tampilan setelah source code dijalankan.

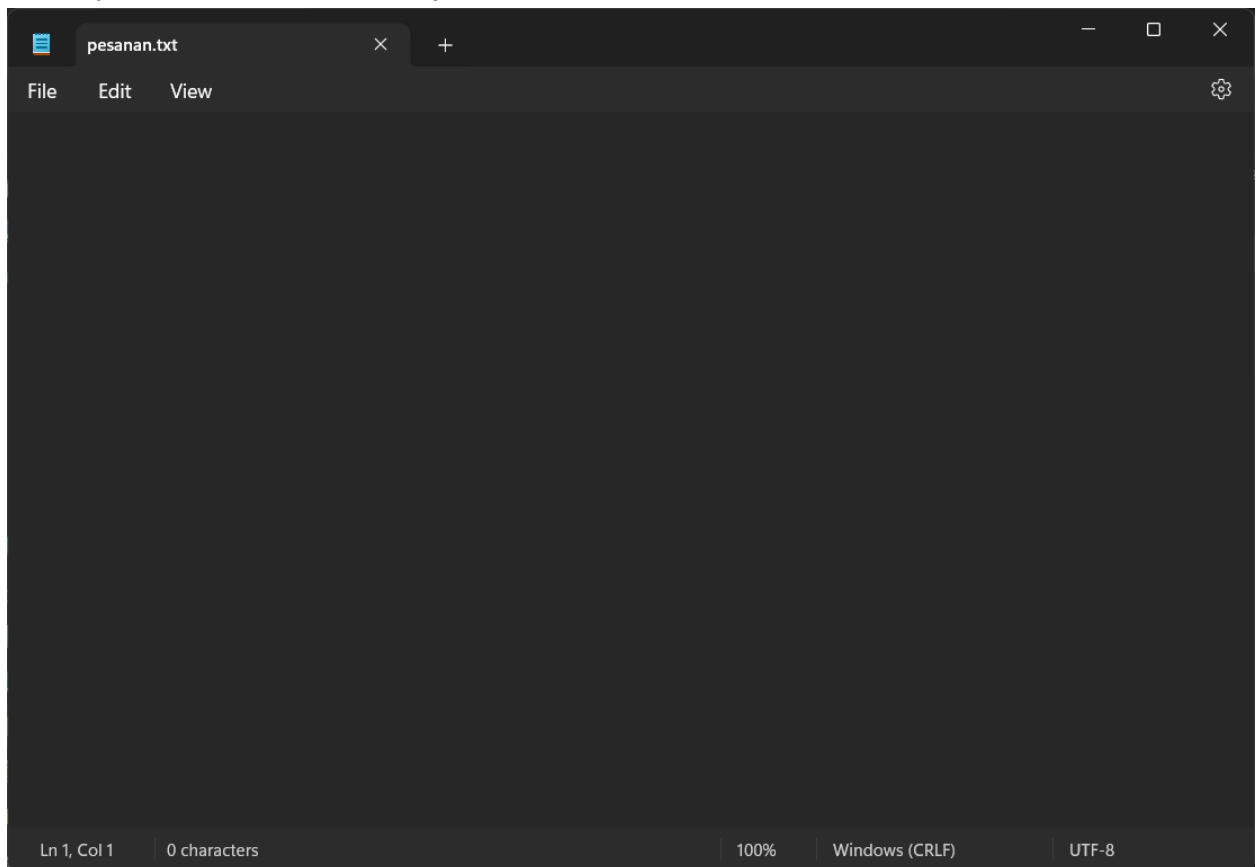
```
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 3
Pesanan terakhir akan dibatalkan.
Pesanan terakhir berhasil dibatalkan.
----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 2

| No | Nama Hotel | Tanggal Check-in | Durasi | Tamu | Kamar | Harga |
-----
----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: |
```

Isi file “pesanan.txt” setelah code dijalankan.



- **File processing “Append”**

Berikut adalah source code yang mengimplementasikan ‘append’

```
void push(struct Tiket tiket) {
    struct TiketNode *newNode = (struct TiketNode *)malloc(sizeof(struct TiketNode));
    if (newNode == NULL) {
        printf("Memori penuh, tidak dapat menambahkan pemesanan tiket.\n");
        return;
    }

    newNode->tiket = tiket;
    newNode->next = top;
    top = newNode;

    FILE *file = fopen(FILE_PESANAN, "a");
    if (file == NULL) {
        printf("Gagal membuka file pesanan.\n");
        return;
    }
    long int hargaNumerik = konversiHarga(tiket.hotel.harga);
    fprintf(file, "%s#%s#%d#%d#%d#%ld\n", tiket.hotel.nama_hotel, tiket.tanggal_checkin, tiket.durasi,
            tiket.jumlah_tamu, tiket.jumlah_kamar, hargaNumerik);
    fclose(file);
}
```

