Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tugas Besar II IF2211 Strategi Algoritma
Semester II Tahun 2022/2023

Pengaplikasian Algoritma BFS dan DFS dalam Menyelesaikan Persoalan Maze Treasure Hunt

Batas pengumpulan : Jumat, 24 Maret 2023 pukul 23.59 WIB
Arsip pengumpulan :

- Source program yang bisa dijalankan disertai README
- Laporan (soft copy)

Latar belakang :

Tuan Krabs menemukan sebuah labirin distorsi terletak tepat di bawah Krusty Krab bernama El Doremi yang Ia yakini mempunyai sejumlah harta karun di dalamnya dan tentu saja Ia ingin mengambil harta karunnya. Dikarenakan labirinnya dapat mengalami distorsi, Tuan Krabs harus terus mengukur ukuran dari labirin tersebut. Oleh karena itu, Tuan Krabs banyak menghabiskan tenaga untuk melakukan hal tersebut sehingga Ia perlu memikirkan bagaimana caranya agar Ia dapat menelusuri labirin ini lalu memperoleh seluruh harta karun dengan mudah.

Gambar 1. Labirin di Bawah Krusty Krab
(Sumber:https://static.wikia.nocookie.net/theloudhouse/images/e/ec/Massive_Mustard_Pocket.png/revision/latest?cb=20180826170029)

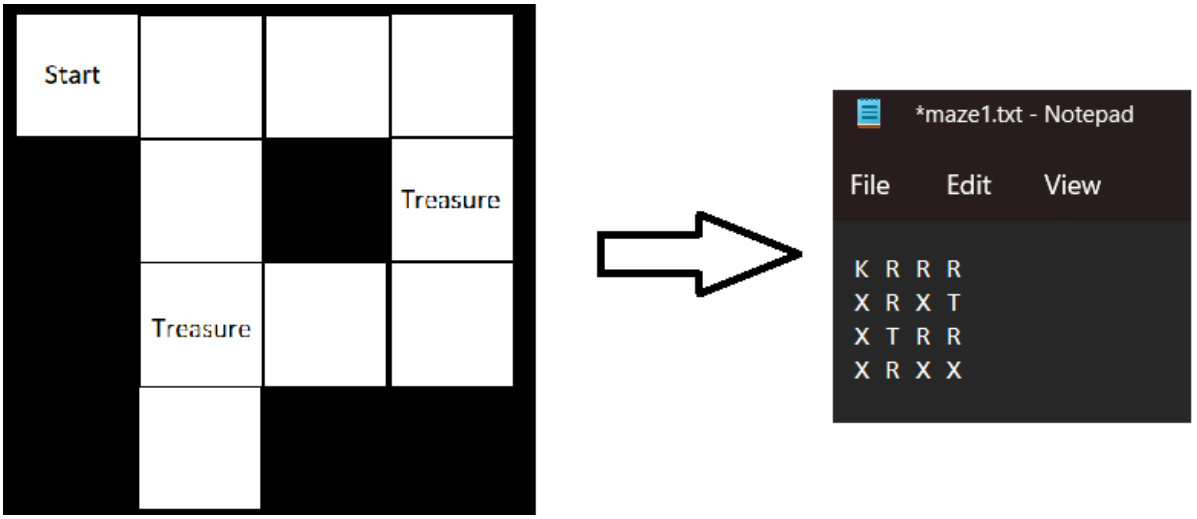
Setelah berpikir cukup lama, Tuan Krabs tiba-tiba mengingat bahwa ketika Ia berada pada kelas Strategi Algoritma-nya dulu, Ia ingat bahwa Ia dulu mempelajari algoritma BFS dan DFS sehingga Tuan Krabs menjadi yakin bahwa persoalan ini dapat diselesaikan menggunakan kedua algoritma tersebut. Akan tetapi, dikarenakan sudah lama tidak menyentuh algoritma, Tuan Krabs telah lupa bagaimana cara untuk menyelesaikan persoalan ini dan Tuan Krabs pun kebingungan. Tidak butuh waktu lama, Ia terpikirkan sebuah solusi yang brilian. Solusi tersebut adalah meminta mahasiswa yang saat ini sedang berada pada kelas Strategi Algoritma untuk menyelesaikan permasalahan ini.

