**TUGAS MAKALAH**

**STUDI KASUS PEMESANAN HOTEL VIA MOBILE**



DISUSUN OLEH :

1. Abie Ramadhan
2. Muhhamad Prendi Arahman
3. Rocky Federik Hamonangan
4. Windo Fuandi

**PROGRAM STUDI TEKNIK INFORMATIKA**

**UNIVERSITAS DIAN NUSANTARA**

**JAKARTA**

**2023**

**Kata Pengantar**

Kami dengan senang hati mempersembahkan makalah ini sebagai bagian dari tugas mata kuliah Algoritma dan Pemrograman 2. Makalah ini berfokus pada studi kasus Hotel BlueDorzz, di mana kami menerapkan berbagai fitur menggunakan konsep algoritma dan pemrograman. Kami berharap makalah ini dapat memberikan pemahaman yang lebih baik tentang implementasi praktis algoritma dan pemrograman dalam industri perhotelan.

Kami ingin mengucapkan terima kasih kepada dosen pengampu mata kuliah yang telah memberikan kesempatan untuk belajar dan menerapkan pengetahuan kami dalam konteks nyata. Kami juga ingin berterima kasih kepada Hotel BlueDorzzyang telah memberikan izin untuk menggunakan studi kasus mereka sebagai bahan dalam makalah ini.

Semoga makalah ini bermanfaat dan memberikan wawasan yang berharga bagi pembaca.

**Daftar Isi**

I. Kata Pengantar................................................................................................................................... i

II. Daftar Isi .......................................................................................................................................... ii

Pendahuluan ....................................................................................................................................... 1

a. Latar Belakang ................................................................................................................................. 1

b. Rumusan Masalah ........................................................................................................................... 1

c. Tujuan .............................................................................................................................................. 1

III. Pembahasan ................................................................................................................................... 2

a. Fitur 1: Input/Output ....................................................................................................................... 2

b. Fitur 2: Operator Logika ................................................................................................................... 2

c. Fitur 3: Percabangan Bersarang (Nested IF) ..................................................................................... 2

d. Fitur 4: Perulangan Bersarang (Nested Loop) .................................................................................. 2

e. Fitur 6: Integrasi Nested IF, Nested Loop, dan Fungsi ...................................................................... 3

f. Fitur 7: Tuples …………………………………………………………....................................................................... 3

g. Fitur 8: Dictionaries dan Exceptions ………………………………………………………….....................................3

IV. Penutup .......................................................................................................................................... 9

a. Kesimpulan ...................................................................................................................................... 9

b. Saran ............................................................................................................................................... 9

c. Daftar Gambar

Gambar 1: Flowchart Fitur 1 - Input/Output (Halaman 5) Gambar 2: Algoritma Fitur 3 - Percabangan Bersarang (Halaman 7) Gambar 3: Contoh Source Code Fitur 5 - Fungsi (Halaman 9)

**Bab I**

**Pendahuluan**

Latar Belakang: Dalam industri perhotelan, sistem pemesanan kamar online menjadi sangat penting dalam memfasilitasi reservasi dan pengelolaan kamar. Hotel BlueDorzz, sebagai salah satu hotel terkemuka, ingin meningkatkan efisiensi operasional mereka dengan mengimplementasikan fitur-fitur yang relevan dalam sistem pemesanan mereka.

Rumusan Masalah: Dalam konteks ini, kami mengidentifikasi beberapa fitur utama yang harus diimplementasikan dalam sistem pemesanan kamar online Hotel BlueDorzz. Fitur-fitur ini meliputi input/output, operator logika, percabangan bersarang, perulangan bersarang, fungsi, integrasi dari nested if, nested loop, dan fungsi, serta penggunaan tuples, dictionaries, exceptions, dan pemrosesan data.

Tujuan: Tujuan dari makalah ini adalah untuk menjelaskan secara detail implementasi masing-masing fitur yang diimplementasikan dalam studi kasus Hotel BlueDorzz. Kami akan membahas algoritma, flowchart, dan contoh source code untuk setiap fitur. Selain itu, kami juga akan menunjukkan hasil dari program yang telah di-compile.

