

Spesifikasi Tugas Besar

IF2121 Logika Komputasional

Global Conquest: Battle for Supremacy

Dipersiapkan Oleh Tim Asisten Lab Grafik dan Intelegensi Buatan
Versi: 1.0 08/11/2023

Topik

Membuat sebuah game strategi sebagai seorang programmer yang ingin mendominasi dunia dengan menggunakan bahasa pemrograman deklaratif Prolog (GNU Prolog).



Gambar 1. Emang boleh sedominasi ini

Implementasi tugas besar harus mengandung materi:

1. Rekurens
2. *List*
3. *Cut*
4. *Fail*
5. *Loop*

Tujuan

Tujuan dari tugas besar ini adalah mengkombinasikan berbagai keterampilan dan teknik yang telah dipelajari dalam perkuliahan Logika Komputasional IF2121, pra-praktikum, dan eksplorasi mandiri mengenai Logika Komputasional dan Prolog.

Timeline

No	Waktu	Kegiatan
1	8 November 2023	Pembahasan Pra-Praktikum, Rilis Tugas Besar IF2121
2	11 November 2023, Pukul 21.21	Batas pengisian daftar kelompok
3	15 November 2023, Pukul 21.21	Milestone 1: Asistensi Tugas Besar, pengerjaan, dan pengumpulan <i>progress</i> pertama tugas besar
4	22 November 2023, Pukul 21.21	Milestone 2: Asistensi Tugas Besar, pengerjaan, dan pengumpulan <i>progress</i> kedua dari tugas besar
5	29 November 2023, Pukul 21.21	Milestone 3: Pengumpulan <i>final deliverables</i> Tugas Besar Logika Komputasional
6	5 Desember 2023	Batas Demo Tugas Besar

Alur Permainan

1. Program diinisiasi dengan meminta *input* jumlah pemain. Permainan dimainkan oleh 2 hingga 4 pemain. Pemain lalu bergiliran melakukan *input* nama.
2. *World map* dibagi menjadi 24 wilayah yang terdiri dari 6 benua.
3. Pada awal permainan, pemain mendapatkan total jumlah tentara tergantung banyaknya pemain.
4. Pemain dapat menduduki wilayah dengan mendistribusikan tentara secara bergantian atau memilih sistem untuk mendistribusikan tentara otomatis secara acak. Distribusi tentara manual dilakukan dengan memilih wilayah yang ingin ditempati beserta jumlah tentara yang ingin ditempatkan.
5. Pada setiap giliran, pemain diberikan 1 tentara tambahan setiap 2 wilayah yang dimiliki (contoh: pemain dengan 9 wilayah akan mendapatkan 4 tentara tambahan). Jika pemain berhasil menguasai sebuah benua, maka total tentara tambahan yang diberikan akan bertambah.
6. Pada gilirannya, pemain juga bisa memindahkan tentara pada wilayah yang dimiliki, mendapatkan *risk card* secara acak, serta menyerang wilayah tetangga. Berikut merupakan daftar perintah yang bisa dipanggil pemain:
 - Draft
 - Move
 - Risk
 - Attack
 - EndTurn
7. (Draft) Meletakkan tentara tambahan yang didapatkan oleh pemain pada awal giliran. Dilakukan dengan memilih wilayah tujuan dan jumlah tentara yang ingin dipindah. Peletakan tentara dapat dilakukan hingga semua tentara tambahan sudah diletakkan pada wilayah.
8. (Move) Pemindahan tentara dilakukan dengan memilih wilayah asal, jumlah tentara yang ingin dipindah, serta wilayah tujuan. Pemindahan tentara wajib menyisakan 1

tentara untuk menjaga wilayah yang dimiliki. Pemindahan tentara dapat dilakukan secara tak terbatas pada setiap giliran.

9. (Risk) *Risk card* didapatkan dengan melempar dadu dan bersifat opsional. Pada setiap giliran, pemain hanya memiliki 1 kesempatan untuk mendapatkan *risk card*.
10. (Attack) Pemain dapat menyerang dengan memilih wilayah penyerang, menentukan jumlah tentara penyerang, dan memilih wilayah tetangga yang akan diserang. Hasil pertempuran antara 2 wilayah ditentukan dengan lemparan dadu. Penyerangan hanya dapat dilakukan sekali setiap giliran.
11. (EndTurn) Giliran pemain diakhiri setelah perintah EndTurn dipanggil.
12. Pemain yang berhasil menguasai setiap wilayah pada papan dan mengeliminasi semua lawan menjadi pemenang.

Spesifikasi

1. Map

World map terdiri dari 24 wilayah dan 6 benua. Implementasi world map dapat memanfaatkan struktur data undirected graph, dimana node merupakan wilayah dan edge merupakan hubungan ketetanggaan antara dua wilayah. Pemain dapat menggunakan perintah displayMap untuk melihat kondisi peta saat ini. Implementasi bentuk map dibebaskan sesuai kreativitas masing-masing dengan syarat dan ketentuan berikut:

- Terdapat total 24 wilayah dan 6 benua dengan rincian pembagian wilayah sebagai berikut:
 - Amerika Utara : 5 wilayah
 - Eropa : 5 wilayah
 - Asia : 7 wilayah
 - Amerika Selatan : 2 wilayah
 - Afrika : 3 wilayah
 - Australia : 2 wilayah
- Batasan antar benua terlihat jelas.
- Lokasi benua dari suatu wilayah terlihat dengan jelas.
- Keterhubungan antar wilayah terlihat dengan jelas.
- Setiap wilayah harus menampilkan kode wilayah dan jumlah tentara yang berada di wilayah tersebut.