Deskripsi tugas:

Dalam tugas besar ini, Anda akan diminta untuk membangun sebuah aplikasi dengan GUI sederhana yang dapat mengimplementasikan BFS dan DFS untuk mendapatkan rute memperoleh seluruh *treasure* atau harta karun yang ada. Program dapat menerima dan membaca input sebuah file txt yang berisi maze yang akan ditemukan solusi rute mendapatkan *treasure*-nya. Untuk mempermudah, batasan dari input maze cukup berbentuk segi-empat dengan spesifikasi simbol sebagai berikut :

- K : Krusty Krab (Titik awal)
- T : Treasure
- R : Grid yang mungkin diakses / sebuah lintasan
- X : Grid halangan yang tidak dapat diakses

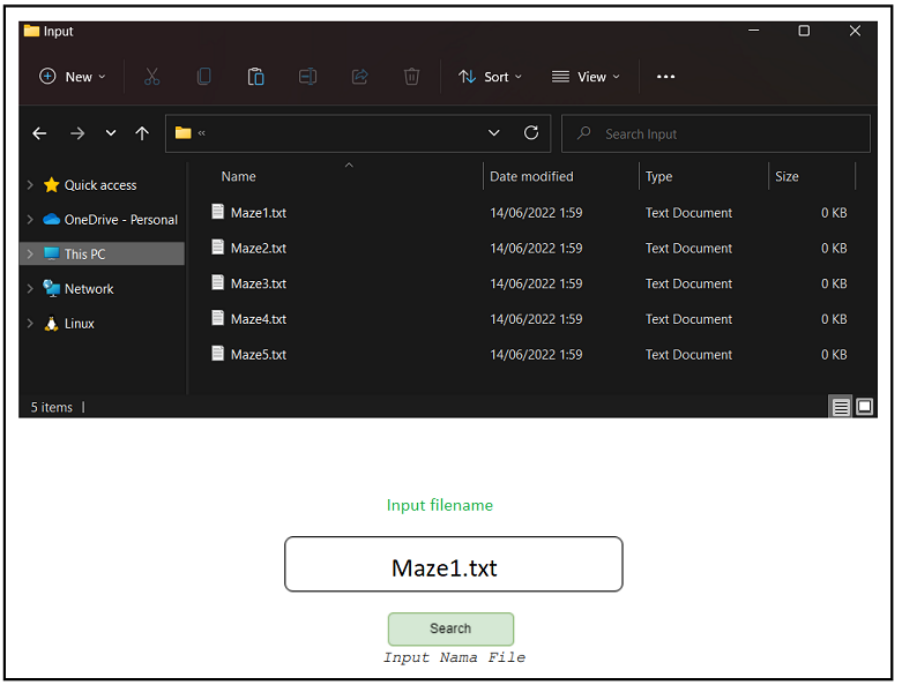
Contoh file input :



Gambar 2. Ilustrasi input file maze

Dengan memanfaatkan algoritma Breadth First Search (BFS) dan Depth First Search (DFS), anda dapat menelusuri grid (simpul) yang mungkin dikunjungi hingga ditemukan rute solusi, baik secara melebar ataupun mendalam bergantung alternatif algoritma yang dipilih. **Rute solusi adalah rute yang memperoleh seluruh treasure pada maze**. Perhatikan bahwa rute yang diperoleh dengan algoritma BFS dan DFS dapat berbeda, dan banyak langkah yang dibutuhkan pun menjadi berbeda. Prioritas arah simpul yang dibangkitkan dibebaskan asalkan ditulis di laporan ataupun readme, semisal LRUD (left right up down). **Tidak ada pergerakan secara diagonal**. Anda juga diminta untuk memvisualisasikan input txt tersebut menjadi suatu grid maze serta hasil pencarian rute solusinya. Cara visualisasi grid dibebaskan, sebagai contoh dalam bentuk matriks yang ditampilkan dalam GUI dengan keterangan berupa teks atau warna. Pemilihan warna dan maknanya dibebaskan ke masing - masing kelompok, asalkan dijelaskan di readme / laporan.