**Bab II**

**Pembahasan**

1. Fitur 1: Input/Output
   * Penjelasan Fitur
   * Flowchart Fitur 1 - Input/Output
   * Contoh Source Code
   * Hasil Compile Program
2. Fitur 2: Operator Logika
   * Penjelasan Fitur
   * Flowchart Fitur 2 - Operator Logika
   * Contoh Source Code
   * Hasil Compile Program
3. Fitur 3: Percabangan Bersarang (Nested IF)
   * Penjelasan Fitur
   * Algoritma Fitur 3 - Percabangan Bersarang
   * Contoh Source Code
   * Hasil Compile Program
4. Fitur 4: Perulangan Bersarang (Nested Loop)

* Flowchart Fitur 1 - Input/Output
* Contoh Source Code
* Hasil Compile Program

1. Fitur 6: Integrasi Nested IF, Nested Loop, dan Fungsi

* Contoh Source Code
* Penjelasan Lengkap

1. Fitur 7: Tuples

* Penjelasan Lengkap
* Contoh Source Code

1. Fitur 8: Dictionaries dan Exceptions

* Penjelasan
* Contoh Source Code

**Fitur 1: Input/Output**

Dalam Python, input dan output merujuk pada proses pengambilan data (input) dari pengguna atau sumber eksternal dan penampilan data (output) kepada pengguna atau ke sumber eksternal. Input dan output sangat penting dalam pemrograman karena mereka memungkinkan interaksi antara program dan pengguna atau lingkungan sekitarnya.

Input:

Input dalam Python mengacu pada mekanisme yang digunakan untuk menerima data atau informasi dari pengguna atau sumber eksternal ke dalam program. Biasanya, input dilakukan melalui keyboard

atau melalui file eksternal. Di Python, fungsi input() digunakan untuk menerima input dari pengguna melalui keyboard.

Output:

Output dalam Python merujuk pada proses menampilkan data atau informasi kepada pengguna atau ke sumber eksternal. Output dapat ditampilkan ke layar (console), ditulis ke file, atau dikirim ke perangkat lain. Di Python, fungsi print() digunakan untuk menampilkan output ke layar.

**FLOWCHART**

**Masukan Jumlah Malam**

**Masukan Jumlah Kamar**

**Masukan Nama Hotel**



**Tambahan**

**Kamar ?**

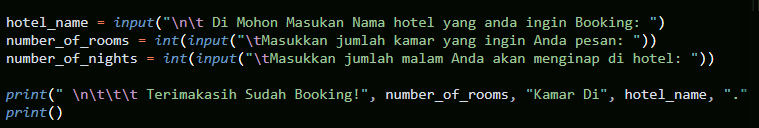


**Masukan Nama Tamu**

**Pilih Tipe Kamar**

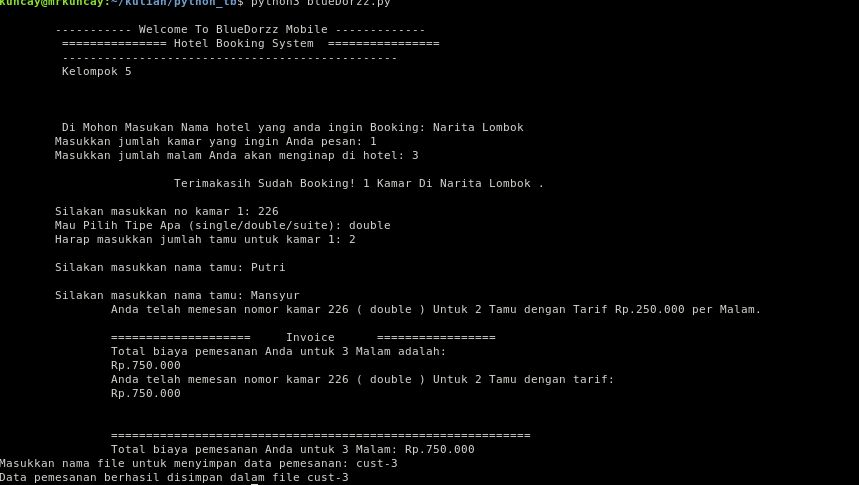
**Masukan No Kamar**

**Contoh Source Code:**

****

Ini contoh input  **nama hotel, nomer ruangan, dan jumlah malam.** Dan output nya menampilkan tersebut