Map wajib ditampilkan setiap kali awal mulai giliran suatu pemain. Diperbolehkan untuk menampilkan map secara otomatis setiap kali pemain mulai melakukan suatu aksi untuk mempermudah dalam memilih wilayah yang akan dikenakan aksi. Mekanisme aksi yang dapat dilakukan pemain dalam suatu giliran dijelaskan pada bagian berikutnya.

[illegible]

- [SA1 (11)] berarti wilayah dengan kode SA1 memiliki 11 tentara
- [A6 (1)] -- [A7 (2)] berarti wilayah A6 bertetangga dengan wilayah A7
- ##### menandakan batas antar benua
- Adanya garis yang keluar dari map di ujung kanan dan ujung kiri pada baris yang sama menandakan garis tersebut terhubung
- + digunakan untuk menyatakan persimpangan/cabang

2. Initiating

- a. Ketika permainan dimulai, akan ditanyakan terlebih dahulu mengenai jumlah dan identitas pemain. Permainan dapat dimainkan oleh **minimal 2** orang dan **maksimal 4** orang. Setelah jumlah pemain ditentukan, setiap pemain dapat **memasukkan namanya** ke dalam permainan.
- b. Setelah menginisiasi pemain, program selanjutnya akan menginisiasi *map* permainan. Lihat lebih lanjut mengenai ketentuan *map* pada bagian *Map*.
- c. Untuk memulai permainan, program akan secara otomatis **melemparkan 2 buah dadu** untuk setiap pemain. Pemain dengan gulungan dadu terbesar akan jalan terlebih dahulu, pemain berikutnya yang jalan akan disesuaikan dengan urutan pemain yang memasukkan nama pada tahap inisiasi pemain. (Sebagai contoh, pada tahap inisiasi, urutan pemasukan nama adalah A - B - C - D. Kemudian C mendapatkan gulungan dadu terbesar. Maka urutan giliran pemain adalah C - D - A - B). Jika terdapat dua atau lebih pemain yang mendapatkan gulungan dadu terbesar, maka lemparan dadu akan diulang kembali untuk setiap pemain.
- d. Selanjutnya akan dilakukan **distribusi tentara**, dimana permainan akan **memberikan sejumlah tentara** awal kepada setiap pemain bergantung pada jumlah pemain yang ada. Berikut adalah ketentuannya:
 1. Terdapat 2 pemain: Setiap pemain mendapatkan 24 tentara
 2. Terdapat 3 pemain: Setiap pemain mendapatkan 16 tentara
 3. Terdapat 4 pemain: Setiap pemain mendapatkan 12 tentara
- e. Terakhir, program akan meminta pemain untuk **membagikan seluruh tentaranya** ke wilayah-wilayah yang ada. Ketentuan dan mekanisme pembagian adalah sebagai berikut:
 1. Pemain akan secara bergantian melakukan pembagian tentara ke wilayah yang ada sesuai dengan urutan giliran pemain.
 2. Pada fase pertama pembagian, setiap pemain secara bergantian memilih wilayah yang akan dikuasainya, hingga pada akhir fase setiap pemain mendapatkan jumlah wilayah yang seragam.
 - Pemain hanya boleh memilih wilayah kosong untuk dikuasai.
 - Pemain tidak boleh lagi memilih wilayah yang sudah dikuasai baik oleh pemain tersebut atau pemain lain.

Pada akhir fase pertama, letakkan 1 tentara milik pemain ke setiap wilayah yang dikuasainya.

3. Pada fase kedua, setiap pemain akan membagikan sisa tentaranya ke wilayah miliknya. Pemain dapat memilih untuk melakukan antara 2 hal berikut:
 - Pemain **secara manual melakukan pembagian** dengan memilih wilayah dan menentukan jumlah tentara untuk diletakkan pada wilayah tersebut. Jumlah tentara yang diletakkan berkisar antara 1 - jumlah tentara yang belum diletakkan (lakukan validasi input dan *error handling*).
 - i. Pembagian akan menambahkan jumlah tentara di wilayah tersebut.
 - ii. Pemain tidak boleh menambahkan tentara ke wilayah yang bukan miliknya.
 - Pemain meminta program untuk **melakukan pembagian secara otomatis**. Dalam hal ini, maka program membangkitkan wilayah dan jumlah tentara secara acak. Ketentuan pembagian sama seperti poin a. Pembagian dilakukan hingga seluruh tentara pemain tersebut diletakkan pada *map*.
4. Setelah setiap pemain sudah membagikan sisa tentaranya. Permainan dimulai dengan urutan yang sama dengan urutan giliran. Baca spesifikasi Turn untuk lebih lanjut.

Contoh Program

```
| ?- startGame.                                /* command dijalankan untuk memulai permainan */

Masukkan jumlah pemain: 5.
Mohon masukkan angka antara 2 - 4.
Masukkan jumlah pemain: 3.

Masukkan nama pemain 1: Dillon.
Masukkan nama pemain 2: Bagas.
Masukkan nama pemain 3: Fio.

Dillon melempar dadu dan mendapatkan 11.
Bagas melempar dadu dan mendapatkan 9.
Fio melempar dadu dan mendapatkan 8.

Urutan pemain: Dillon - Bagas - Fio
Dillon dapat mulai terlebih dahulu.

Setiap pemain mendapatkan 16 tentara.

Giliran Dillon untuk memilih wilayahnya.

(69 ms) yes

| ?- takeLocation(af3).                        /* command dijalankan untuk mengambil wilayah */

Dillon mengambil wilayah AF3.
```


Giliran Bagas untuk memilih wilayahnya.

(69 ms) yes

| ?- takeLocation(af3).

Wilayah sudah dikuasai. Tidak bisa mengambil.

Giliran Bagas untuk memilih wilayahnya.

(69 ms) yes

| ?- takeLocation(e1).

Bagas mengambil wilayah E1.

Giliran Fio untuk memilih wilayahnya.