Tampilan setelah source code dijalankan

```
C:\Users\Jeremy Marvel\Down x + v

----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 1
Masukkan tanggal check-in (dd/mm/yyyy): 22/04/2024
Masukkan durasi menginap (hari): 4
Masukkan jumlah tamu (1-4 orang): 4
Masukkan jumlah kamar: 4

Daftar Hotel:
No  Nama                               Bintang  Rating  Jenis  Harga
1   Episode Gading Serpong             4        8.8     Hotel  Rp 1.454.545
2   JHL Solitaire Gading Serpong        5        8.7     Hotel  Rp 2.242.680
3   Hotel Gading Serpong                2        8.2     Hotel  Rp 343.802
4   Atria Residences Gading Serpong     4        8.7     Hotel  Rp 845.179
5   D' Rooms Studio & 2BR              0        9.0     Apartemen  Rp 429.752
6   Avilla Residence                   2        8.0     Guest House  Rp 413.223
7   Lemo Hotel Serpong                 3        8.4     Hotel  Rp 409.000
8   VIVERE Hotel                       4        8.8     Hotel  Rp 1.015.666
9   Ibis Gading Serpong                3        8.5     Hotel  Rp 643.720
10  Starlet Hotel Serpong               1        8.3     Hotel  Rp 288.000

Silakan pilih hotel dari daftar di atas (masukkan nomor hotel):
Masukkan nomor hotel: 5

Tiket berhasil dipesan.

----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 2

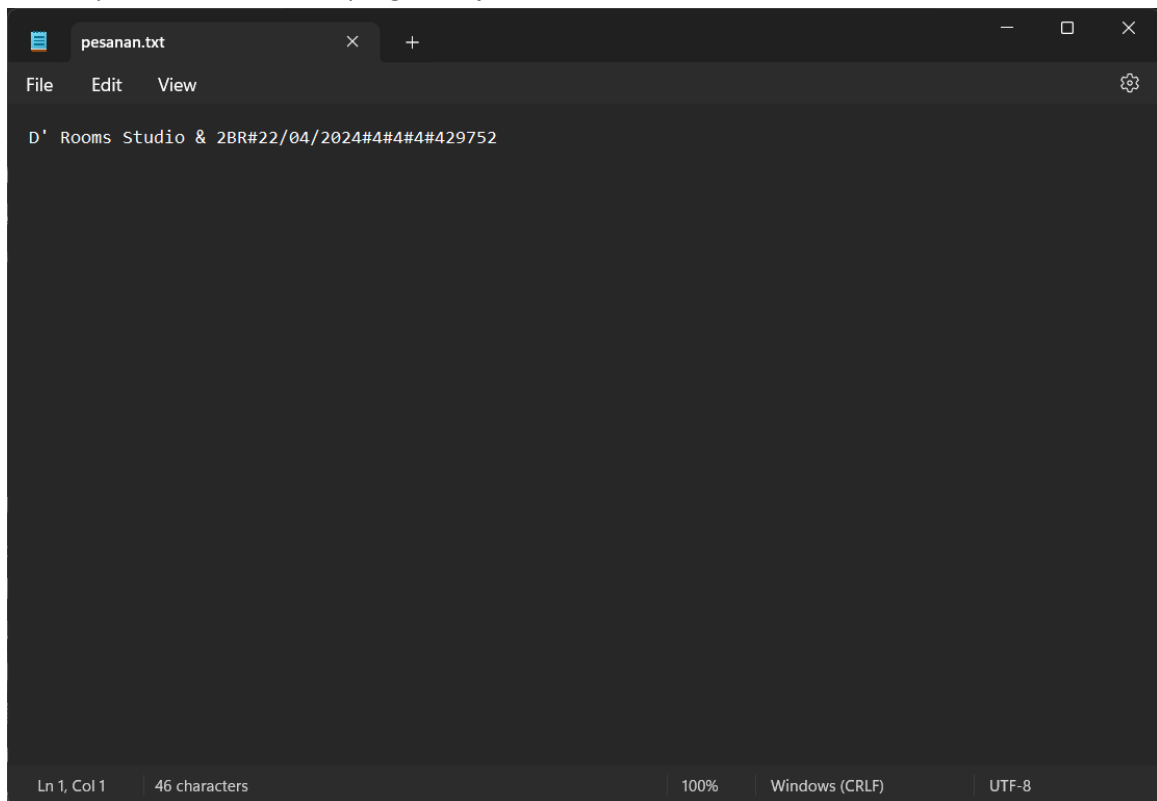
| No | Nama Hotel | Tanggal Check-in | Durasi | Tamu | Kamar | Harga |
| 1 | D' Rooms Studio & 2BR | 22/04/2024 | 4 | 4 | 4 | Rp 6876032 |

----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: |
```



Isi file “pesanan.txt” setelah program dijalankan.

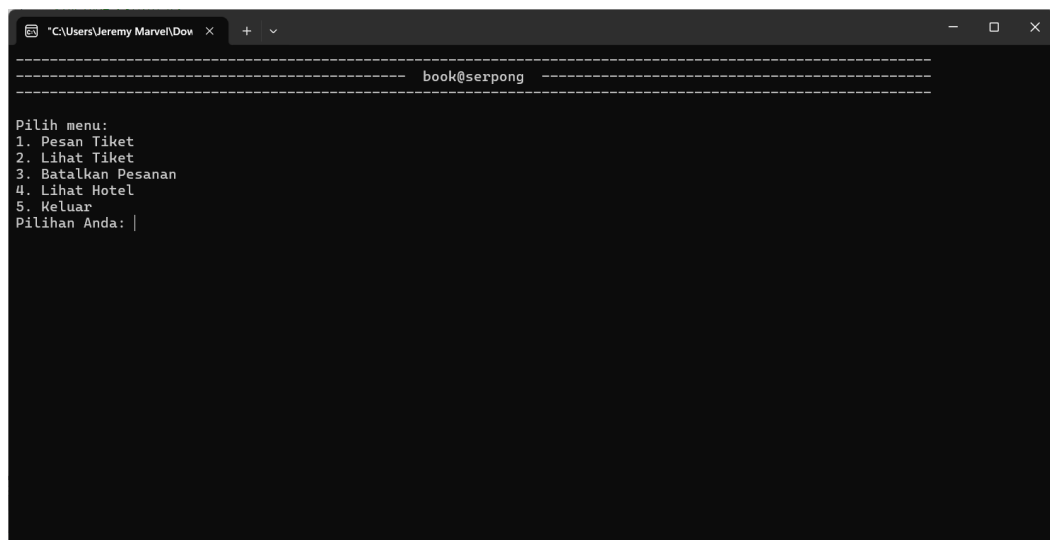


A screenshot of a text editor window titled "pesanan.txt". The window has a dark theme and a menu bar with "File", "Edit", and "View". The main text area contains the string "D' Rooms Studio & 2BR#22/04/2024#4#4#429752". The status bar at the bottom shows "Ln 1, Col 1", "46 characters", "100%", "Windows (CRLF)", and "UTF-8".

```
D' Rooms Studio & 2BR#22/04/2024#4#4#429752
```