**Output**



**Fitur 2: Operator Logika**

Pada fungsi **calculate\_total\_cost**, terdapat penggunaan operator aritmatika untuk melakukan perhitungan biaya total. Berikut adalah penjelasan operator aritmatika yang digunakan:

Operator Tambah (+): Operator ini digunakan untuk menjumlahkan nilai atau melakukan penambahan. Dalam konteks fungsi ini, operator tambah digunakan untuk menambahkan biaya per malam (**room\_rate**) ke biaya total (**calculate\_total\_cost(...)**) setiap kali rekursi dilakukan.

Operator Perkalian (\*): Operator ini digunakan untuk mengalikan nilai atau melakukan perkalian. Pada fungsi ini, operator perkalian digunakan untuk mengalikan jumlah malam (**number\_of\_nights**) dengan biaya per malam (**room\_rate**) untuk mendapatkan biaya total setiap kali rekursi dilakukan.

Operator-operator aritmatika ini digunakan secara berurutan dalam rekursi fungsi **calculate\_total\_cost** untuk mengakumulasi biaya total dengan menjumlahkan biaya per malam untuk setiap kamar yang dipesan.

**FLowchart**

**Room rate**

**Number of Night**

**Number of Room**

**\* Room Rate**

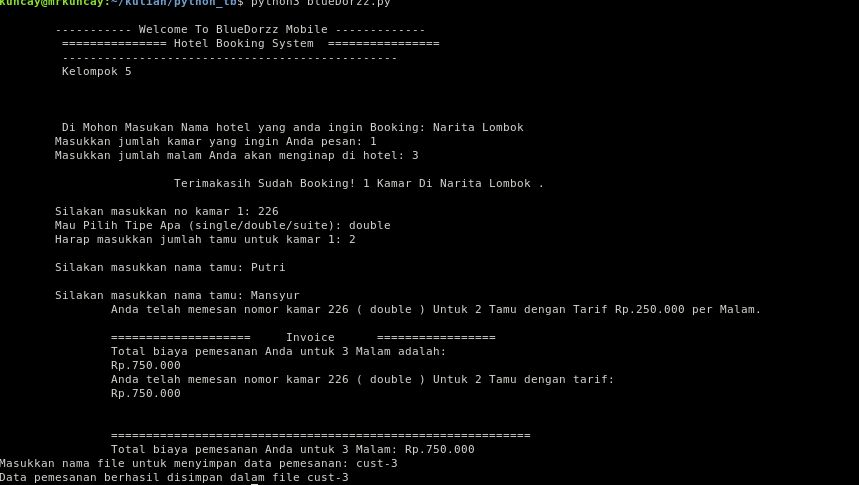
**+ Number Of Night**

**Current room +1**

**Contoh Source Code**

****

**Output**



Dari Output ini adalah jumlah dari harga kamar per malam dan tipe kamarnya.

**Fitur 3: Percabangan Bersarang (Nested IF)**

Pertama, terdapat kondisi if pertama yang memeriksa apakah **room\_type** ada dalam daftar **room\_types**. **room\_types** merupakan sebuah variabel yang mungkin berisi tipe-tipe kamar yang valid. Jika **room\_type** tidak ada dalam **room\_types**, maka kondisi ini akan terpenuhi dan akan dicetak pesan "Tipe Kamar Tidak Valid, Harap Di Coba Lagi". Setelah itu, program akan melanjutkan iterasi ke iterasi selanjutnya menggunakan pernyataan **continue**.