(69 ms) yes

(dan seterusnya..)

/* pengambilan wilayah terakhir */

| ?- takeLocation(sa2).

Fio mengambil wilayah SA2.

Seluruh wilayah telah diambil pemain.

Memulai pembagian sisa tentara.

Giliran Dillon untuk meletakkan tentaranya.

(69 ms) yes

/* Lanjut ke pembagian sisa tentara */

| ?- placeTroops(af3, 3). /* command dijalankan untuk meletakkan tentara */

Dillon meletakkan 3 tentara di wilayah AF3.

Terdapat 5 tentara yang tersisa.

Giliran Dillon untuk meletakkan tentaranya.

(69 ms) yes

| ?- placeAutomatic.

Dillon meletakkan 2 tentara di wilayah A1.

Dillon meletakkan 3 tentara di wilayah A4.

Seluruh tentara Dillon sudah diletakkan.

Giliran Bagas untuk meletakkan tentaranya.

(69 ms) yes

| ?- placeAutomatic. /* command otomatis meletakkan tentara secara acak */

Bagas meletakkan 2 tentara di wilayah E1.
Bagas meletakkan 5 tentara di wilayah AU2.
Bagas meletakkan 1 tentara di wilayah A5.
Seluruh tentara Bagas sudah diletakkan.

Giliran Fio untuk meletakkan tentaranya.

(69 ms) yes

| ?- placeTroops(af3, 1).

Wilayah tersebut dimiliki pemain lain.
Silahkan pilih wilayah lain.

Giliran Fio untuk meletakkan tentaranya.

(69 ms) yes

| ?- placeTroops(sa2, 5).

Fio meletakkan 5 tentara di wilayah SA2.
Terdapat 3 tentara yang tersisa.

Giliran Fio untuk meletakkan tentaranya.

(69 ms) yes

(dan seterusnya..)

/* Setelah semua pemain meletakkan sisa tentara, maka permainan dimulai */

| ?- placeTroops(au2, 3).

Fio meletakkan 3 tentara di wilayah AU2.
Seluruh tentara Fio sudah diletakkan.

Seluruh pemain telah meletakkan sisa tentara.
Memulai permainan.

Sekarang giliran Player Bagas!
Player Bagas mendapatkan...

/* Mekanisme giliran */

3. Turn

Pada setiap *turn* atau putaran permainan *Risk*, pemain dapat memanggil perintah *draft*, *move*, *risk*, *attack*, dan *end turn*.

a. End Turn

Perintah `endTurn` digunakan untuk mengakhiri giliran pemain dan dilanjutkan ke pemain selanjutnya. Setiap awal giliran, pemain akan mendapatkan tentara tambahan. Pemain diberikan 1 tentara tambahan setiap 2 wilayah yang dimiliki (contoh: pemain dengan 9 wilayah akan mendapatkan 4 tentara tambahan) ditambah dengan bonus dari benua yang dimiliki oleh pemain. Bonus tentara dari setiap benua dibebaskan, namun dapat menggunakan list berikut sebagai referensi.

- Amerika Utara : 3 tentara
- Eropa : 3 tentara
- Asia : 5 tentara
- Amerika Selatan : 2 tentara
- Afrika : 2 tentara
- Australia : 1 tentara

Contoh Program

```
/* Player Bagas memanggil endTurn lalu giliran berlanjut ke Player Fio */
/* Player Fio memiliki 4 wilayah dan menguasai benua Afrika */

| ?- endTurn.

Player Bagas mengakhiri giliran.

Sekarang giliran Player Fio!
Player Fio mendapatkan 4 tentara tambahan.

(69 ms) yes
```

b. Draft

Tentara tambahan dapat diletakkan di wilayah teritori milik pemain dengan query `draft(X, Y)` dimana `X` merupakan wilayah teritori dan `Y` merupakan jumlah tentara yang ingin diletakkan. Lakukan validasi apakah peletakkan tentara di teritori tersebut valid.

Contoh Program
<pre> /* Kondisi awal: AU1 merupakan wilayah Player Fio */ /* terdapat 1 tentara awal di AU1 */ /* Player Fio memiliki 4 tentara tambahan */ ?- draft(aul, 3). Player Fio meletakkan 3 tentara tambahan di AU1 Tentara total di AU1: 4 Jumlah Pasukan Tambahan Player Fio: 1 (69 ms) yes </pre>
<pre> /* Meletakkan tentara di wilayah pemain lain */ ?- draft(a1, 3). Player Fio tidak memiliki wilayah A1 (69 ms) yes </pre>
<pre> /* Pasukan yang diletakkan tidak mencukupi */ ?- draft(aul, 5). Player Fio meletakkan 5 tentara tambahan di AU1 Pasukan tidak mencukupi. Jumlah Pasukan Tambahan Player Fio: 4 draft dibatalkan. (69 ms) yes </pre>

c. Move

Pemain dapat memindahkan tentara yang sudah ditempatkan di wilayahnya. Pemindahan tentara dilakukan dengan query `move(X1, X2, Y)` dimana `X1` merupakan wilayah asal, `X2` merupakan wilayah tujuan, dan `Y` merupakan jumlah tentara yang ingin dipindah. Pemindahan tentara wajib menyisakan 1 tentara untuk menjaga wilayah yang dimiliki. Perlu dilakukan validasi untuk memastikan bahwa `X1` dan `X2` merupakan wilayah pemain serta `Y` merupakan jumlah tentara yang valid dan bisa dipindahkan. **Pemindahan tentara dapat dilakukan maksimum 3 kali per turn.**