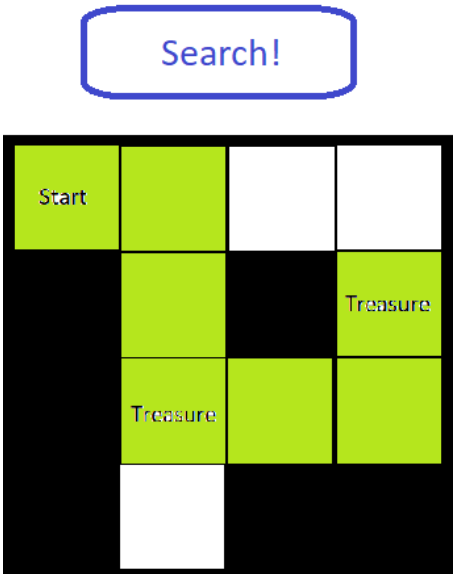
Contoh input aplikasi :



Gambar 3. Contoh input program

Daftar input maze akan dikemas dalam sebuah folder yang dinamakan test dan terkandung dalam repository program. Folder tersebut akan setara kedudukannya dengan folder src dan doc (struktur folder repository akan dijelaskan lebih lanjut di bagian bawah spesifikasi tubes). Cara input maze boleh langsung input file atau dengan textfield sehingga pengguna dapat mengetik nama maze yang diinginkan. Apabila dengan textfield, harus menghandle kasus apabila tidak ditemukan dengan nama file tersebut.

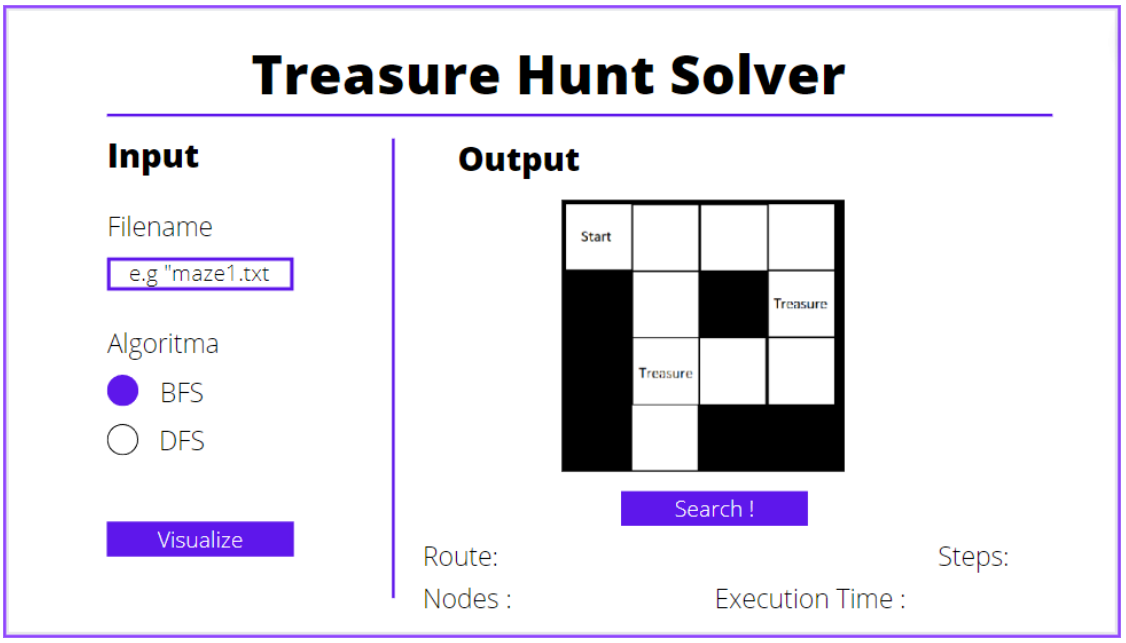
Contoh output Aplikasi :