Jika **room\_type** terdapat dalam **room\_types**, maka kondisi if pertama tidak terpenuhi dan program akan melanjutkan ke dalam blok if yang lebih dalam. Di dalam blok if ini, terdapat beberapa kondisi if yang memeriksa nilai **room\_type** secara lebih spesifik. Jika **room\_type** adalah "single", maka variabel **room\_rate** akan diatur menjadi 200. Jika **room\_type** adalah "double", maka **room\_rate** diatur menjadi 250. Jika **room\_type** adalah "suite", maka **room\_rate** diatur menjadi 500.

Dengan menggunakan nested if ini, program akan memeriksa tipe kamar yang dimasukkan oleh pengguna dan mengatur **room\_rate** sesuai dengan tipe kamar yang valid. Jika tipe kamar tidak valid, maka pesan akan dicetak. Jika tipe kamar valid, maka **room\_rate** akan diatur sesuai dengan tipe kamar tersebut.

**Flowchart**

**Suite**

**Double**

**Single**

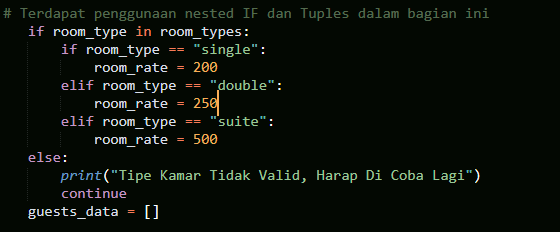


**Room Rate 500**

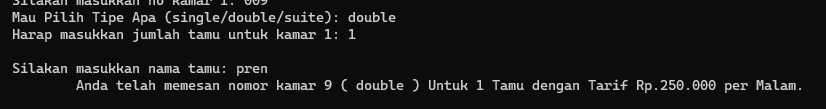
**Room Rate 250**

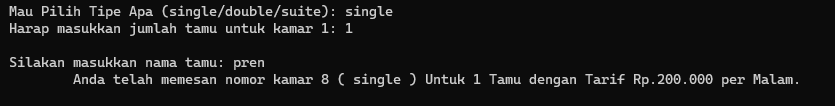
**Room Rate 200**

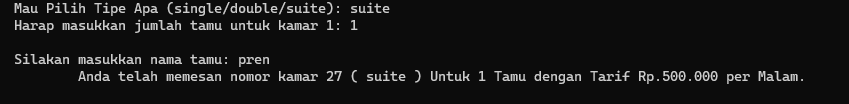
**Contoh Source Code**

****

**Ouput**

****

****

****

**Tipe:**

**Double: 250,000 / Single: 200,000 / Suite = 500,000**

**Fitur 4: Perulangan Bersarang (Nested Loop)**

Dalam konteks ini, terdapat dua loop yang bersarang (nested). Loop pertama digunakan untuk mengiterasi melalui jumlah kamar yang ingin dipesan oleh pengguna. Loop kedua digunakan untuk mengiterasi melalui jumlah tamu yang akan menginap di setiap kamar.

Pada setiap iterasi loop pertama, pengguna diminta untuk memasukkan nomor kamar, tipe kamar, dan jumlah tamu untuk kamar tersebut. Kemudian, loop kedua digunakan untuk mengumpulkan informasi tamu sebanyak jumlah tamu yang dimasukkan.

Dengan menggunakan nested loop, program dapat mengatur pengumpulan data tamu untuk setiap kamar secara terpisah. Ini memungkinkan pengguna untuk memasukkan informasi tamu yang sesuai dengan jumlah tamu yang diinginkan untuk setiap kamar yang dipesan.

Dengan demikian, nested loop pada program tersebut membantu dalam pengumpulan data tamu secara terstruktur dan memungkinkan pengguna untuk mengelola informasi tamu dengan lebih baik dalam konteks pemesanan hotel.