Contoh Program

```
/* Kondisi awal: AU1 dan AU2 merupakan wilayah Fio */
/*          terdapat 4 tentara di AU1 dan 2 tentara di AU2 */

| ?- move(au1, au2, 3).

Fio memindahkan 3 tentara dari AU1 ke AU2.

Jumlah tentara di AU1: 1
Jumlah tentara di AU2: 5

(69 ms) yes
```

```
/* Jumlah tentara tidak valid */

| ?- move(au1, au2, 4).

Fio memindahkan 4 tentara dari AU1 ke AU2.

Tentara tidak mencukupi.
pemindahan dibatalkan.

(69 ms) yes
```

```
/* Wilayah tidak valid */

| ?- move(a1, a2, 3).

Fio tidak memiliki wilayah A1.
pemindahan dibatalkan.

(69 ms) yes
```

d. Attack

Fungsi *attack* digunakan oleh pemain untuk melakukan penyerangan kepada wilayah pemain lain. Tujuannya dilakukan penyerangan ini untuk merebut wilayah pemain lain menjadi wilayah kita. **Attack hanya dapat dilakukan satu kali tiap turn pemain.** Terdapat beberapa mekanisme yang perlu diperhatikan dalam melakukan penyerangan.

1. *Attack* dimulai dengan pemain memilih salah satu wilayah yang dimilikinya **dan** terdapat wilayah pemain lain di sekitarnya.
2. Wilayah yang bisa dipilih untuk memulai penyerangan adalah wilayah dengan jumlah tentara lebih dari satu. Perlu diketahui bahwa harus terdapat satu tentara yang tetap menjaga wilayah agar wilayah tetap dikuasai oleh pemain yang

menyerang jika seluruh tentara lainnya yang berperang gugur atau berpindah tempat.

3. Wilayah yang bisa diserang adalah wilayah yang bersebelahan secara langsung dengan wilayah mulai.
4. Pemain memilih salah satu wilayah yang bisa diserang.
5. Pemain memilih banyaknya tentara yang dikirim untuk berperang (ingat bahwa maksimal tentara yang dikirim adalah jumlah tentara pada wilayah tersebut - 1)
6. Proses perang dilakukan dengan adu pelemparan dadu.
7. Banyaknya dadu yang dilempar bergantung kepada jumlah tentara yang berperang. Untuk pemain yang diserang, akan selalu melempar dadu sebanyak jumlah tentara yang berada pada wilayah tersebut.
8. Pemain penyerang dan pemain diserang melempar dadu secara bersamaan. Pemain yang menyerang dikatakan menang jika jumlah total angka seluruh dadu miliknya **LEBIH BESAR (>)** dari jumlah total angka seluruh dadu milik yang diserang.
9. Jika pemain penyerang kalah (yang diserang menang), maka seluruh tentara yang dikirim akan hangus dan lanjut ke fase berikutnya.
10. Jika pemain penyerang menang, maka seluruh tentara milik musuh hangus. Pemain dapat memilih berapa banyak tentara yang akan berpindah ke wilayah tersebut (minimal 1 dan maksimal adalah jumlah tentara yang berperang) dan lanjut ke fase berikutnya.

Contoh pemanggilan perintah (**HANYA CONTOH**):

Contoh Program

```
/* Kasus saat pemain penyerang menang. */  
  
| ?- attack.  
  
Sekarang giliran Player Dillon menyerang.  
  
/* PETA */  
  
Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan: A1.  
  
Player Dillon ingin memulai penyerangan dari daerah A1.  
Dalam daerah A1, Anda memiliki sebanyak 8 tentara.  
  
Masukkan banyak tentara yang akan bertempur: 6.  
Player Dillon mengirim sebanyak 6 tentara untuk berperang.
```

```
/* PETA */

Pilihlah daerah yang ingin Anda serang.
1. E2
2. A4

Pilih: 1.

Perang telah dimulai.
Player Dillon
Dadu 1: 6.
Dadu 2: 3.
Dadu 3: 2.
Dadu 4: 3.
Dadu 5: 1.
Dadu 6: 4.
Total: 19.

Player Bagus
Dadu 1: 5.
Dadu 2: 3.
Dadu 3: 2.
Dadu 4: 6.
Total: 16.

Player Dillon menang! Wilayah E2 sekarang dikuasai oleh Player Dillon.
Silahkan tentukan banyaknya tentara yang menetap di wilayah E2: 5.

Tentara di wilayah A1: 3.
Tentara di wilayah E2: 5.

(69 ms) yes
```

```
/* Kasus saat pemain penyerang kalah */

/* Mengikuti alur kasus sebelumnya */

Perang telah dimulai.
Player Dillon
Dadu 1: 6.
Dadu 2: 3.
Dadu 3: 2.
Dadu 4: 3.
Dadu 5: 1.
Dadu 6: 4.
Total: 19.

Player Bagus
Dadu 1: 5.
Dadu 2: 6.
Dadu 3: 5.
Dadu 4: 6.
Total: 22.

Player Bagus menang! Sayang sekali penyerangan Anda gagal :(.

```

Tentara di wilayah A1: 1.
Tentara di wilayah E2: 4.

(69 ms) yes

/* Daerah penyerangan tidak valid */

| ?- attack.

Sekarang giliran Player Dillon menyerang.

/* PETA */

Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan: A10.

Daerah tidak valid. Silahkan input kembali.

Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan:

/* Banyak tentara tidak valid */

/* Mengikuti alur kasus sebelumnya */

Player Dillon ingin memulai penyerangan dari daerah A1.
Dalam daerah A1, Anda memiliki sebanyak 8 tentara.

Masukkan banyak tentara yang akan bertempur: 9.

Banyak tentara tidak valid. Silahkan input kembali.

Masukkan banyak tentara yang akan bertempur:

/* Daerah yang diserang tidak valid */

/* Mengikuti alur kasus sebelumnya */

/* PETA */

Pilihlah daerah yang ingin Anda serang.

1. E2
2. A4

Pilih: 5.