### III . 10 Fitur terkait topik/sistem menggunakan array, stack, queue dan linked-list.

#### 1. Menu utama dan pesan tiket



A screenshot of a terminal window with a dark background. The window title is "C:\Users\Jeremy Marvel\Down". The prompt is "book@serpong". The menu text is as follows:

```
----- book@serpong -----
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: |
```

Source code function pesanTiket(), yang akan dijalankan ketika memilih opsi 1.

```
void pesanTiket() {
    int jumlah_hotel = 0;

    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];
    while (fscanf(file, "%[^%]#%c%f#%c#%s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        jumlah_hotel++;
    }
    fclose(file);

    printf("Masukkan tanggal check-in (dd/mm/yyyy): ");
    char tanggal_checkin[20];
    scanf("%s", tanggal_checkin);

    printf("Masukkan durasi menginap (hari): ");
    int durasi;
    scanf("%d", &durasi);

    printf("Masukkan jumlah tamu (1-4 orang): ");
    int jumlah_tamu;
    scanf("%d", &jumlah_tamu);

    printf("Masukkan jumlah kamar: ");
    int jumlah_kamar;
    scanf("%d", &jumlah_kamar);

    // Menampilkan daftar hotel
    file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }
    printf("\nDaftar Hotel:\n");
    printf("%s-%s %s-%s %s-%s %s-%s %s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    int nomor_hotel = 1;
    while (fscanf(file, "%[^%]#%c%f#%c#%s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        printf("%s-%s %s-%s %s-%s %s-%s %s\n", nomor_hotel, nama_hotel, bintang_hotel, rating,
            jenis_penginapan == 'H' ? "Hotel" : (jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), harga);
        nomor_hotel++;
    }
    fclose(file);

    printf("\nSilakan pilih hotel dari daftar di atas (masukkan nomor hotel):\n");
    printf("\nMasukkan nomor hotel: ");
    int nomor_terpilih;
    scanf("%d", &nomor_terpilih);

    if (nomor_terpilih < 1 || nomor_terpilih > jumlah_hotel) {
        printf("Nomor hotel tidak valid.\n");
        return;
    }

    file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }
    for (int i = 1; i <= nomor_terpilih; i++) {
        fscanf(file, "%[^%]#%c%f#%c#%s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga);
    }
    fclose(file);

    struct Tiket tiket;
    strcpy(tiket.hotel.nama_hotel, nama_hotel);
    tiket.hotel.bintang_hotel = bintang_hotel;
    tiket.hotel.rating = rating;
    tiket.hotel.jenis_penginapan = jenis_penginapan;
    strcpy(tiket.hotel.harga, harga);
    strcpy(tiket.tanggal_checkin, tanggal_checkin);
    tiket.durasi = durasi;
    tiket.jumlah_tamu = jumlah_tamu;
    tiket.jumlah_kamar = jumlah_kamar;

    push(tiket); // Menambahkan pemesanan tiket ke dalam tumpukan
    printf("\nTiket berhasil dipesan.\n");
}
```

Tampilan program ketika function pesanTiket() dijalankan

```
"C:\Users\Jeremy MarvohDow" x + -
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 1
Masukkan tanggal check-in (dd/mm/yyyy): 22/04/2024
Masukkan durasi menginap (hari): 22
Masukkan jumlah tamu (1-4 orang): 3
Masukkan jumlah kamar: 3

Daftar Hotel:
No    Nama                               Bintang  Rating  Jenis    Harga
1     Episode Gading Serpong             4        8.8     Hotel    Rp 1.454.545
2     JHL Solitaire Gading Serpong        5        8.7     Hotel    Rp 2.242.680
3     Hotel Gading Serpong                2        8.2     Hotel    Rp 343.802
4     Atria Residences Gading Serpong     4        8.7     Hotel    Rp 845.179
5     D' Rooms Studio & 2BR               0        9.0     Apartemen Rp 429.752
6     Avilla Residence                   2        8.0     Guest House Rp 413.223
7     Lemo Hotel Serpong                 3        8.4     Hotel    Rp 409.000
8     VIVERE Hotel                       4        8.8     Hotel    Rp 1.015.666
9     Ibis Gading Serpong                3        8.5     Hotel    Rp 643.720
10    Starlet Hotel Serpong               1        8.3     Hotel    Rp 288.000

Silakan pilih hotel dari daftar di atas (masukkan nomor hotel):

Masukkan nomor hotel: 5

Tiket berhasil dipesan.
-----
```

## 2. Lihat tiket

```
----- book@serpong -----
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 2|
```

Source code lihat tiket, yang akan dijalankan ketika memilih opsi menu 2.

```
void lihatTiket() {
    FILE *file = fopen(FILE_PESANAN, "r");
    if (file == NULL) {
        printf("Gagal membuka file pesanan.\n");
        return;
    }

    printf("-----\n");
    printf("| No |          Nama Hotel          | Tanggal Check-in | Durasi | Tamu | Kamar |          Harga          |\n");
    printf("-----\n");

    int nomor_pesanan = 1;
    struct Tiket pesanan;
    while (fscanf(file, "%[^#]#[^#]#%d#%d#%d#%s\n", &pesanan.hotel.nama_hotel, &pesanan.tanggal_checkin,
        &pesanan.durasi, &pesanan.jumlah_tamu, &pesanan.jumlah_kamar, &pesanan.hotel.harga) != EOF) {
        // Calculate the modified price
        long int hargaNumerik = konversiHarga(pesanan.hotel.harga);
        hargaNumerik *= pesanan.jumlah_kamar * pesanan.durasi;

        printf("| %2d | %-35s | %-16s | %-5d | %-5d | %-5d | Rp %-15ld |\n", nomor_pesanan, pesanan.hotel.nama_hotel, pesanan.tanggal_checkin,
            pesanan.durasi, pesanan.jumlah_tamu, pesanan.jumlah_kamar, hargaNumerik);
        nomor_pesanan++;
    }

    printf("-----\n\n");

    fclose(file);
}
```