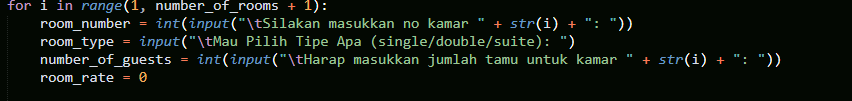
**Flowchart**

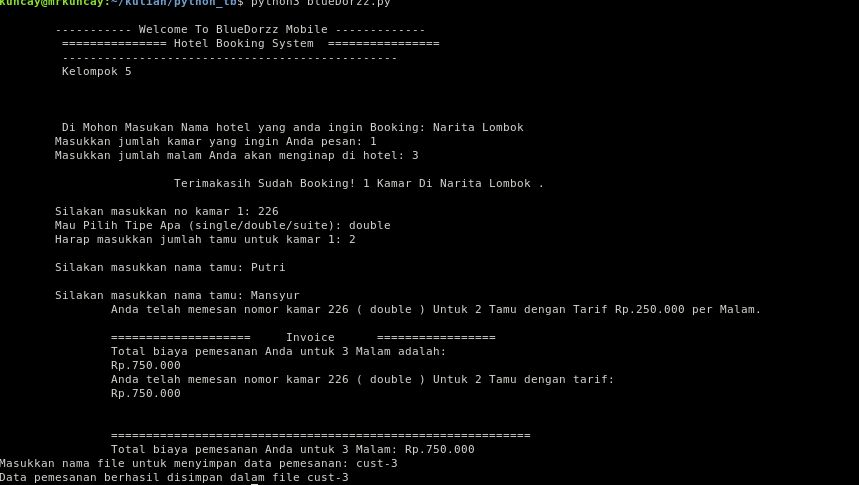
**Masukan Jumlah Tamu**

**Pilih Tipe Kamar**

**Masukan No Kamar**

**Source Code**

****

**Output**

**Fitur 6: Integrasi Nested IF, Nested Loop, dan Fungsi**

Fungsi **print\_booking\_details** digunakan untuk mencetak rincian pemesanan kamar. Fungsi ini memiliki empat parameter:

* **room\_number**: Nomor kamar yang telah dipesan.
* **room\_type**: Tipe kamar yang telah dipesan (misalnya, "single", "double", atau "suite").
* **number\_of\_guests**: Jumlah tamu yang akan menginap di kamar tersebut.
* **room\_rate**: Tarif kamar per malam.

Fungsi ini akan mencetak pesan yang berisi rincian pemesanan, termasuk nomor kamar, tipe kamar, jumlah tamu, dan tarif kamar. Contoh pesan yang dicetak adalah sebagai berikut: "Anda telah memesan nomor kamar 123 (single) Untuk 2 Tamu dengan Tarif Rp.200.000 per Malam."

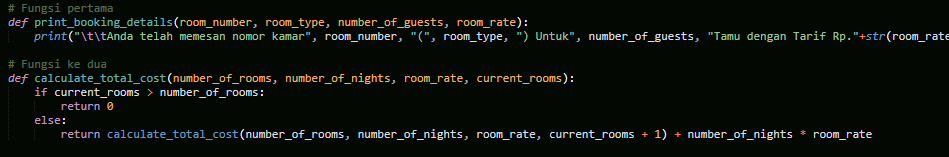
Fungsi **calculate\_total\_cost** digunakan untuk menghitung total biaya pemesanan. Fungsi ini memiliki empat parameter:

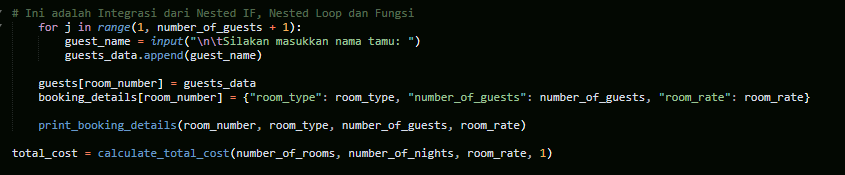
* **number\_of\_rooms**: Jumlah kamar yang dipesan.
* **number\_of\_nights**: Jumlah malam menginap.
* **room\_rate**: Tarif kamar per malam.
* **current\_rooms**: Jumlah kamar yang telah dihitung saat melakukan rekursi (digunakan dalam pemanggilan rekursif).