Input tidak valid. Silahkan input kembali.

Pilih:

Perlu diingat bahwa dalam fase penyerangan ini juga terdapat konsep *risk*, dimana distribusi penempatan banyaknya tentara pada seluruh wilayah itu perlu diperhatikan. Contoh, jika kita memasang terlalu banyak tentara pada suatu wilayah, tentu probabilitas kemenangannya menjadi tinggi saat menyerang, namun

untuk wilayah kita yang tentaranya sedikit memiliki probabilitas bertahan yang rendah.

e. Risk

Pada gilirannya pemain dapat memanggil perintah *risk* untuk mendapatkan *risk card* secara acak. Perintah *risk* bersifat opsional dan pemain hanya dapat **memanggilnya sekali tiap giliran**. Berikut merupakan jenis *risk* yang wajib diimplementasikan:

- i. **Ceasefire Order:** Perintah gencatan senjata. Pemain memerintahkan seluruh lawan untuk melakukan gencatan senjata. Hingga giliran berikutnya, wilayah pemain tidak dapat diserang oleh lawan.
- ii. **Super Soldier Serum:** Serum tentara super. Seluruh tentara pemain mendapatkan serum peningkatan. Hingga giliran berikutnya, semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 6.
- iii. **Auxiliary Troops:** Tentara tambahan. Pemain mendapatkan bantuan tentara tambahan dari Perserikatan Bangsa. Pada giliran berikutnya, tentara tambahan yang didapatkan pemain akan bernilai 2 kali lipat.
- iv. **Rebellion:** Pemberontakan. Terdapat wilayah pemain yang tentaranya memilih untuk bergabung dengan pihak lawan. Salah satu wilayah acak pemain akan berpindah kekuasaan menjadi milik lawan (acak).
- v. **Disease Outbreak:** Wabah penyakit. Seluruh wilayah pemain akan dilanda wabah. Semua hasil lemparan dadu saat penyerangan dan pertahanan hingga giliran berikutnya akan bernilai 1.
- vi. **Supply Chain Issue:** Masalah rantai pasokan. Terdapat masalah pada pengiriman tentara tambahan. Pemain tidak mendapatkan tentara tambahan pada giliran berikutnya.

Contoh Program

```
| ?- risk.
```

```
Player Fio mendapatkan risk card CEASEFIRE ORDER.  
Hingga giliran berikutnya, wilayah pemain tidak dapat diserang oleh lawan.
```

```
(69 ms) yes
```

```
/* Player Bagas melakukan attack pada wilayah Player Fio */  
  
| ?- attack.  
  
/* Mekanisme Attack */  
  
Pilihlah daerah yang ingin Anda serang.  
1. A2  
2. AU1 /* wilayah Player Fio */  
  
Pilih: 2.  
  
Tidak bisa menyerang!  
Wilayah ini dalam pengaruh CEASEFIRE ORDER.  
  
Pilih:
```

Contoh Program

```
| ?- risk.  
  
Player Fio mendapatkan risk card SUPER SOLDIER SERUM.  
Hingga giliran berikutnya,  
semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 6.  
  
(69 ms) yes
```

```
/* Player Fio Menyerang dalam Pengaruh SUPER SOLDIER SERUM */  
  
| ?- attack.  
  
/* Mekanisme Attack */  
/* Player Fio (2 tentara) melawan Player Bagas (4 Tentara) */  
  
Tentara Fio dalam pengaruh SUPER SOLDIER SERUM.  
  
Perang telah dimulai.  
Player Fio  
Dadu 1: 6.  
Dadu 2: 6.  
Total: 12.  
  
Player Bagas  
Dadu 1: 3.  
Dadu 2: 3.  
Dadu 3: 2.  
Total: 8.  
  
Player Fio menang!  
/* Lanjutan mekanisme Attack */
```

```
/* Player Fio Diserang dalam Pengaruh SUPER SOLDIER SERUM */  
  
/* Mekanisme Attack */  
/* Player Bagas (3 tentara) melawan Player Fio (1 Tentara) */
```

```
Tentara Fio dalam pengaruh SUPER SOLDIER SERUM.
```

```
Perang telah dimulai.
```

```
Player Bagas
```

```
Dadu 1: 2.
```

```
Dadu 2: 1.
```

```
Dadu 3: 2.
```

```
Total: 5.
```

```
Player Fio
```

```
Dadu 1: 6.
```

```
Total: 6.
```

```
Player Fio menang!
```

```
/* Lanjutan mekanisme Attack */
```

Contoh Program

```
| ?- risk.
```

```
Player Fio mendapatkan risk card AUXILIARY TROOPS.
```

```
Pada giliran berikutnya,
```

```
Tentara tambahan yang didapatkan pemain akan bernilai 2 kali lipat.
```

```
(69 ms) yes
```

```
/* Giliran Player Fio berikutnya */
```

```
/* Player Fio mendapat 8 tentara tambahan dalam keadaan normal */
```

```
| ?- endTurn.
```

```
Player Bagas mengakhiri giliran.
```

```
Sekarang giliran Player Fio!
```

```
Player Fio mendapatkan AUXILIARY TROOPS!
```

```
Player Fio mendapatkan 16 tentara tambahan.
```

```
(69 ms) yes
```

Contoh Program

```
| ?- risk.
```

```
Player Fio mendapatkan risk card REBELLION.
```

```
Salah satu wilayah acak pemain akan berpindah kekuasaan menjadi milik lawan.
```

```
Wilayah B2 sekarang dikuasai oleh Player Bagas.
```

```
(69 ms) yes
```

Contoh Program

```
| ?- risk.
```

Player Fio mendapatkan risk card DISEASE OUTBREAK.
Hingga giliran berikutnya,
semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 1.

```
(69 ms) yes
```

```
/* Player Fio Menyerang dalam Pengaruh DISEASE OUTBREAK */
```

```
| ?- attack
```

```
/* Mekanisme Attack */
```

```
/* Player Fio (3 tentara) melawan Player Dillon (1 Tentara) */
```

Tentara Fio dalam pengaruh DISEASE OUTBREAK.