Tampilan program ketika function dijalankan.

```
----- book@serpong -----
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 2

| No | Nama Hotel | Tanggal Check-in | Durasi | Tamu | Kamar | Harga |
| 1 | D' Rooms Studio & 2BR | 22/04/2024 | 4 | 4 | 4 | Rp 6876032 |
| 2 | D' Rooms Studio & 2BR | 22/04/2024 | 22 | 3 | 3 | Rp 28363632 |

----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: |
```

### 3. Batalan pesanan

Source code batalkan pesanan yang akan dijalankan ketika memilih opsi menu 3.

```
void batalkanPesan() {
    FILE *file = fopen(FILE_PESANAN, "r");
    if (file == NULL) {
        printf("Gagal membuka file pesanan.\n");
        return;
    }

    struct Tiket pesanan[MAX_TIKET];
    int jumlah_pesanan = 0;

    while (fscanf(file, "%[^#]#[^#]#d#d#d#s\n", pesanan[jumlah_pesanan].hotel.nama_hotel,
        pesanan[jumlah_pesanan].tanggal_checkin, &pesanan[jumlah_pesanan].durasi,
        &pesanan[jumlah_pesanan].jumlah_tamu, &pesanan[jumlah_pesanan].jumlah_kamar,
        pesanan[jumlah_pesanan].hotel.harga) != EOF) {
        jumlah_pesanan++;
    }
    fclose(file);

    if (jumlah_pesanan > 0) {
        printf("Pesanan terakhir akan dibatalkan.\n");
        jumlah_pesanan--;
        file = fopen(FILE_PESANAN, "w");
        if (file == NULL) {
            printf("Gagal membuka file pesanan.\n");
            return;
        }
        for (int i = 0; i < jumlah_pesanan; i++) {
            fprintf(file, "%s#s#d#d#d#s\n", pesanan[i].hotel.nama_hotel, pesanan[i].tanggal_checkin,
                pesanan[i].durasi, pesanan[i].jumlah_tamu, pesanan[i].jumlah_kamar,
                pesanan[i].hotel.harga);
        }
        fclose(file);

        printf("Pesanan terakhir berhasil dibatalkan.\n");
    } else {
        printf("Tidak ada pesanan yang bisa dibatalkan.\n");
    }
}
```

Tampilan program ketika function dijalankan

```
----- book@serpong -----
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 3
Pesanan terakhir akan dibatalkan.
Pesanan terakhir berhasil dibatalkan.
----- book@serpong -----

Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: |
```

#### 4. Lihat Hotel

Berikut adalah source code lihat hotel, yang akan dijalankan Ketika memilih opsi menu 4

```
void lihatHotel() {
    int pilihan;
    printf("\nPilih opsi:\n");
    printf("1. Lihat Hotel Berdasarkan Harga Tertinggi\n");
    printf("2. Lihat Hotel Berdasarkan Harga Terendah\n");
    printf("3. Urutkan Hotel Berdasarkan Rating Tertinggi\n");
    printf("4. Urutkan Hotel Berdasarkan Rating Terendah\n");
    printf("5. Urutkan Hotel Berdasarkan Bintang Terkecil\n");
    printf("6. Urutkan Hotel Berdasarkan Bintang Terbesar\n");
    printf("7. Lihat Hotel Berdasarkan Tipe Penginapan\n");
    printf("Pilihan Anda: ");
    scanf("%d", &pilihan);

    switch (pilihan) {
        case 1:
            lihatHotelBerdasarkanHarga('T');
            break;
        case 2:
            lihatHotelBerdasarkanHarga('K');
            break;
        case 3:
            urutHotelBerdasarkanRating('T');
            break;
        case 4:
            urutHotelBerdasarkanRating('K');
            break;
        case 5:
            urutHotelBerdasarkanBintang('K');
            break;
        case 6:
            urutHotelBerdasarkanBintang('T');
            break;
        case 7:
            lihatHotelBerdasarkanTipe();
            break;
        default:
            printf("Pilihan tidak valid.\n");
    }
}
```

Tampilan program Ketika function dijalankan

```
----- book@serpong -----
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 4

Pilih opsi:
1. Lihat Hotel Berdasarkan Harga Tertinggi
2. Lihat Hotel Berdasarkan Harga Terendah
3. Urutkan Hotel Berdasarkan Rating Tertinggi
4. Urutkan Hotel Berdasarkan Rating Terendah
5. Urutkan Hotel Berdasarkan Bintang Terkecil
6. Urutkan Hotel Berdasarkan Bintang Terbesar
7. Lihat Hotel Berdasarkan Tipe Penginapan
Pilihan Anda: |
```

## 5. Lihat Hotel berdasarkan harga tertinggi

Source code lihat hotel berdasarkan harga tertinggi, yang akan dijalankan Ketika memilih opsi lihat hotel 1.

```
void lihatHotelBerdasarkanHarga(char urutan) {
    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }
    struct Hotel hotels[MAX_TIKET];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];

    while (fscanf(file, "%[^%]%%c%lf%%c%fs\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
        hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
        hotels[jumlah_hotel].rating = rating;
        hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
        strcpy(hotels[jumlah_hotel].harga, harga);
        jumlah_hotel++;
    }

    fclose(file);

    for (int i = 0; i < jumlah_hotel - 1; i++) {
        for (int j = i + 1; j < jumlah_hotel; j++) {
            long int harga_i = konversiHarga(hotels[i].harga);
            long int harga_j = konversiHarga(hotels[j].harga);
            if ((urutan == 'I' && harga_i < harga_j) ||
                (urutan == 'K' && harga_i > harga_j)) {
                struct Hotel temp = hotels[i];
                hotels[i] = hotels[j];
                hotels[j] = temp;
            }
        }
    }

    if (urutan == 'I') {
        printf("\nDaftar Hotel Berdasarkan Harga Tertinggi:\n");
    } else {
        printf("\nDaftar Hotel Berdasarkan Harga Terendah:\n");
    }

    printf("%-5s %-35s %-10s %-7s %-14s %s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-5d %-35s %-10c %-7.1f %-14s %s\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}
```