Fungsi ini menggunakan rekursi untuk menghitung total biaya pemesanan. Pada setiap rekursi, biaya kamar untuk satu malam dikalikan dengan jumlah malam menginap, dan rekursi dilakukan untuk kamar selanjutnya. Pada rekursi terakhir (ketika **current\_rooms** melebihi **number\_of\_rooms**), fungsi mengembalikan nilai 0 untuk menghentikan rekursi. Contoh penggunaan fungsi ini adalah sebagai berikut: **total\_cost = calculate\_total\_cost(3, 5, 200, 1)** Hasilnya akan menghasilkan total biaya pemesanan untuk 3 kamar selama 5 malam dengan tarif kamar Rp.200.000 per malam.

Harap dicatat bahwa untuk kedua fungsi ini, parameter yang diteruskan harus sesuai dengan tipe data yang diharapkan. Misalnya, **room\_number** harus berupa bilangan bulat, **room\_type** harus berupa string, **number\_of\_guests** harus berupa bilangan bulat, dan **room\_rate** harus berupa bilangan.

**Contoh Source Code**

****

****

Bagian tersebut merupakan integrasi dari nested IF, nested loop, dan fungsi dalam sebuah program. Berikut penjelasan untuk setiap baris kode:

**for j in range(1, number\_of\_guests + 1):**

Ini adalah loop **for** yang akan mengulang sebanyak **number\_of\_guests** kali.

Variabel **j** digunakan sebagai iterator untuk menghitung tamu ke-1, ke-2, ke-3, dan seterusnya.

**guest\_name = input("\n\tSilakan masukkan nama tamu: ")**

Pada setiap iterasi loop, program akan meminta pengguna untuk memasukkan nama tamu melalui input.

Nama tamu yang dimasukkan akan disimpan dalam variabel **guest\_name**.

**guests\_data.append(guest\_name)**

Nama tamu yang telah dimasukkan akan ditambahkan ke dalam list **guests\_data**.

Setiap tamu akan memiliki entri nama sendiri dalam list ini.

**guests[room\_number] = guests\_data**

Setelah loop selesai, list **guests\_data** yang berisi nama-nama tamu akan disimpan dalam dictionary **guests**.

Key-nya adalah nomor kamar **room\_number**, dan value-nya adalah list nama-nama tamu untuk kamar tersebut.

**booking\_details[room\_number] = {"room\_type": room\_type, "number\_of\_guests": number\_of\_guests, "room\_rate": room\_rate}**

Informasi rincian pemesanan, seperti tipe kamar, jumlah tamu, dan tarif kamar, disimpan dalam dictionary **booking\_details**.

Key-nya adalah nomor kamar **room\_number**, dan value-nya adalah dictionary yang berisi rincian pemesanan tersebut.

**print\_booking\_details(room\_number, room\_type, number\_of\_guests, room\_rate)**

Fungsi **print\_booking\_details** dipanggil untuk mencetak rincian pemesanan kamar berdasarkan nilai yang diberikan, seperti nomor kamar, tipe kamar, jumlah tamu, dan tarif kamar.

**total\_cost = calculate\_total\_cost(number\_of\_rooms, number\_of\_nights, room\_rate, 1)**

Fungsi **calculate\_total\_cost** dipanggil untuk menghitung total biaya pemesanan berdasarkan jumlah kamar, jumlah malam, tarif kamar, dan jumlah kamar saat ini.

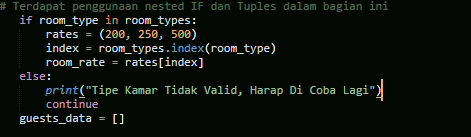
Hasilnya akan disimpan dalam variabel **total\_cost**.