Perang telah dimulai.

Player Fio

Dadu 1: 1.

Dadu 2: 1.

Dadu 3: 1.

Total: 3.

Player Dillon

Dadu 1: 4.

Total: 4.

Player Dillon menang!

```
/* Lanjutan mekanisme Attack */
```

```
/* Player Fio Diserang dalam Pengaruh DISEASE OUTBREAK */
```

```
/* Mekanisme Attack */
```

```
/* Player Dillon (1 tentara) melawan Player Fio (3 Tentara) */
```

Tentara Fio dalam pengaruh DISEASE OUTBREAK.

Perang telah dimulai.

Player Dillon

Dadu 1: 5.

Total: 5.

Player Fio

Dadu 1: 1.

Dadu 2: 1.

Dadu 3: 1.

Total: 3.

Player Dillon menang!

```
/* Lanjutan mekanisme Attack */
```

Contoh Program

```
| ?- risk.
```

```
Player Fio mendapatkan risk card SUPPLY CHAIN ISSUE.  
Pada giliran berikutnya, pemain tidak mendapatkan tentara tambahan.
```

```
(69 ms) yes
```

```
/* Giliran Player Fio berikutnya */
```

```
| ?- endTurn.
```

```
Player Bagas mengakhiri giliran.
```

```
Sekarang giliran Player Fio!  
Player Fio terdampak SUPPLY CHAIN ISSUE!
```

```
Player Fio tidak mendapatkan tentara tambahan.
```

```
(69 ms) yes
```

4. Wilayah

Terdapat fitur untuk memeriksa detail dari setiap wilayah. **Detail minimal** berisi kode, nama, pemilik, jumlah tentara, dan tetangga. Berikut adalah contoh programnya.

Contoh Program	
<code>checkLocationDetail(x).</code>	
<pre> ?- checkLocationDetail(na1). /* melihat detail wilayah NA1 */ Kode : NA1 Nama : Greenland Pemilik : Bagas Total Tentara : 4 Tetangga : Mexico, Kanada, Amerika Serikat (69 ms) yes</pre>	

5. Player

1. Terdapat fitur untuk mengetahui kondisi pemain. Kondisi yang perlu dicantumkan pada fitur ini antara lain nama pemain, benua yang dikuasai, total wilayah, total tentara aktif, dan total tentara tambahan.

Contoh Program	
<code>checkPlayerDetail(x).</code>	
<pre> ?- checkPlayerDetail(p2). /* melihat detail player P2 */ PLAYER P2 Nama : Fio Benua : Afrika, Asia, Australia Total Wilayah : 13 Total Tentara Aktif : 23 Total Tentara Tambahan: 0 (69 ms) yes</pre>	

2. Terdapat fitur yang dapat melakukan highlight teritori yang dikuasai oleh seorang pemain.

Contoh Program

```
checkPlayerTerritories(x).
```

```
| ?- checkPlayerTerritories(p3).    /* melihat detail player P3 */
Nama                               : Bagas

Benua Amerika Selatan (2/2)
SA1
Nama                               : Brazil
Jumlah tentara                     : 11

SA2
Nama                               : Argentina
Jumlah tentara                     : 1

Benua Amerika Utara (2/5)
NA2
Nama                               : Amerika Serikat
Jumlah tentara                     : 3

NA3
Nama                               : Mexico
Jumlah tentara                     : 5

(69 ms) yes
```

3. Pemain juga dapat memeriksa jumlah tentara yang akan masuk pada giliran selanjutnya.

Contoh Program

```
checkIncomingTroops(x).
```

```
| ?- checkIncomingTroops(p2).    /* melihat detail player P2 */

Nama                               : Fio
Total wilayah                     : 13
Jumlah tentara tambahan dari wilayah : 6
Bonus benua australia             : 2
Bonus benua asia                  : 5
Bonus benua afrika                : 2
Total tentara tambahan             : 15

(69 ms) yes
```

4. Mekanisme kalah

- a. Jika seorang pemain diserang hingga jumlah wilayahnya 0, maka pemain tersebut keluar dari permainan. Seluruh aset milik pemain tersebut seperti tentara tambahan akan hangus.

- b. Seorang pemain memenangkan permainan ketika dia mengontrol seluruh wilayah yang ada di map.

Contoh Program

```
/*Dillon menyerang wilayah terakhir Fio*/  
  
Player Dillon menang! Wilayah AU1 sekarang dikuasai oleh Player Dillon.  
Silahkan tentukan banyaknya tentara yang menetap di wilayah AU1: 5.  
  
Tentara di wilayah AU2: 3.  
Tentara di wilayah AU1: 5.  
  
Jumlah wilayah Player Fio 0.  
Player Fio keluar dari permainan!  
  
(69 ms) yes
```

```
/*Dillon menyerang wilayah terakhir Bagus*/  
  
Player Dillon menang! Wilayah E2 sekarang dikuasai oleh Player Dillon.  
Silahkan tentukan banyaknya tentara yang menetap di wilayah B2: 1.  
  
Tentara di wilayah A1: 7.  
Tentara di wilayah E2: 1.  
  
Jumlah wilayah Player Bagus 0.  
Bagas keluar dari permainan!  
  
*****  
*Dillon telah menguasai dunia*  
*****  
  
(69 ms) yes
```

BONUS

1. Reinforcement

- *Reinforcement* adalah sebuah territory tambahan yang dapat menyimpan Tentara untuk membantu penyerangan. Tentara ini dapat diluncurkan sebagai tambahan kapanpun pemain menyerang.
- Wilayah *reinforcement* bersifat terpisah dari peta, sehingga tidak bisa diserang oleh pemain lain.
- Pada setiap ronde, salah satu pemain mendapatkan kesempatan kecil untuk menerima wilayah *Reinforcement*. Probabilitas mendapatkan wilayah *reinforcement*

ditentukan saat inisialisasi permainan.