Tampilan progam Ketika function dijalankan

```
----- book@serpong -----
-----
Pilih menu:
1. Pesan Tiket
2. Lihat Tiket
3. Batalkan Pesanan
4. Lihat Hotel
5. Keluar
Pilihan Anda: 4

Pilih opsi:
1. Lihat Hotel Berdasarkan Harga Tertinggi
2. Lihat Hotel Berdasarkan Harga Terendah
3. Urutkan Hotel Berdasarkan Rating Tertinggi
4. Urutkan Hotel Berdasarkan Rating Terendah
5. Urutkan Hotel Berdasarkan Bintang Terkecil
6. Urutkan Hotel Berdasarkan Bintang Terbesar
7. Lihat Hotel Berdasarkan Tipe Penginapan
Pilihan Anda: 1

Daftar Hotel Berdasarkan Harga Tertinggi:
No      Nama                               Bintang  Rating  Jenis      Harga
1       JHL Solitaire Gading Serpong         5         8.7     Hotel      Rp 2.242.680
2       Episode Gading Serpong              4         8.8     Hotel      Rp 1.454.545
3       VIVERE Hotel                        4         8.8     Hotel      Rp 1.015.666
4       Atria Residences Gading Serpong      4         8.7     Hotel      Rp 845.179
5       Ibis Gading Serpong                  3         8.5     Hotel      Rp 643.720
6       D' Rooms Studio & 2BR                 0         9.0     Apartemen  Rp 429.752
7       Avilla Residence                    2         8.0     Guest House Rp 413.223
8       Lemo Hotel Serpong                   3         8.4     Hotel      Rp 409.000
9       Hotel Gading Serpong                 2         8.2     Hotel      Rp 343.802
10      Starlet Hotel Serpong                 1         8.3     Hotel      Rp 288.000
```

## 6. Lihat hotel berdasarkan harga terendah

Source code lihat hotel berdasarkan harga terendah yang akan dijalankan ketika memilih menu lihat hotel 2.

```
void lihatHotelBerdasarkanHarga(char urutan) {
    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }
    struct Hotel hotels[MAX_TIKET];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];

    while (fscanf(file, "%[^#]#%c%f#%c#%s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
        hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
        hotels[jumlah_hotel].rating = rating;
        hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
        strcpy(hotels[jumlah_hotel].harga, harga);
        jumlah_hotel++;
    }

    fclose(file);

    for (int i = 0; i < jumlah_hotel - 1; i++) {
        for (int j = i + 1; j < jumlah_hotel; j++) {
            long int harga_i = konversiHarga(hotels[i].harga);
            long int harga_j = konversiHarga(hotels[j].harga);
            if ((urutan == 'T' && harga_i < harga_j) ||
                (urutan == 'K' && harga_i > harga_j)) {
                struct Hotel temp = hotels[i];
                hotels[i] = hotels[j];
                hotels[j] = temp;
            }
        }
    }

    if (urutan == 'T') {
        printf("\nDaftar Hotel Berdasarkan Harga Tertinggi:\n");
    } else {
        printf("\nDaftar Hotel Berdasarkan Harga Terendah:\n");
    }

    printf("%-5s %-35s %-10s %-7s %-14s %s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-5d %-35s %-10c %-7.1f %-14s Rp %s\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}
```

Tampilan program Ketika function dijalankan

```
Pilih opsi:
1. Lihat Hotel Berdasarkan Harga Tertinggi
2. Lihat Hotel Berdasarkan Harga Terendah
3. Urutkan Hotel Berdasarkan Rating Tertinggi
4. Urutkan Hotel Berdasarkan Rating Terendah
5. Urutkan Hotel Berdasarkan Bintang Terkecil
6. Urutkan Hotel Berdasarkan Bintang Terbesar
7. Lihat Hotel Berdasarkan Tipe Penginapan
Pilihan Anda: 2

Daftar Hotel Berdasarkan Harga Terendah:
No    Nama                               Bintang  Rating  Jenis      Harga
1     Starlet Hotel Serpong              1        8.3     Hotel      Rp 288.000
2     Hotel Gading Serpong              2        8.2     Hotel      Rp 343.802
3     Lemo Hotel Serpong                3        8.4     Hotel      Rp 409.000
4     Avilla Residence                  2        8.0     Guest House Rp 413.223
5     D' Rooms Studio & 2BR              0        9.0     Apartemen  Rp 429.752
6     Ibis Gading Serpong               3        8.5     Hotel      Rp 643.720
7     Atria Residences Gading Serpong    4        8.7     Hotel      Rp 845.179
8     VIVERE Hotel                     4        8.8     Hotel      Rp 1.015.666
9     Episode Gading Serpong            4        8.8     Hotel      Rp 1.454.545
10    JHL Solitaire Gading Serpong       5        8.7     Hotel      Rp 2.242.680
```



