**Ruang Lingkup Masalah**

**Deskripsi Masalah:** Masalah yang akan diselesaikan adalah menemukan semua jalur dari titik awal (**S**) ke titik akhir (**E**) dalam sebuah labirin yang diwakili oleh file teks. Labirin terdiri dari sel-sel yang bisa berupa dinding (**#**), jalur yang bisa dilalui (**.**), titik awal (**S**), dan titik akhir (**E**).

**Tugas Utama:**

1. Membaca labirin dari file teks dan menyimpannya dalam struktur data yang sesuai.
2. Menemukan semua jalur yang mungkin dari titik awal (**S**) ke titik akhir (**E**) menggunakan algoritma Greedy.
3. Menentukan dan mencetak jalur terpendek dan jalur terpanjang di antara semua jalur yang ditemukan.
4. Mencetak total jumlah jalur yang ditemukan.

**Kasus :**

1. **File Masukkan Tidak Ada:** Jika file masukkan tidak ada atau tidak bisa dibuka, program akan mencetak pesan kesalahan "File tidak ditemukan" dan keluar.
2. **Tidak Ada Solusi:** Jika tidak ada jalur dari titik awal (**S**) ke titik akhir (**E**), program akan mencetak pesan "Tidak ada jalur yang ditemukan dari Start ke End" dan keluar.
3. **Format Masukkan Tidak Valid:** Jika format file tidak sesuai (misalnya, tidak ada **S** atau **E**, atau labirin tidak berbentuk persegi panjang), program akan mencetak pesan kesalahan "Format file tidak valid" dan keluar.
4. **Ukuran Labirin Minimal:** Labirin harus berukuran minimal 7x7. Jika ukuran kurang dari itu, program akan mencetak pesan kesalahan "Ukuran labirin tidak valid" dan keluar.

**Output:**

* Semua jalur yang mungkin dari **S** ke **E**.
* Jalur terpendek dari **S** ke **E**.
* Jalur terpanjang dari **S** ke **E**.
* Total jumlah jalur yang ditemukan.

**Batasan Implementasi**

1. **Ukuran Labirin:**
   * Program harus menangani labirin dengan ukuran minimal 7x7.
   * Tidak ada batasan maksimal yang keras, tetapi labirin yang sangat besar akan meningkatkan kompleksitas dan waktu eksekusi secara signifikan.
2. **Format Masukkan:**
   * Setiap baris dalam file teks harus memiliki panjang yang sama untuk membentuk labirin persegi panjang.
   * File harus berisi tepat satu **S** (titik awal) dan satu **E** (titik akhir).
3. **Karakter yang Valid:**
   * **S** untuk titik awal.
   * **E** untuk titik akhir.
   * **#** untuk dinding.
   * **.** untuk jalur yang bisa dilalui.
4. **Strategi Pencarian Jalur:**
   * Algoritma menggunakan pendekatan Depth-First Search (DFS) dengan elemen Greedy untuk menemukan semua jalur dari **S** ke **E**.
   * Algoritma akan mencatat setiap jalur yang ditemukan dan menghitung panjangnya untuk menentukan jalur terpendek dan terpanjang.
5. **Penggunaan Memori:**
   * Jalur disimpan dalam array untuk mencatat semua titik yang dilalui dari **S** ke **E**.
   * Penggunaan memori bergantung pada ukuran labirin dan jumlah jalur yang ditemukan.
6. **Waktu Eksekusi:**
   * Waktu eksekusi dipengaruhi oleh ukuran labirin dan kompleksitas jalur di dalamnya.
   * Algoritma Greedy akan mencoba setiap langkah yang terlihat menjanjikan, tetapi tetap memerlukan eksplorasi menyeluruh (seperti DFS) untuk menemukan semua jalur.