Note: *probabilitas memiliki range 0.0-1.0, pastikan untuk divalidasi*

- *Reinforcement* tidak memiliki batas minimal tentara, sehingga dapat menyimpan nol buah tentara.
- Wilayah *Reinforcement* dibatasi dua untuk setiap game, dan seorang pemain tidak diperkenankan untuk memiliki lebih dari satu wilayah *reinforcement*.
- Tampilan notifikasi dan penggunaan *Reinforcement* dibebaskan.

2. **World Leaders**

- *World Leaders* adalah sekumpulan karakter yang dapat memberi keunggulan pada seorang pemain.
- *World Leaders* hanya dibatasi satu untuk setiap pemain yang mendapatkan.
- Untuk mendapatkan *World Leaders*, pemain harus menyelesaikan serangkaian *Secret Missions*.
- Diperbolehkan untuk menyelesaikan bonus ini secara parsial (Hanya satu atau dua *world leaders* diselesaikan). Bonus dinilai sesuai jumlah *world leaders* yang diimplementasikan.
- Tampilan notifikasi dan penggunaan *World Leaders* dibebaskan.
- Berikut adalah rincian dari *World Leaders* yang ada:

a. **Napoleon Bonaparte**

“Napoleon Bonaparte was a French military leader and emperor known for his strategic brilliance and his impact on European history in the late 18th and early 19th centuries.”

Cara mendapatkan:

1. Kuasai seluruh wilayah di benua Eropa.
2. Miliki satu buah wilayah *Reinforcement*.

Advantage:

25% *Chance* untuk otomatis memenangkan *attack* ke wilayah lain tanpa dadu. Efek ini berlaku untuk setiap turn pemain yang memiliki.

b. **Genghis Khan**

“Genghis Khan was the founder and leader of the Mongol Empire, known for his exceptional military prowess and the vast territory he conquered.”

Cara mendapatkan:

1. Kuasai seluruh wilayah pada benua Asia.
2. Eliminasi satu buah pemain, yakni mengakuisisi wilayah terakhir yang dimiliki pemain lawan.

Advantage:

Seluruh territory yang dimiliki dapat menyerang wilayah manapun (tidak harus bersebelahan). Efek ini berlaku untuk setiap turn pemain yang memiliki.

c. **Kanye West**

“Ye is the last living jedi...”

Cara mendapatkan:

1. Kuasai seluruh wilayah pada benua Amerika Utara.
2. Menangkan lima wilayah pada lima ronde berturut.

Advantage:

10% Chance untuk merebut seluruh tentara (kecuali satu) pada sebuah territory random yang tidak dimiliki pemain. Apabila tersisa satu tentara pada wilayah terpilih, wilayah tersebut otomatis diakuisisi. Efek ini berlaku pada awal setiap ronde.

Contoh:

A mengakuisisi Kanye. Pada awal ronde selanjutnya, terdapat 10% kemungkinan A merebut tentara pada salah satu wilayah random milik B, C, atau D (Asumsi terdapat 4 pemain). Terdapat dua kasus:

- Wilayah yang terpilih secara random memiliki lebih dari satu tentara → Pemain A dapat merebut sejumlah $(n-1)$ tentara pada wilayah tersebut. Tentara yang direbut berlaku layaknya tambahan dari tentara yang biasa diperoleh pada setiap ronde.
- Wilayah yang terpilih secara random memiliki satu tentara → Pemain A otomatis mengakuisisi wilayah tersebut.

3. Win Probabilities

Pada command attack, terdapat fase dimana kita harus memilih daerah yang ingin diserang. Dengan mekanisme adu pelemparan dadu, tentu terdapat probabilitas kemenangan yang bisa dihitung ketika ingin melawan suatu wilayah dengan jumlah tentara yang dikirim. Pada bonus ini, Anda bisa berikan nilai probabilitas kemenangan untuk seluruh pilihan wilayah yang dapat diserang. Rumus perhitungan probabilitas dibebaskan namun perlu **dituliskan dan dijelaskan pada laporan**. Berikut adalah contoh tampilannya.

```
/* Mengikut fase attack sebelumnya */

Masukkan banyak tentara yang akan bertempur: 6.
Player A mengirim sebanyak 6 tentara untuk berperang.

/* PETA */

Pilihlah daerah yang ingin Anda serang.
1. B2 (82% win chance)
2. C3 (69% win chance)
3. D4 (30% win chance)
4. E5 (12% win chance)

Pilih:
```

4. Cheat

Untuk mempermudah pengujian, teramat sangat disarankan sekali untuk mengimplementasikan query atau perintah cheat. Cheat yang diimplementasikan dibebaskan sesuai kebutuhan pengujian. Contoh cheat yang dapat diimplementasikan:

- Extra Tentara: Menambahkan tentara dengan jumlah yang diinginkan
- Akuisisi Wilayah: Mengakuisisi wilayah secara langsung
- Ambil Kartu: Memilih *Risk Card* yang diinginkan

Tampilan dan mekanisme cheat dibebaskan.

Kelompok

Pembagian kelompok ditentukan sendiri oleh mahasiswa dengan mengisi [sheet kelompok](#) berikut ini dengan 1 kelompok terdiri dari 4 hingga 5 mahasiswa yang berasal dari kelas yang sama. Batas waktu pengisian kelompok adalah **Jumat, 11 November 2023 pukul 21:21**. Silakan lihat instruksi lebih lanjut di *sheet* kelompok di atas.