**Fitur 7: Tuples**

1. Membuat Tuples:

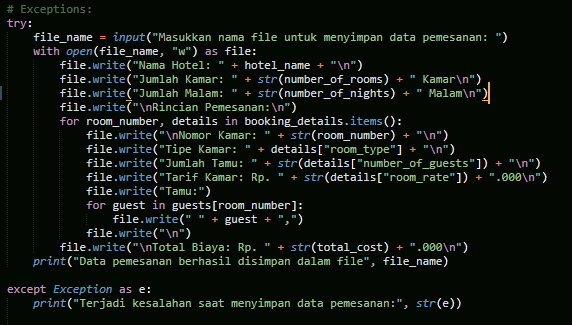
****

2. Pengecekan Tipe kamar:

****

* Pertama, program melakukan pengecekan apakah **room\_type** (tipe kamar yang dimasukkan oleh pengguna) ada dalam tuples **room\_types**. Jika tipe kamar valid, program akan melanjutkan ke langkah berikutnya. Jika tidak valid, program akan mencetak pesan kesalahan dan melanjutkan iterasi berikutnya dalam loop.
* Jika tipe kamar valid, program akan membuat tuples **rates** yang berisi tarif kamar yang sesuai dengan tipe kamar. Misalnya, **(200, 250, 500)** berarti tarif untuk tipe kamar "single" adalah 200, "double" adalah 250, dan "suite" adalah 500.
* Selanjutnya, program mencari indeks tipe kamar dalam tuples **room\_types** menggunakan metode **index()**. Indeks ini akan digunakan untuk mengakses tarif kamar yang sesuai dari tuples **rates**.
* Terakhir, program menetapkan **room\_rate** (tarif kamar) dengan menggunakan tarif yang sesuai berdasarkan indeks yang ditemukan.
* Dengan penggunaan tuples, program dapat memvalidasi tipe kamar yang dimasukkan dan menghubungkannya dengan tarif kamar yang tepat. Ini membantu memastikan bahwa hanya tipe kamar yang valid yang memiliki tarif yang benar yang digunakan dalam proses pemesanan.

**Fitur 8: Dictionaries dan Exceptions**

****

Blok try digunakan untuk menempatkan potensi kode yang dapat menimbulkan pengecualian.

Jika terjadi pengecualian saat menjalankan kode di dalam blok try, proses normal program akan dihentikan, dan program akan mencari blok except yang sesuai.

Exception:

Exception adalah tipe pengecualian umum yang menangkap semua jenis pengecualian yang terjadi.

as e digunakan untuk mengaitkan pengecualian yang terjadi dengan variabel e, sehingga kita dapat mengakses informasi lebih lanjut tentang pengecualian tersebut.

****

Pada bagian ini, pesan kesalahan yang menjelaskan kesalahan yang terjadi akan dicetak.

str(e) digunakan untuk mendapatkan representasi string dari objek pengecualian, sehingga kita dapat mencetak informasi pengecualian dengan lebih jelas.

Dengan menggunakan blok try-except, program dapat menangani pengecualian yang mungkin terjadi saat menyimpan data pemesanan ke file. Jika terjadi kesalahan, pesan kesalahan yang informatif akan dicetak, memungkinkan pengguna untuk mengetahui dan mengatasi masalah yang mungkin terjadi.

**Bab III**

**Penutup**

Kesimpulan: Melalui studi kasus Hotel BlueDorzz, kami berhasil mengimplementasikan berbagai fitur yang diminta menggunakan konsep algoritma dan pemrograman. Fitur-fitur tersebut meliputi input/output, operator logika, percabangan bersarang, perulangan bersarang, fungsi, integrasi dari nested if, nested loop, dan fungsi, serta penggunaan tuples, dictionaries, exceptions, dan pemrosesan data. Implementasi ini memberikan manfaat signifikan bagi efisiensi dan fungsionalitas sistem pemesanan kamar online Hotel BlueDorzz.

Saran: Meskipun implementasi fitur-fitur ini berhasil, ada beberapa saran yang dapat kami berikan untuk perbaikan lebih lanjut. Pertama, penting untuk terus memperbarui dan memelihara sistem agar tetap memenuhi kebutuhan dan perkembangan industri perhotelan. Selain itu, memperluas fitur yang ditawarkan, seperti integrasi dengan sistem pembayaran online, dapat meningkatkan kenyamanan bagi pengguna.

Kesimpulan dan saran tersebut akan menjadi panduan untuk pengembangan dan perbaikan sistem pemesanan kamar online Hotel BlueDorzzdi masa depan.

Terima kasih atas perhatian yang diberikan dalam membaca makalah ini.