

# Guide Book

## Mobitangga Legends: Dor Dor

IF2111, 2021-25-11

NOTE: Dokumen ini dibuat dengan tujuan untuk memberikan panduan dan keterangan mengenai Mobitangga Legends: Dor Dor (MLDD) yang tidak bisa dijelaskan di laporan.

## Opening

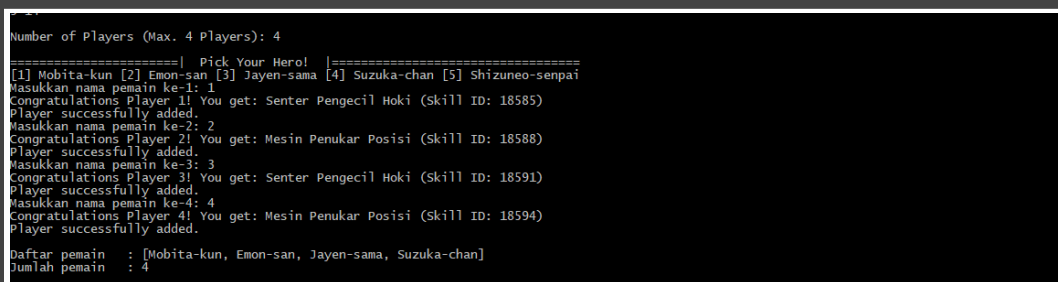
### New Game

1. User akan disambut oleh sistem dan diberikan penjelasan singkat terkait *game*.
2. User dapat memasukkan "0" untuk keluar dari *game* atau "1" untuk bermain.



Gambar 1. Opening permainan.

3. Apabila user memasukkan "1", user akan diminta untuk mengatur konfigurasi permainan.
4. Selanjutnya, pemain akan diminta untuk memasukkan jumlah pemain. Mobitangga Legends: Dor Dor (MLDD) dapat dimainkan oleh 5 orang (max).
5. Pilih *hero* yang kamu inginkan dengan memasukkan nomor urut *hero*!



Gambar 2. Proses mendaftarkan pemain.

## Load Game

```
(([0] Exit [1] New Game [2] Load Game) >>> 2
Loading data...
Player successfully added.
Player successfully added.
Player successfully added.
Player successfully added.
Player successfully added.
Player successfully added.
Congratulations Player 1! You get: Teknologi Gagal (Junk)
Congratulations Player 2! You get: Teknologi Gagal (Junk)
Congratulations Player 3! You get: Teknologi Gagal (Junk)
Congratulations Player 4! You get: Teknologi Gagal (Junk)
Congratulations Player 5! You get: Teknologi Gagal (Junk)
Data successfully loaded.
Daftar pemain : [Mobita-kun, Suzuka-chan, Emon-san, Jayen-sama, Shizuneo-senpai]
Jumlah pemain : 5
===== Round 1 |=====
Mobita-kun : .....%.....
Suzuka-chan : .....%.....
Emon-san : .....%.....
Jayen-sama : .....%.....
Shizuneo-senpai : .....%.....
[Mobita-kun] >>> .
```

Gambar 3. Tampilan permainan saat *load game* berhasil.

Jika user memasukkan angka “2”, maka sistem akan melakukan *load data* dari *state* permainan sebelumnya. Dalam hal ini, user harus memastikan bahwa ia telah melakukan seminimalnya satu kali SAVE saat pertama kali bermain. Apabila belum pernah SAVE dan langsung memilih *load game*, sistem akan memunculkan pesan bahwa data gagal di-load karena file data tidak ditemukan.

Setiap kali melakukan *load game*, permainan akan dimulai dari pemain pertama. *State* permainan yang di-load adalah nama pemain, skill pemain, dan posisi pemain. Setiap pemain akan diberikan *skill* baru secara acak (misal: sebelumnya pemain A punya 1 *skill*. Maka saat memulai permainan dengan *load game*, *skill* pemain A ditambah sehingga mempunyai 2 *skill*). *Stack* penyimpan *state* juga dimulai dari kondisi kosong dan ronde dimulai dari Ronde 1.