FAQ

Pertanyaan dapat ditanyakan pada link [FAQ](#) berikut. Pastikan pertanyaan yang ditanyakan tidak berulang.

Aturan

Terdapat beberapa hal yang harus diperhatikan dalam pengerjaan tugas ini, yakni:

1. **Query dan keluaran tidak harus persis dengan contoh program yang diberikan.**
Hal-hal yang tidak dispesifikkan dapat dianggap sebagai hal yang bebas diimplementasikan seperti apa, silakan berkreativitas sendiri.
2. Semua perubahan yang mungkin Anda lakukan diimplementasi Anda, ataupun fitur-fitur lain yang Anda ubah atau tambahkan, dapat dituliskan di laporan agar dapat dinilai oleh asisten dengan lengkap.
3. Apabila Anda mencari dan mencontoh kode dari Internet, harap cantumkan sumbernya dalam bentuk komentar dalam program.
4. Jika terdapat hal yang tidak dimengerti, silahkan ajukan pertanyaan kepada asisten melalui link FAQ yang telah diberikan di atas. Pertanyaan yang diajukan secara personal ke asisten tidak akan dijawab untuk menghindari perbedaan informasi yang didapatkan oleh peserta praktikum.
5. Dilarang melakukan **plagiarisme**. Pelanggaran pada poin ini akan menyebabkan pemberian **nilai E** pada setiap anggota kelompok yang melakukan maupun memberi.

Deliverables

1. Untuk tugas ini Anda diwajibkan menggunakan *version control system* **git** dengan menggunakan sebuah *repository* **private** di Github Classroom “**IF2121 Logika Komputasional 2023**” (gunakan surel *student* agar gratis).
2. Anda diwajibkan untuk membuat *repository* di dalam Github Classroom menggunakan [link](#)

invitation berikut ini. Berikut adalah alur untuk membuat *repository*:

- a. Ketua kelompok mengisi kolom 'NIM 1' dan 'Username GitHub 1' pada [sheet kelompok](#) dan anggota lainnya mengisi kolom 'NIM 2' sampai 'NIM 5' beserta 'Username GitHub 2' sampai 'Username GitHub 5'.
 - b. Semua anggota kelompok memilih *identifier* yang sesuai dengan kelas, NIM, dan nama mereka.
 - c. Ketua kelompok membuat *team* baru dengan nama kelompok yang sama dengan yang dicantumkan di [sheet kelompok](#).
 - d. Anggota kelompok melakukan *join* ke *team* yang sudah dibuat ketua kelompok.
3. Meskipun commit tidak dinilai, lakukanlah *commit* yang wajar dan sesuai *best practice* (tidak semua kode satu *commit*). Disarankan juga menggunakan [semantic commits](#) agar asisten lebih mudah menilai beban pekerjaan masing-masing anggota kelompok.
4. *Repository* memiliki struktur sebagai berikut.

```
nama-repository
├── src
│   ├── file1.pl
│   ├── file2.pl
│   └── main.pl
├── doc
│   ├── Progress1_GXX.pdf
│   ├── Progress2_GXX.pdf
│   └── Laporan_KXX_GYY.pdf
└── README.md
```

5. Pengumpulan dilakukan dengan membuat **release** dengan tag **vX.Y** pada repository yang telah kelompok Anda buat sebelum deadline, dengan **X** adalah angka sesuai dengan nomor milestone dan **Y** adalah angka revisi dimulai dari 0 jika tidak ada revisi/versi awal. Pastikan tag sesuai format.
6. Berikut adalah informasi tambahan mengenai Milestone dan *deliverables*-nya:
- a. Pada **Milestone 1**, Anda diminta untuk paling tidak membuat **detail lengkap rencana fakta dan rule** yang akan digunakan, baik yang menggunakan list, control loop. Anda juga diminta untuk melampirkan sebuah file berisi progress pengerjaan, rancangan rule, dan rancangan fakta dengan format sebagai [berikut](#) dengan nama **Progress1_GXX.pdf** dengan XX merupakan nomor kelompok.

- b. Pada **Milestone 2**, Anda diminta paling tidak sudah mengimplementasikan fakta-fakta terkait permainan dan Implementasi rule-rule kendali dasar dalam bentuk *source code*. Anda juga diminta untuk melampirkan sebuah file berisi progress pengerjaan keseluruhan dengan format sebagai [berikut](#) dengan nama **Progress2_GXX.pdf** dengan XX merupakan nomor kelompok.
 - c. Pada **Milestone 3**, Anda diminta untuk mengumpulkan berkas-berkas berikut:
 - i. Source code program keseluruhan
 - ii. Laporan hasil kerja dengan format penamaan **Laporan_GXX.pdf** dengan XX adalah nomor kelompok yang terdiri atas:
 - 1. Halaman *cover* yang memuat judul tugas, kode dan nama mata kuliah, dan identitas anggota kelompok
 - 2. Penjelasan setiap *command* yang kelompok Anda telah buat, termasuk command bonus (jika ada), penjelasan meliputi:
 - a. Kegunaan command tersebut
 - b. Skenario-skenario penggunaannya (beserta contoh)
 - c. Tidak perlu menjelaskan cara kerja *command*
 - 3. Hasil eksekusi program berupa jalannya alur permainan (dalam bentuk *screenshot*)
 - 4. Pembagian dan persentase kerja masing-masing anggota kelompok
 - iii. README file. Gunakan [template](#) ini untuk membantu pengerjaan kalian.
7. Tugas Besar IF2121 - Logika Komputasional adalah kegiatan yang bersifat **mandiri berkelompok**. Semua bentuk kecurangan akan ditindaklanjuti sesuai dengan sanksi akademik yang ada.