## 7. Urutkan berdasarkan rating tertinggi

Source code lihat hotel berdasarkan rating yang akan dijalankan ketika memilih opsi menu lihat hotel 3.

```
void urutHotelBerdasarkanRating(char urutan) {
    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }

    struct Hotel hotels[MAX_TIKET];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];

    while (fscanf(file, "%[^$]${%c}%f%fc%s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
        hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
        hotels[jumlah_hotel].rating = rating;
        hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
        strcpy(hotels[jumlah_hotel].harga, harga);
        jumlah_hotel++;
    }

    fclose(file);

    if (urutan == 'T') {
        printf("\nDaftar Hotel Berdasarkan Rating Tertinggi:\n");
    } else {
        printf("\nDaftar Hotel Berdasarkan Rating Terendah:\n");
    }

    for (int i = 0; i < jumlah_hotel - 1; i++) {
        for (int j = i + 1; j < jumlah_hotel; j++) {
            if ((urutan == 'T' && hotels[i].rating < hotels[j].rating) ||
                (urutan == 'K' && hotels[i].rating > hotels[j].rating)) {
                struct Hotel temp = hotels[i];
                hotels[i] = hotels[j];
                hotels[j] = temp;
            }
        }
    }

    printf("%-8s %-35s %-10s %-7s %-15s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-8s %-35s %-10s %-7.1f %-15s Rp %s\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}
```

Tampilan program ketika function dijalankan.

Pilih opsi:

1. Lihat Hotel Berdasarkan Harga Tertinggi
  2. Lihat Hotel Berdasarkan Harga Terendah
  3. Urutkan Hotel Berdasarkan Rating Tertinggi
  4. Urutkan Hotel Berdasarkan Rating Terendah
  5. Urutkan Hotel Berdasarkan Bintang Terkecil
  6. Urutkan Hotel Berdasarkan Bintang Terbesar
  7. Lihat Hotel Berdasarkan Tipe Penginapan
- Pilihan Anda: 3

Daftar Hotel Berdasarkan Rating Tertinggi:

No	Nama	Bintang	Rating	Jenis	Harga
1	D' Rooms Studio & 2BR	0	9.0	Apartemen	Rp 429.752
2	Episode Gading Serpong	4	8.8	Hotel	Rp 1.454.545
3	VIVERE Hotel	4	8.8	Hotel	Rp 1.015.666
4	JHL Solitaire Gading Serpong	5	8.7	Hotel	Rp 2.242.680
5	Atria Residences Gading Serpong	4	8.7	Hotel	Rp 845.179
6	Ibis Gading Serpong	3	8.5	Hotel	Rp 643.720
7	Lemo Hotel Serpong	3	8.4	Hotel	Rp 409.000
8	Starlet Hotel Serpong	1	8.3	Hotel	Rp 288.000
9	Hotel Gading Serpong	2	8.2	Hotel	Rp 343.802
10	Avilla Residence	2	8.0	Guest House	Rp 413.223

## 8. Urutkan berdasarkan rating terendah

Source code lihat hotel berdasarkan rating yang akan dijalankan ketika memilih opsi menu lihat hotel 4.

```
void urutHotelBerdasarkanRating(char urutan) {
    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }

    struct Hotel hotels[MAX_TIKET];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];

    while (fscanf(file, "%[^$]${c%#ff%#s}\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
        hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
        hotels[jumlah_hotel].rating = rating;
        hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
        strcpy(hotels[jumlah_hotel].harga, harga);
        jumlah_hotel++;
    }

    fclose(file);

    if (urutan == 'T') {
        printf("\nDaftar Hotel Berdasarkan Rating Tertinggi:\n");
    } else {
        printf("\nDaftar Hotel Berdasarkan Rating Terendah:\n");
    }

    for (int i = 0; i < jumlah_hotel - 1; i++) {
        for (int j = i + 1; j < jumlah_hotel; j++) {
            if ((urutan == 'T' && hotels[i].rating < hotels[j].rating) ||
                (urutan == 'K' && hotels[i].rating > hotels[j].rating)) {
                struct Hotel temp = hotels[i];
                hotels[i] = hotels[j];
                hotels[j] = temp;
            }
        }
    }

    printf("%-5s %-35s %-10s %-7s %-15s %s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-5s %-35s %-10s %-7.1f %-15s Rp %s\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}
```

Tampilan program ketika function dijalankan

```
Pilih opsi:
1. Lihat Hotel Berdasarkan Harga Tertinggi
2. Lihat Hotel Berdasarkan Harga Terendah
3. Urutkan Hotel Berdasarkan Rating Tertinggi
4. Urutkan Hotel Berdasarkan Rating Terendah
5. Urutkan Hotel Berdasarkan Bintang Terkecil
6. Urutkan Hotel Berdasarkan Bintang Terbesar
7. Lihat Hotel Berdasarkan Tipe Penginapan
Pilihan Anda: 4

Daftar Hotel Berdasarkan Rating Terendah:
```

No	Nama	Bintang	Rating	Jenis	Harga
1	Avilla Residence	2	8.0	Guest House	Rp 413.223
2	Hotel Gading Serpong	2	8.2	Hotel	Rp 343.802
3	Starlet Hotel Serpong	1	8.3	Hotel	Rp 288.000
4	Lemo Hotel Serpong	3	8.4	Hotel	Rp 409.000
5	Ibis Gading Serpong	3	8.5	Hotel	Rp 643.720
6	JHL Solitaire Gading Serpong	5	8.7	Hotel	Rp 2.242.680
7	Atria Residences Gading Serpong	4	8.7	Hotel	Rp 845.179
8	Episode Gading Serpong	4	8.8	Hotel	Rp 1.454.545
9	VIVERE Hotel	4	8.8	Hotel	Rp 1.015.666
10	D' Rooms Studio & 2BR	0	9.0	Apartemen	Rp 429.752