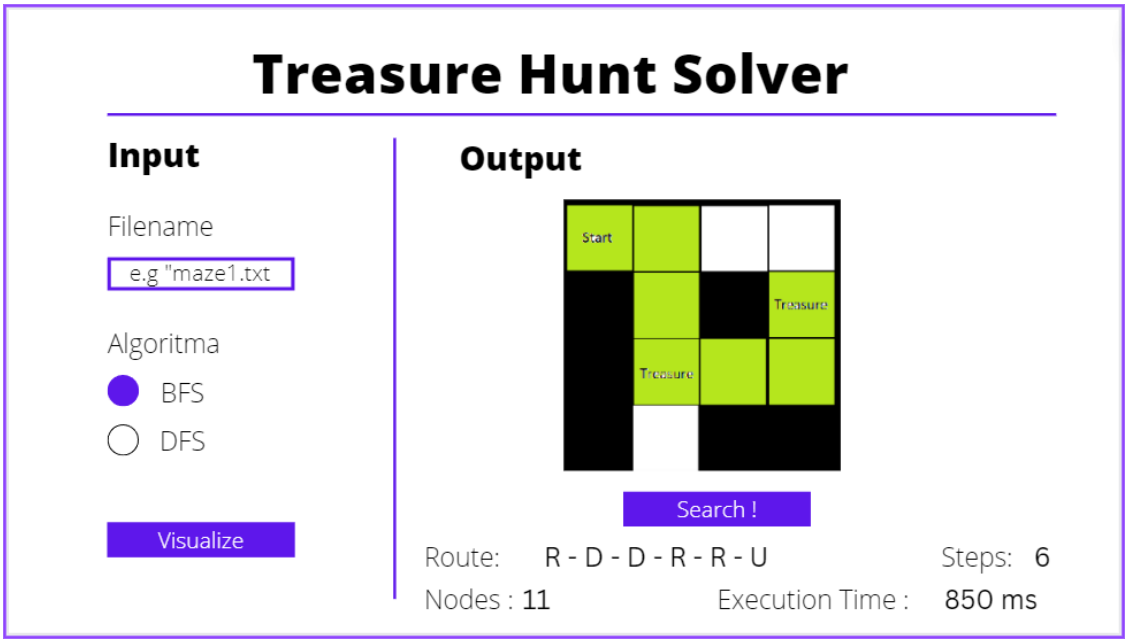
Gambar 4. Contoh output program untuk gambar 2

Setelah program melakukan pembacaan input, program akan memvisualisasikan gridnya terlebih dahulu tanpa pemberian rute solusi. Hal tersebut dilakukan agar pengguna dapat mengerjakan terlebih dahulu treasure hunt secara manual jika diinginkan. Kemudian, program menyediakan tombol solve untuk mengeksekusi algoritma DFS dan BFS. Setelah tombol diklik, program akan melakukan pemberian warna pada rute solusi.

Spesifikasi Program:
Aplikasi yang akan dibangun dibuat berbasis GUI. Berikut ini adalah contoh tampilan dari aplikasi GUI yang akan dibangun



Gambar 5. Tampilan Program Sebelum dicari solusinya



Gambar 6. Tampilan Program setelah dicari solusinya

Catatan: Tampilan diatas hanya berupa contoh layout dari aplikasi saja, untuk design layout aplikasi dibebaskan dengan syarat mengandung seluruh input dan output yang terdapat pada spesifikasi.

Spesifikasi GUI:

1. **Masukan program** adalah file maze treasure hunt tersebut atau nama filenya.
2. Program dapat menampilkan visualisasi dari input file maze dalam bentuk grid dan pewarnaan sesuai deskripsi tugas.
3. Program memiliki toggle untuk menggunakan alternatif algoritma BFS ataupun DFS.
4. Program memiliki tombol search yang dapat mengeksekusi pencarian rute dengan algoritma yang bersesuaian, kemudian memberikan warna kepada rute solusi output.
5. **Luaran program** adalah banyaknya node (grid) yang diperiksa, banyaknya langkah, rute solusinya, dan waktu eksekusi algoritma.
6. **(Bonus)** Program dapat menampilkan progress pencarian grid dengan algoritma yang bersesuaian. Hal tersebut dilakukan dengan memberikan slider / input box untuk menerima durasi jeda tiap step, kemudian memberikan warna kuning untuk tiap grid yang sudah diperiksa dan biru untuk grid yang sedang diperiksa.
7. **(Bonus)** Program membuat toggle tambahan untuk persoalan TSP. Jadi apabila toggle dinyalakan, rute solusi yang diperoleh juga harus kembali ke titik awal setelah menemukan segala harta karunnya (Tetap dengan algoritma BFS atau DFS).
8. GUI dapat dibuat **sekreatif** mungkin asalkan memuat 5 (7 jika mengerjakan bonus) spesifikasi di atas.