# Konfigurasi Permainan

- Line 1: Bilangan N yang menandakan panjang peta atau banyaknya petak pada peta.
  - $2 \leq N \leq 100$
- Line 2: String dengan N, merepresentasikan petak-petak pada peta.
  - (.) = Petak kosong (titik)
  - (#) = Petak terlarang (pagar)
- Line 3: MaxRoll, bilangan bulat yang merepresentasikan angka maksimal yang dapat dihasilkan saat ROLL.
  - $1 \leq \text{MaxRoll} \leq N - 1$
- Line 4: Bilangan M yang merepresentasikan banyaknya teleporter yang ada pada peta
  - $1 \leq M \leq N - 2$
- M baris selanjutnya menyatakan petak yang memiliki teleporter dan petak tujuan dari teleporter tersebut, bilangan bulat yang dipisahkan dengan spasi (TelIn TelOut).
  - Misal: 5 12, menandakan bahwa petak ke-5 memiliki teleporter menuju petak ke-12.
  - Petak teleporter (TelIn) dan petak tujuan (TelOut) harus berbeda, dan petak teleporter (TelIn) harus unik.

```
([0] Exit [1] New Game) >>> 1
Set The Game's Configuration!
(check our documentation for configuration format)
20
...#...#...#...#...#
10
10
3 5
5 10
6 2
8 20
13 14
10 13
18 5
17 1
16 20
9 14
```


Gambar 4. Contoh mengatur konfigurasi.

NOTE: Salin dan tempel konfigurasi di bawah ini untuk percobaan.

```
20
...#...#...#...#...#
10
10
3 5
5 10
6 2
8 20
13 14
10 13
18 5
17 1
16 20
9 14
```

# Gameplay

Saat permainan dimulai, sistem akan meminta *command* dari pemain. Pemain yang sedang bermain dapat dilihat dari nama *hero* di dalam kurung siku (lihat Gambar 4). Sesampainya di tahap ini, pemain sudah dapat memasukkan *command* untuk maju, menggunakan skill, ataupun *undo*. Permainan selesai apabila terdapat pemain yang berhasil mencapai petak terakhir pada peta. Di akhir game, sistem akan menampilkan beberapa pesan. Pemain dapat memasukkan atau memberi *input* apa saja setelah kata “Ciao!” karena di akhir program diberi **scanf** agar saat permainan selesai, program tidak langsung tertutup.



```
[Mobita-kun] >>>
```

Gambar 5. Pemain yang sedang bermain adalah Mobita-kun.

*Game* mungkin saja error saat baru pertama kali main, terutama saat *opening*. Hal tersebut wajar dan tidak masalah karena sistem mungkin masih bekerja dalam proses *compiling* atau *loading game*. Apabila terjadi, tutup dan buka kembali programnya.

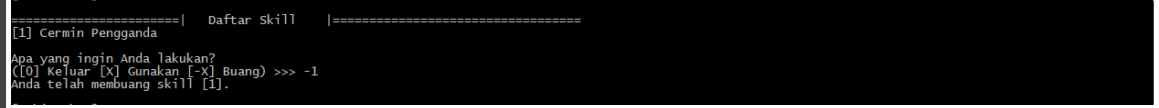
**[PERINGATAN!]** Permainan ini berasumsi bahwa masukan dari pemain selalu *valid*. Hal ini untuk mencegah terjadinya *error* atau *crash* saat permainan sedang berlangsung.

## Istilah dalam Permainan

- **ID/Turn** - Urutan pemain
- **Round** - Ronde permainan. Pergantian ronde ditandai dengan sudahnya pemain dengan urutan terakhir melakukan ENDTURN.
- **Buff** - Efek dari *skill* yang digunakan oleh pemain. Efek dari *buff* akan di-reset setiap *round*, kecuali “# Imunitas Teleport #”. *Buff* dapat di-replace oleh *buff* lain sehingga perhatikan penggunaan *skill* dan manajemen *buff* saat bermain!

## Masukan dalam Permainan

Dalam melakukan *command*, pemain akan diberikan beberapa pilihan untuk melanjutkan aksinya. Untuk memilih pilihan tersebut, pemain dapat memasukkan *key* yang ada di dalam tanda kurung siku. Khusus untuk *command* SKILL, substitusi X pada [X] dan [-X] dengan nomor *skill* (lihat Gambar 5).



```
=====|  Daftar Skill  |=====
[1] Cermin Pengganda
Apa yang ingin Anda lakukan?
([0] Keluar [X] Gunakan [-X] Buang) >>> -1
Anda telah membuang skill [1].
```

Gambar 6. Contoh input saat menggunakan SKILL