## 9. Urutkan hotel berdasarkan bintang terendah

Source code lihat hotel berdasarkan bintang yang akan dijalankan ketika memilih opsi menu lihat hotel 5.

```
void urutHotelBerdasarkanBintang(char urutan) {
    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }

    struct Hotel hotels[MAX_TIKET];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];
    while (fscanf(file, "%[^%c%c%c%c%c%c\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
        hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
        hotels[jumlah_hotel].rating = rating;
        hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
        strcpy(hotels[jumlah_hotel].harga, harga);
        jumlah_hotel++;
    }

    fclose(file);

    if (urutan == 'T') {
        printf("\nDaftar Hotel Berdasarkan Bintang Terbesar:\n");
    } else {
        printf("\nDaftar Hotel Berdasarkan Bintang Terkecil:\n");
    }

    for (int i = 0; i < jumlah_hotel - 1; i++) {
        for (int j = i + 1; j < jumlah_hotel; j++) {
            if ((urutan == 'T' && hotels[i].bintang_hotel < hotels[j].bintang_hotel) ||
                (urutan == 'K' && hotels[i].bintang_hotel > hotels[j].bintang_hotel)) {
                struct Hotel temp = hotels[i];
                hotels[i] = hotels[j];
                hotels[j] = temp;
            }
        }
    }

    printf("%-5s %-35s %-10s %-7s %-10s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-5s %-35s %-10s %-7s %-10s Rp\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}
```

Tampilan program ketika function dijalankan

Pilih opsi:

1. Lihat Hotel Berdasarkan Harga Tertinggi
  2. Lihat Hotel Berdasarkan Harga Terendah
  3. Urutkan Hotel Berdasarkan Rating Tertinggi
  4. Urutkan Hotel Berdasarkan Rating Terendah
  5. Urutkan Hotel Berdasarkan Bintang Terkecil
  6. Urutkan Hotel Berdasarkan Bintang Terbesar
  7. Lihat Hotel Berdasarkan Tipe Penginapan
- Pilihan Anda: 5

Daftar Hotel Berdasarkan Bintang Terkecil:

No	Nama	Bintang	Rating	Jenis	Harga
1	D' Rooms Studio & 2BR	0	9.0	Apartemen	Rp 429.752
2	Starlet Hotel Serpong	1	8.3	Hotel	Rp 288.000
3	Avilla Residence	2	8.0	Guest House	Rp 413.223
4	Hotel Gading Serpong	2	8.2	Hotel	Rp 343.802
5	Ibis Gading Serpong	3	8.5	Hotel	Rp 643.720
6	Lemo Hotel Serpong	3	8.4	Hotel	Rp 409.000
7	VIVERE Hotel	4	8.8	Hotel	Rp 1.015.666
8	Atria Residences Gading Serpong	4	8.7	Hotel	Rp 845.179
9	Episode Gading Serpong	4	8.8	Hotel	Rp 1.454.545
10	JHL Solitaire Gading Serpong	5	8.7	Hotel	Rp 2.242.680

**10. Urutkan hotel berdasarkan bintang terbesar.**

Source code lihat hotel berdasarkan bintang, yang akan dijalankan ketika memilih opsi menu lihat hotel 6.

```
void urutHotelBerdasarkanBintang(char urutan) {
    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }

    struct Hotel hotels[MAX_HOTEL];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];

    while (fscanf(file, "%[^\n] %c %f %c %s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
        hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
        hotels[jumlah_hotel].rating = rating;
        hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
        strcpy(hotels[jumlah_hotel].harga, harga);
        jumlah_hotel++;
    }

    fclose(file);

    if (urutan == 'T') {
        printf("\nUrut Hotel Berdasarkan Bintang Terbesar:\n");
    } else {
        printf("\nUrut Hotel Berdasarkan Bintang Terkecil:\n");
    }

    for (int i = 0; i < jumlah_hotel - 1; i++) {
        for (int j = i + 1; j < jumlah_hotel; j++) {
            if ((urutan == 'T' && hotels[i].bintang_hotel < hotels[j].bintang_hotel) ||
                (urutan == 'K' && hotels[i].bintang_hotel > hotels[j].bintang_hotel)) {
                struct Hotel temp = hotels[i];
                hotels[i] = hotels[j];
                hotels[j] = temp;
            }
        }
    }

    printf("%-3s %-3s %-10s %-7s %-10s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-3s %-3s %-10s %-7.1f %-10s\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}
```

Tampilan program ketika function dijalankan.

```
Pilih opsi:
1. Lihat Hotel Berdasarkan Harga Tertinggi
2. Lihat Hotel Berdasarkan Harga Terendah
3. Urutkan Hotel Berdasarkan Rating Tertinggi
4. Urutkan Hotel Berdasarkan Rating Terendah
5. Urutkan Hotel Berdasarkan Bintang Terkecil
6. Urutkan Hotel Berdasarkan Bintang Terbesar
7. Lihat Hotel Berdasarkan Tipe Penginapan
Pilihan Anda: 6

Daftar Hotel Berdasarkan Bintang Terbesar:
No      Nama                               Bintang  Rating  Jenis      Harga
1       JHL Solitaire Gading Serpong        5         8.7    Hotel      Rp 2.242.680
2       Episode Gading Serpong              4         8.8    Hotel      Rp 1.454.545
3       Atria Residences Gading Serpong      4         8.7    Hotel      Rp 845.179
4       VIVERE Hotel                        4         8.8    Hotel      Rp 1.015.666
5       Lemo Hotel Serpong                  3         8.4    Hotel      Rp 409.000
6       Ibis Gading Serpong                 3         8.5    Hotel      Rp 643.720
7       Avilla Residence                    2         8.0    Guest House Rp 413.223
8       Hotel Gading Serpong                2         8.2    Hotel      Rp 343.802
9       Starlet Hotel Serpong               1         8.3    Hotel      Rp 288.000
10      D' Rooms Studio & 2BR               0         9.0    Apartemen  Rp 429.752
```

