**Assignment – Week 1**

Azhar Harisandi

22320011

1. **Soal**

* Construct an algorithm (flowchart) to solve sudoku!
* Bonus : Solve the following riddle and submit the code

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 3 |  |  | 7 |  |  |  |  |
| 6 |  |  | 1 | 9 | 5 |  |  |  |
|  | 9 | 8 |  |  |  |  | 6 |  |
| 8 |  |  |  | 6 |  |  |  | 3 |
| 4 |  |  | 8 |  | 3 |  |  | 1 |
| 7 |  |  |  | 2 |  |  |  | 6 |
|  | 6 |  |  |  |  | 2 | 8 |  |
|  |  |  | 4 | 1 | 9 |  |  | 5 |
|  |  |  |  | 8 |  |  | 7 | 9 |

1. **Solusi**

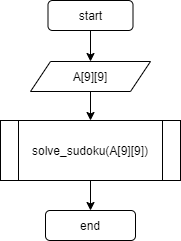
Pendekatan yang saya ajukan adalah :

1. Cari indeks baris dan kolom untuk kotak kosong, jika tidak terdapat kotak kosong maka puzzle sudah selesai.
2. Jika terdapat kotak kosong maka tebak dengan angka dari 1 hingga 9, dan cek validitasnya dengan melakukan *horizontal scanning, vertical scanning*, dan *local-grid scanning* (grid 3x3).
3. Jika memenuhi kriteria, maka tebakan akan mengisi kotak kosong tersebut, jika tidak posisi di indeks tersebut akan dikembalikan menjadi kotak kosong.
4. Panggil fungsi utama secara rekursif untuk menyelesaikan kotak kotak berikutnya. Disini terlihat bahwa jika di panggilan rekursif ke i suatu tebakan tidak memenuhi syarat atau tidak dapat menyelesaikan puzzle, panggilan rekursi ke i tersebut akan mengalami terminasi dan kembali ke panggilan rekursi ke i-1 dan seterusnya hingga ke panggilan rekursi ke n (dimana n < i) dan di saat itu tebakan akan diubah sehingga panggilan rekursi berikutnya dapat dijalankan.
5. Jika semua tebakan untuk semua sel tidak ada yang memenuhi maka puzzle tidak dapat diselesaikan.

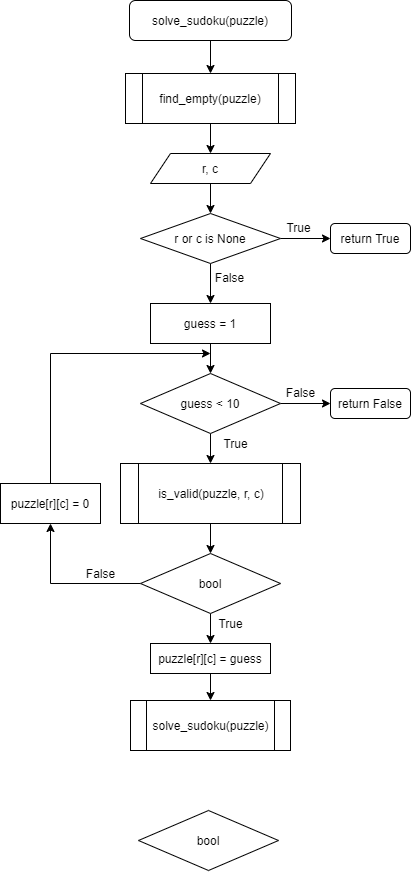
Implementasinya dalam program adalah sebagai berikut. Saya menggunakan struktur data berupa array dua dimensi, dimana dalam python diimplementasikan dalam tipe list. Untuk merepresentasikan tabel 9x9 dibutuhkan list dengan sembilan elemen yang kesembilan elemen tersebut sendiri adalah list, dan setiap elemen dari list tersebut memiliki 9 elemen lagi. Sel kosong ditandai dengan angka 0 karena nilai yang mungkin untuk mengisi puzzle hanya berada di rentang 1 sampai 9. Angka-angka di dalam list akan dimutasi sesuai dengan solusi yang dihasilkan oleh algoritma tanpa harus membuat list baru.

**Flowchart penyelesaian**

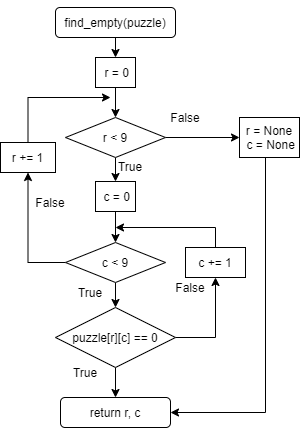
Program utama :



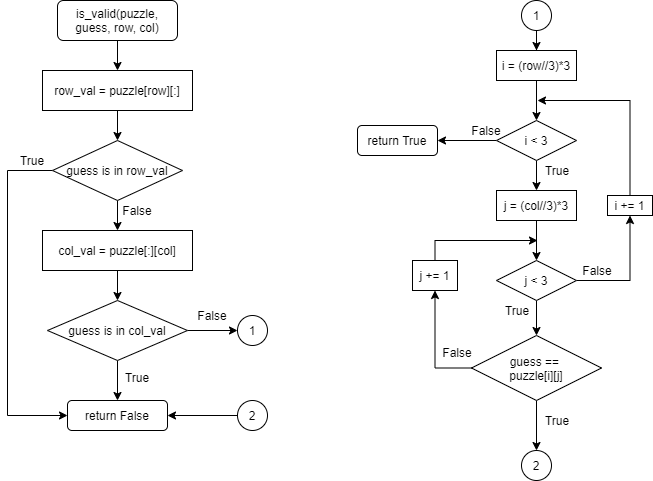
Fungsi solve\_sudoku(puzzle):



Fungsi find\_empty(puzzle):



Fungsi is\_valid(puzzle, guess, row, col)



Hasil penyelesaian puzzle

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 5 | 3 | 4 | 6 | 7 | 8 | 9 | 1 | 2 |
| 6 | 7 | 2 | 1 | 9 | 5 | 3 | 4 | 8 |
| 1 | 9 | 8 | 3 | 4 | 2 | 5 | 6 | 7 |
| 8 | 5 | 9 | 7 | 6 | 1 | 4 | 2 | 3 |
| 4 | 2 | 6 | 8 | 5 | 3 | 7 | 9 | 1 |
| 7 | 1 | 3 | 9 | 2 | 4 | 8 | 5 | 6 |
| 9 | 6 | 1 | 5 | 3 | 7 | 2 | 8 | 4 |
| 2 | 8 | 7 | 4 | 1 | 9 | 6 | 3 | 5 |
| 3 | 4 | 5 | 2 | 8 | 6 | 1 | 7 | 9 |

Cuplikan code