Program yang dibuat harus memenuhi spesifikasi wajib sebagai berikut:

1) Buatlah program dalam bahasa **C#** untuk mengimplementasi *Treasure Hunt Solver* sehingga diperoleh output yang diinginkan. Penelusuran harus memanfaatkan algoritma **BFS** dan **DFS**.
2) Awalnya program menerima file atau nama file maze treasure hunt.
3) Apabila filename tersebut ada, Program akan melakukan validasi dari file input tersebut. Validasi dilakukan dengan memeriksa apakah tiap komponen input hanya berupa K, T, R, X. Apabila validasi gagal, program akan memunculkan pesan bahwa file tidak valid. Apabila validasi berhasil, program akan menampilkan **visualisasi** awal dari maze treasure hunt.
4) Pengguna memilih algoritma yang digunakan menggunakan toggle yang tersedia.
5) Program kemudian dapat menampilkan **visualisasi** akhir dari maze (dengan pewarnaan rute solusi).
6) Program menampilkan luaran berupa durasi eksekusi, rute solusi, banyaknya langkah, serta banyaknya node yang diperiksa.

Proses visualisasi ini boleh memanfaatkan pustaka atau kakas yang tersedia. Sebagai referensi, salah satu kakas yang tersedia untuk memvisualisasikan matrix dalam bentuk grid adalah **DataGridView.** Berikut adalah panduan singkat terkait penggunaannya http://csharp.net-informations.com/datagridview/csharp-datagridview-tutorial.htm

7) Mahasiswa **tidak diperkenankan** untuk melihat atau menyalin library lain yang mungkin tersedia bebas terkait dengan pemanfaatan BFS dan DFS. Akan tetapi, untuk algoritma lain diperbolehkan menggunakan library jika ada.

Lain – lain:

1. Anda dapat menambahkan fitur-fitur lain yang menunjang program yang anda buat (unsur kreativitas).
2. Tugas dikerjakan berkelompok, minimal 2 orang dan maksimal 3 orang, boleh lintas kelas namun **tidak diperbolehkan sekelompok** dengan **orang yang sama dengan tubes ataupun tucil stima sebelumnya**
3. Semua kelompok harap mengisi data kelompok mereka pada tautan berikut: https://bit.ly/KelompokTubes2Stima
4. Program dibuat dengan bahasa C# dengan kakas Visual Studio .NET, pelajarilah C# desktop development, **disarankan untuk menggunakan Visual Studio** untuk mempermudah pengerjaan
5. Program harus modular dan mengandung komentar yang jelas.
6. Beri nama aplikasi anda tersebut dengan nama-nama yang menarik dan mudah diingat.
7. Dilarang menggunakan kode program yang diunduh dari Internet. Mahasiswa harus membuat program sendiri, tetapi belajar dari program yang sudah ada tidak dilarang.
8. Batas akhir pengumpulan tugas adalah **Jumat, 24 Maret 2023 23:59 WIB**. Keterlambatan dalam mengumpulkan akan diberi penalti pengurangan skor yang cukup signifikan.
9. Semua pertanyaan menyangkut tugas ini dapat dikomunikasikan lewat QnA yang bisa diakses pada bit.ly/TugasStimaQnA
10. Contoh test case untuk tubes ini dapat diunduh melalui tautan berikut: https://drive.google.com/drive/folders/1y-hrW6U2w5HoWdkf722yinZMOJVvLQoF?usp=share_link
11. **Bonus (nilai maksimal 5)**: Setiap kelompok membuat video aplikasi yang mereka buat kemudian mengunggahnya ke Youtube. Video yang dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Pada waktu demo aplikasi di depan asisten, mahasiswa mengakses video Youtube tersebut dan memutarnya di depan asisten. Beberapa contoh video tubes tahun-tahun sebelumnya dapat dilihat di YouTube dengan menggunakan kata kunci “Tubes Stima”, “Tugas besar stima”, “strategi algoritma”, dll.
12. Demo akan dilakukan, tunggu informasi lanjut setelah waktu pengerjaan tugas berakhir
13. Setiap anggota kelompok harus memahami seluruh program, termasuk bagian yang bukan bagian mereka
14. Program disimpan dalam folder **Tubes2_NamaKelompok**. Berikut adalah struktur dari isi folder tersebut :
    a. Folder **src** berisi source code
    b. Folder **test** berisi input file txt untuk maze
    c. Folder **bin** berisi **executable** code / hasil build dari program C#
    d. Folder **doc** berisi **laporan tugas besar** dengan format **nama_kelompok.pdf**
    e. **README** untuk tata cara penggunaan yang minimal berisi :
        i. Deskripsi singkat program yang dibuat
        ii. Requirement program dan instalasi module/package tertentu bila ada
        iii. Langkah meng-compile program jika diperlukan
        iv. Cara menggunakan program
        v. Author / identitas pembuat
    f. Catatan untuk **README agar dibuat selengkap-lengkapnya**. Anggap asisten sebagai orang awam yang tidak tahu apa-apa. Jangan sampai asisten mencari tahu sendiri bagaimana cara menjalankan program Anda.
15. Program disimpan pada repository Github yang di-private sebelum deadline dan di-public setelah deadline pengumpulan. Kelompok juga diminta untuk mengundang asisten sebagai collaborator dalam repository. Kelompok dapat melakukan submisi pengumpulan dengan cara mensubmit file laporan kalian pada tautan yang tersedia pada sheet QnA.