**11. Lihat hotel berdasarkan tipe penginapan**

Source code lihat hotel berdasarkan bintang, yang akan dijalankan ketika memilih opsi menu lihat hotel 6.

```

id LihatHotelBerdasarkanTipe() {
    char tipe;
    printf("\nSilah tipe penginapan (H = Hotel, A = Apartemen, G = Guest House): ");
    scanf("%c", &tipe);

    FILE *file = fopen(FILE_HOTEL, "r");
    if (file == NULL) {
        printf("Gagal membuka file hotel.\n");
        return;
    }

    struct Hotel hotels[MAX_TIKET];
    int jumlah_hotel = 0;

    char nama_hotel[MAX_NAMA_HOTEL];
    char bintang_hotel;
    float rating;
    char jenis_penginapan;
    char harga[20];

    while (fscanf(file, "%[^\n] %c %f %f %c %s\n", nama_hotel, &bintang_hotel, &rating, &jenis_penginapan, harga) != EOF) {
        if (jenis_penginapan == tipe) {
            strcpy(hotels[jumlah_hotel].nama_hotel, nama_hotel);
            hotels[jumlah_hotel].bintang_hotel = bintang_hotel;
            hotels[jumlah_hotel].rating = rating;
            hotels[jumlah_hotel].jenis_penginapan = jenis_penginapan;
            strcpy(hotels[jumlah_hotel].harga, harga);
            jumlah_hotel++;
        }
    }

    fclose(file);

    printf("\nDaftar Hotel Berdasarkan Tipe Penginapan '%c':\n", tipe);
    printf("%-3s %-3s %-10s %-3s %-1s %s\n", "No", "Nama", "Bintang", "Rating", "Jenis", "Harga");
    for (int i = 0; i < jumlah_hotel; i++) {
        printf("%-3d %-3s %-10s %-7.1f %-1s %s\n", i + 1, hotels[i].nama_hotel, hotels[i].bintang_hotel, hotels[i].rating,
            hotels[i].jenis_penginapan == 'H' ? "Hotel" : (hotels[i].jenis_penginapan == 'A' ? "Apartemen" : "Guest House"), hotels[i].harga);
    }
}

```

Tampilan program ketika function dijalankan.

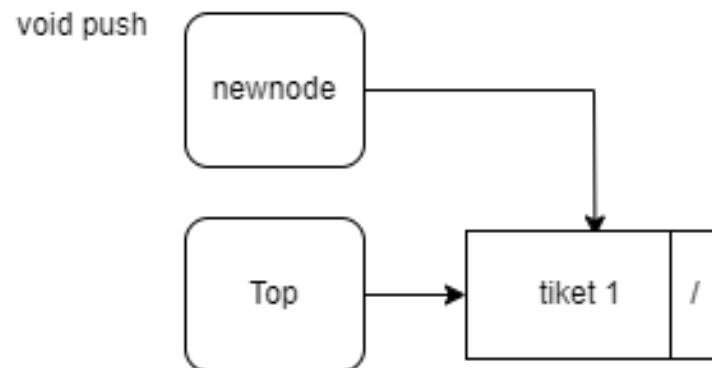
```
Pilih opsi:
1. Lihat Hotel Berdasarkan Harga Tertinggi
2. Lihat Hotel Berdasarkan Harga Terendah
3. Urutkan Hotel Berdasarkan Rating Tertinggi
4. Urutkan Hotel Berdasarkan Rating Terendah
5. Urutkan Hotel Berdasarkan Bintang Terkecil
6. Urutkan Hotel Berdasarkan Bintang Terbesar
7. Lihat Hotel Berdasarkan Tipe Penginapan
Pilihan Anda: 7

Pilih tipe penginapan (H = Hotel, A = Apartemen, G = Guest House): H

Daftar Hotel Berdasarkan Tipe Penginapan 'H':
```

No	Nama	Bintang	Rating	Jenis	Harga
1	Episode Gading Serpong	4	8.8	Hotel	Rp 1.454.545
2	JHL Solitaire Gading Serpong	5	8.7	Hotel	Rp 2.242.680
3	Hotel Gading Serpong	2	8.2	Hotel	Rp 343.802
4	Atria Residences Gading Serpong	4	8.7	Hotel	Rp 845.179
5	Lemo Hotel Serpong	3	8.4	Hotel	Rp 409.000
6	VIVERE Hotel	4	8.8	Hotel	Rp 1.015.666
7	Ibis Gading Serpong	3	8.5	Hotel	Rp 643.720
8	Starlet Hotel Serpong	1	8.3	Hotel	Rp 288.000

#### IV. Implementasi Linked List (Stack)



Void ini digunakan untuk menambah pesanan tiket ke dalam stack. Data tiket yang dipesan akan disimpan ke dalam node baru yang selanjutnya ditambah ke stack.

## V. Evaluasi

Nama	Pembagian Tugas	Kinerja Kelompok	Hasil Kerja
Hyun Jung	Pembagian tugas masing-masing anggota sudah berjalan dengan sangat efektif	Kinerja setiap anggota kelompok sudah sangat baik, meskipun terdapat masalah, dan tidak mengerti tapi tetap berusaha untuk mencari solusi.	Hasil kerja kelompok sudah baik, lebih dari puas
Jeremy Marvel	Pembagian tugas adil dan merata, semua anggota mendapat bagian masing-masing tugas yang perlu dikerjakan.	Kinerja kelompok sangat baik, mengerjakan sesuai dengan jobdesk masing-masing, dan saling membantu jika ada yang kesulitan.	Hasil kerja yang dibuat sangat baik, memuaskan dan sesuai dengan ekspektasi awal.
Yua Pedroza	Pembagian tugas sudah merata dan adil, setiap anggota mendapat bagian dengan baik.	Kinerja kelompok sudah baik, saling membantu dan saling support.	Hasil kerja memuaskan, bisa selesai sesuai target yang diinginkan dan tepat waktu.
Marcello Zefanya Albertz zega	Pembagian tugas yang sudah baik dan tanggung jawab yang jelas, setiap anggota dapat berkontribusi secara maksimal dalam mencapai tujuan bersama.	kelompok memiliki keaktifan yang tinggi dalam melakukan diskusi untuk menyelesaikan masalah.	kami telah berhasil menyelesaikan tugas-tugas yang diberikan dengan baik dan efisien.