

# ScrabbleBot

Tuan Vin mempunyai salah satu hobi yang sangat menarik, yaitu bermain *Scrabble*. Ketika dia masuk ITB, dia mendaftar di GSF(Ganesha Scrabble Forum). Setiap Kamis, dia berlatih bermain Scrabble dengan anggota-anggota lainnya. Namun, semua berubah ketika ~~negara api menyerang~~ kepengurusan yang lama sudah berlalu. Dia pun ditunjuk untuk menjadi ketua GSF yang baru, tetapi dia menolaknya dengan alasan bahwa dia akan keos di tingkat 2 sampai 4 di Informatika. Namun, tidak ada yang bersedia untuk mengambil jabatan bergengsi tersebut, dan pada akhirnya GSFpun dibubarkan.

Tuan Vin menjadi sangat kesal karena dia tidak mempunyai teman bermain *Scrabble* lagi, padahal waktu untuk perlombaan terdekat, 2019 Indonesian Open hanya tinggal 3,5 bulan lagi. Permasalahannya, jumlah kata dalam Scrabble mencapai ratusan ribuan, dan tidak mungkin untuk menghafalkan satu per satu kata yang ada. Teknik yang paling mungkin dilakukan baginya adalah latihan dengan orang lain. Permasalahannya, dia tidak mempunyai *partner* yang memacunya untuk berlatih lebih keras. Dia sudah mendaftar di salah satu klub Scrabble di kota Bandung, tetapi lawan terbaik di klub tersebut justru kalah 10 kali dari 12 kali bermain dengannya! Tuan Vin pun merasa bingung, dan karena lawan-lawan *top* berada pada kota yang berbeda dengannya serta masing-masing dari mereka tidak ingin diketahui permainannya oleh Tuan Vin, permintaan duel Tuan Vinpun ditolak oleh mereka semua. Kasihan Tuan Vin. Oleh karena itu, dia ingin membuat *game* ScrabbleBot yang akan membantunya dalam mengembangkan kemampuan Scrabblenya.

Kalian berpikir-pikir bagaimana cara membantu permasalahannya. Akhirnya, kalian memutuskan untuk membuat ScrabbleBot dalam *desktop application* yang akan membantunya bermain. Untuk itu, kalian membuat inti dari permainan ini, yaitu penyusunan katanya.

Spesifikasi Program:

1. Program dibuat dalam bahasa Ruby
2. Program menerima masukan dari *command line*, dengan format:

```
ruby ScrabbleBot.rb input.txt
```

dengan input.txt berisi 2 bagian. Bagian pertama adalah papan 15x15 untuk bermain dengan huruf-huruf yang sudah disediakan. Setiap petak akan dipisahkan dengan spasi. Petak dapat terdiri dari huruf 'A' sampai 'Z' dan '\_' yang menandakan petak kosong. Bagian kedua dipisahkan dengan enter, dan terdiri dari tujuh tile yang dipisahkan spasi. Setiap tile bisa terdiri dari huruf 'A' sampai 'Z' dan '?'. Bentuk input.txt adalah seperti berikut:

-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
-----
A B C D E F ?

- Setelah di-load, program menampilkan papan Scrabble dengan petak-petak khusus, seperti papan aslinya. Untuk posisi petak khusus(*Double Word*, *Double Letter*, *Triple Word* dan *Triple Letter*) dapat dilihat di Internet, dengan '\_' menandakan petak kosong non-petak khusus, '!' menandakan petak *Double Letter* yang kosong, '@' menandakan petak *Triple Letter* yang kosong, '#' menandakan petak *Double Word* yang kosong, dan '\$' menandakan petak *Triple Word* yang kosong, serta petak yang berisi dengan huruf tertentu ditampilkan huruf tersebut.

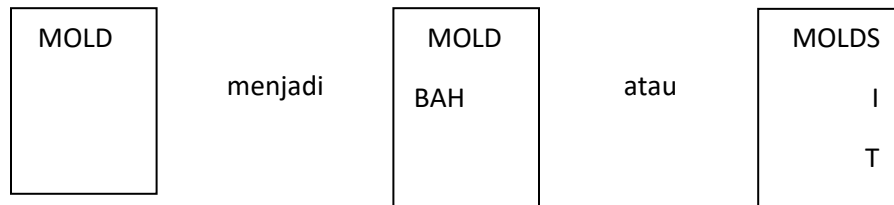
4. Program akan memasang kata-kata yang valid pada papan Scrabble dari satu atau lebih dari ketujuh huruf tersebut. Kata-kata yang valid dapat dilihat pada berkas CSW15.txt. Di dalam Scrabble, kita boleh membuat kata-kata dengan teknik sebagai berikut:

- a. Membuat kata secara horizontal atau vertikal. Misalkan



Gambar 1. Pembuatan kata secara horizontal/vertikal

- b. Menambahkan 2 atau lebih huruf ke kata yang sudah ada. Misalkan PLAY ditambah ER menjadi PLAYER.
- c. Membuat kata secara horizontal atau vertikal, dengan membuat kata yang baru. Setiap huruf yang bersinggungan pada sebuah kata harus membentuk kata Scrabble yang valid juga.



Gambar 2. Pembuatan kata dengan membentuk kata-kata baru yang bersinggungan

5. Sebelum masuk ke penilaian Scrabble, berikut ini adalah istilah-istilah yang digunakan dalam proses penilaian Scrabble:
- a. Nilai dasar, yaitu nilai total dari semua huruf tanpa adanya tambahan poin dari petak-petak khusus
- b. Petak-petak khusus, yaitu petak-petak yang bisa meningkatkan poin sesuai dengan tipenya masing-masing, yaitu:
- i. *Double Letter*, yaitu petak yang mengalikan nilai dari huruf yang berada pada petak tersebut dengan 2.

- ii. *Triple Letter*, yaitu petak yang mengalikan nilai dari huruf yang berada pada petak tersebut dengan 3.
- iii. *Double Word*, yaitu petak yang mengalikan nilai dasar dengan huruf yang berada pada petak tersebut dengan 2.
- iv. *Triple Word*, yaitu petak yang mengalikan nilai dasar dengan huruf yang berada pada petak tersebut dengan 3.

6. Perhitungan nilai dalam suatu permainan Scrabble mengikuti aturan sebagai berikut:

- a. Nilai setiap huruf dihitung sehingga didapatkan nilai dasar. Nilai dasar dapat dilihat pada tautan berikut: <https://scrabble.hasbro.com/en-us/faq>. Huruf '?' bernilai 0, meskipun bisa menggantikan huruf A-Z. Misalkan kata GO bernilai 3, sedangkan kata **G**O (huruf merah menunjukkan huruf '?') bernilai 1.
- b. Jika ada huruf yang menempati petak *Double Letter* dan/atau *Triple Letter*, huruf tersebut dikalikan sesuai dengan definisi petak tersebut. Misalkan kata GO dengan huruf 'G' pada *Triple Letter* bernilai 7, sedangkan kata **G**O dengan huruf '**G**' pada *Triple Letter* bernilai 1.
- c. Jika ada huruf yang menempati petak *Double Word* dan/atau *Triple Word*, nilai dasar kata tersebut dikalikan sesuai dengan definisi petak tersebut. Misalkan kata GO dengan huruf 'G' pada *Triple Word* bernilai 9, sedangkan kata **G**O dengan huruf '**G**' pada *Triple Word* bernilai 3.
- d. Jika ada suatu kata melalui lebih dari satu petak khusus, mekanisme perhitungannya adalah dikalikan semua sesuai dengan petaknya dengan petak *Double Letter* dan/atau *Triple Letter* dihitung terlebih dahulu, dilanjutkan dengan petak *Double Word* dan/atau *Triple Word*. Misalkan kata GO dengan huruf 'G' pada *Triple Word* dan 'O' pada *Triple Letter* bernilai  $(2+1 \times 3) \times 3 = 15$ , sedangkan kata GO dengan huruf 'O' pada *Triple Word* dan 'G' pada *Triple Letter* bernilai  $(2 \times 3 + 1) \times 3 = 21$ .
- e. Jika ada suatu kata yang terbentuk dengan proses paralel, nilai kata tersebut adalah penjumlahan nilai dari setiap kata yang terbentuk beserta dengan nilai-nilai dari petak khusus tersebut. Misalkan penambahan kata SIT pada MOLD seperti pada gambar 2 tanpa ada petak khusus menghasilkan total poin

$(3+1+1+2+1)+(1+1+1) = 11$  (bagian pertama merupakan hasil dari MOLDS, sedangkan yang kedua berasal dari SIT).