Isi laporan :

- **Cover**: Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo "gajah" ganesha.
- **Bab 1**: Deskripsi tugas (dapat menyalin spesifikasi tugas ini).
- **Bab 2**: Landasan Teori.
    - ○ Dasar teori (graph traversal, BFS, DFS) secara umum
    - ○ Penjelasan singkat mengenai C# desktop application development
- **Bab 3**: Landasan Teori.
    - ○ Langkah-langkah pemecahan masalah
    - ○ Proses mapping persoalan menjadi elemen-elemen algoritma BFS dan DFS
    - ○ Contoh ilustrasi kasus lain yang berbeda dengan contoh pada spesifikasi tugas
- **Bab 4**: Analisis Pemecahan Masalah.
    - ○ Implementasi program (pseudocode program utama).
    - ○ Penjelasan struktur data yang digunakan dalam program dan spesifikasi program
    - ○ Penjelasan tata cara penggunaan program (interface program, fitur-fitur yang disediakan program, dan sebagainya)
    - ○ Hasil pengujian (screenshot antarmuka program dan beberapa data uji beserta skenario pengujian). Diharapkan bahwa kelompok melakukan hasil pengujian untuk beberapa test case dengan ukuran input yang berbeda.
    - ○ Analisis dari desain solusi algoritma BFS dan DFS yang diimplementasikan pada setiap pengujian yang dilakukan. Misalnya adalah apakah strategi DFS lebih baik dari BFS pada kasus kasus tertentu, dan analisis kalian mengenai mengapa hal itu bisa terjadi.
- **Bab 5** : Kesimpulan dan Saran.
    - ○ Kesimpulan
    - ○ Saran
    - ○ Refleksi
    - ○ Tanggapan anda terkait tugas besar ini.
- Daftar Pustaka dan link menuju Repository

Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan mengikuti format pada section "Isi laporan" dengan baik dan benar.
3. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).
4. Link menuju repository wajib dituliskan.

Penilaian :

1. **Bagian 1 :** Laporan (30%)
    a. Mapping persoalan ke dalam elemen algoritma BFS dan DFS (5%)
    b. Hasil pengujian dan analisis algoritma (10%)
    c. Komponen – komponen lain dalam laporan (10%)
    d. Kesesuaian laporan dengan program (5%)
2. **Bagian 2 :** Implementasi Program (70%)
    a. Kebenaran program (30%)
    b. Pemahaman terhadap cara kerja program (25%)
    c. Kreativitas pembuatan GUI (15%)
3. **Bagian 3 :** Bonus (15%)

a. Mengimplementasikan tampilan progres pencarian solusi (5%)
b. Mengimplementasikan toggle untuk persoalan TSP (5%)
c. Membuat video demonstrasi program (5%)

--- *Selamat Mengerjakan!* ---

"Alhamdulilah dengan program ini saya mendapatkan banyak harta"
— Saul ---