- f. Jika kata yang sudah diletakkan ditambah dengan satu atau lebih huruf sesuai dengan kaidah pembentukan kata, petak khusus yang ditempati oleh kata tersebut tidak berlaku. Misalkan penambahan kata SIT pada MOLD seperti pada gambar 2 dengan *Triple Word* pada huruf 'M' dan *Double Letter* pada 'D' menghasilkan total poin  $(3+1+1+2+1)+(1+1+1) = 11$  (bagian pertama merupakan hasil dari MOLDS, sedangkan yang kedua berasal dari SIT).
- g. Jika penambahan kata seperti pada teknik iii, dan huruf yang bersisian dengan lebih dari 1 kata merupakan petak khusus, petak khusus akan berlaku untuk setiap kata yang bersinggungan. Misalkan penambahan kata SIT pada MOLD seperti pada gambar 2 dengan *Triple Word* pada huruf 'S' menghasilkan total poin  $(3+1+1+2+1) \times 3 + (1+1+1) \times 3 = 33$  (bagian pertama merupakan hasil dari MOLDS, sedangkan yang kedua berasal dari SIT).
- h. Jika langsung ketujuh huruf yang ada pada deknya dalam suatu langkah atau *Bingo*, akan mendapatkan bonus nilai sebesar 50, dan jika ada huruf yang mengenai petak *Double Word* dan/atau *Triple Word*, bonus nilai tersebut berlaku seperti pada aturan nomor ii sampai vii.
7. Program akan menampilkan posisi awal papan ditambah kata yang dibentuk, kata yang dibentuk dan nilainya. Jika ada lebih dari 1 kata yang terbentuk, penulisan kata dilakukan secara horizontal per kata, dengan kata teratas adalah kata utama yang dibentuk, dan nilainya. Kata utama adalah kata yang sebenarnya ingin dibuat oleh program, tetapi bersentuhan dengan kata-kata lainnya. Contohnya pada MOLDS dan SIT di gambar 2, kata utamanya adalah SIT. Untuk nilainya, ditampilkan sejajar dengan kata utamanya. Contohnya pada MOLDS dan SIT akan ditampilkan:

<posisi\_papan\_akhir>

SIT <nilai\_skor>

MOLDS

sehingga secara keseluruhan tampilan program adalah sebagai berikut:

<posisi\_papan\_awal>

<posisi\_papan\_akhir>

SIT <nilai\_skor>

MOLDS

8. Kata yang dibuat harus menghasilkan nilai sebesar mungkin. Selain itu, program juga harus mencatat waktu yang dibutuhkan, dengan maksimal 5 detik.
9. Program memiliki *Readme* dalam *Markdown* yang berisi:
  - a. *Environment* pembuatan program
  - b. Fitur yang kalian buat saja dari semua fitur di bawah ini:
    - i. Pembacaan file input.txt
    - ii. Tampilan Scrabble *Board*
    - iii. Pembuatan kata
    - iv. Tampilan kata dan nilainya
    - v. GUI(bonus)

Jika ada bagian fitur yang tidak kalian kerjakan, silahkan tuliskan hal apa saja yang dikerjakan dan yang tidak dikerjakan, baik dari segi *backend* maupun tampilannya. Jika fitur tersebut tidak dikerjakan, tuliskan sebagai berikut:

<Nama Fitur> : Tidak buat

- c. Asumsi dalam pembuatan program(jika ada)
- d. Deskripsi algoritma yang digunakan
- e. Cara menggunakan program
- f. Data diri

10. Bonus(untuk yang ingin tantangan lebih): Buatlah GUI dari program ini(tidak perlu menu). Diperbolehkan menggunakan *library* luar. Silahkan ambil asset-aset yang diperlukan dari Internet.
- 11. Program harus ditulis dengan menggunakan nama-nama variabel, fungsi, dan prosedur yang dapat dimengerti. Program perlu memasukkan komentar pada beberapa bagian yang mungkin tidak mudah ditangkap dengan baik oleh saya.**
- 12. Jika memiliki asumsi, tanyakan terlebih dahulu kepada saya. Asumsi yang dituliskan dalam *Readme* adalah setiap pertanyaan yang saya jawab, “Asumsikan sendiri.”**
- 13. Penilaian tugas ini dalam bentuk fitur-fitur, sehingga jika ada yang ingin mengirimkan program yang tidak lengkap, silahkan dikumpulkan saja.**
14. Nilai maksimum pada tugas ini adalah 2800 untuk *deliverable* utama dan 1500 untuk bonus.

-----SELAMAT MENGERJAKAN-----