

LAPORAN TUGAS BESAR IF2121 LOGIKA KOMPUTASIONAL

Global Conquest: Battle for Supremacy

**Diajukan sebagai salah satu tugas mata kuliah IF2121 Logika Komputasional pada
Semester I Tahun Akademik 2023/2024**



Disusun oleh:

Kelompok Denghis Ghan / K-01

Denise Felicia Tiowanni	13522013
Amalia Putri	13522042
Erdianti Wiga Putri Andini	13522053
Immanuel Sebastian Girsang	13522058

SEKOLAH TEKNIK ELEKTRO DAN INFORMATIKA

INSTITUT TEKNOLOGI BANDUNG

BANDUNG

2023

DAFTAR ISI

DAFTAR ISI.....	1
BAB I	
PENDAHULUAN.....	2
1.1. Deskripsi Permasalahan.....	2
BAB II	
DASAR TEORI.....	3
2.1. Rekurens.....	3
2.2. List.....	3
2.3. Cut.....	4
2.4. Fail.....	4
2.5. Loop.....	5
BAB III	
PROGRAM.....	6
3.1 Map.....	6
3.2 Initiating.....	6
3.3 Turn.....	8
3.4 Wilayah.....	9
3.5 Player.....	10
BAB IV	
UJI COBA PROGRAM.....	12
4.1. Map.....	12
4.2. Initiating.....	12
4.3. Turn.....	14
4.4. Wilayah.....	18
4.5. Player.....	18
BAB V	
PENUTUP.....	21
5.1. Kesimpulan.....	21
5.2. Saran.....	21
BAB VI	
PEMBAGIAN KERJA.....	23

BAB I

PENDAHULUAN

1.1. Deskripsi Permasalahan

Global Conquest: Battle for Supremacy adalah sebuah game strategi yang kami buat dengan menggunakan bahasa pemrograman deklaratif Prolog (GNU Prolog). Tujuan dari pembuatan game ini adalah untuk mengkombinasikan keterampilan dan teknik yang telah dipelajari pada kuliah IF2121 Logika Komputasional meliputi rekurens, list, cut, fail, dan loop.

Permainan Global Conquest: Battle for Supremacy dapat dimainkan oleh 2-4 orang. Permainan ini bertujuan untuk menguasai dunia, sesuai dengan wilayah-wilayah yang ditentukan di dalam permainan. Masing-masing pemain akan mendapatkan tentara dan mengoperasikan tentara-tentara tersebut sesuai dengan *command-command* game yang tertera. *Command* yang tersedia antara lain adalah:

- displayMap
- startGame
- takeLocation
- placeTroops
- placeAutomatic
- endTurn
- draft
- move
- attack
- risk
- checkLocationDetail
- checkPlayerDetail
- checkIncomingTroops

BAB II

DASAR TEORI

2.1. Rekurens

Rekurens adalah suatu konsep di Prolog yang memungkinkan sebuah aturan atau fakta dapat merujuk kembali pada dirinya sendiri. Ini sering digunakan untuk mendefinisikan fungsi yang berulang atau mengakses struktur data yang bersifat rekursif. Analisis rekurens harus mendefinisikan aturan basis dimana merupakan kondisi ketika program berhenti dan aturan rekurens yang harus menuju basis. Contoh singkat program dengan rekurens adalah sebagai berikut.

```
/* basis */
factorial(0, 1).

/* rekurens */
factorial(N, Result) :-
    N > 0,
    N1 is N - 1,
    factorial(N1, Result1),
    Result is N * Result1.
```

2.2. List

List adalah struktur data dalam Prolog yang dapat berisi elemen-elemen yang bisa memiliki tipe data yang berbeda, seperti konstanta, variabel, ataupun struktur yang mungkin juga berupa list. Dalam Prolog, list dituliskan dengan kurung siku [] dan pemisahan Head dan Tail dilakukan dengan pipe |. Dalam list, setiap elemen boleh muncul lebih dari satu kali. Berikut adalah contoh list.

```
% fakta list kosong
empty_list([]).

% fakta list dengan elemen-elemen
non_empty_list([1, 2, 3, 4]).
```

2.3. Cut

Cut (!) adalah suatu predikat di Prolog yang digunakan untuk mengontrol backtracking. Cut memutuskan cabang pencarian dan menghindari solusi alternatif. Penggunaan cut bisa meningkatkan efisiensi program, tetapi harus digunakan dengan hati-hati karena dapat mempengaruhi semantik program. Prolog memanfaatkan cut agar program lebih efisien. Berikut adalah contoh penggunaan cut.

```
% contoh penggunaan cut
member(X, [X|_]) :- !.
member(X, [_|T]) :- member(X, T).
```

2.4. Fail

Fail adalah kondisi dimana sebuah statement gagal memenuhi suatu persyaratan. Fail digunakan untuk memanipulasi program sehingga suatu proses tertentu harus dilakukan. Pada defaultnya, jika suatu program mencapai kondisi fail, program akan melakukan backtrack. Kondisi fail tidak selalu harus tercapai dalam suatu program, bisa saja program tersebut langsung benar pada persyaratan pertama sehingga tidak diperlukan adanya backtrack. Fail umumnya digunakan untuk merealisasikan Error Message. Dengan Fail, Prolog justru memaksa proses backtrack ke suatu cabang tertentu.

```
example_predicate(X) :- condition(X), !, fail.
example_predicate(X) :- alternative_condition(X).
```

2.5. Loop

Walaupun Prolog merupakan bahasa deklaratif yang tidak memiliki konsep loop seperti bahasa pemrograman imperatif, pengulangan dalam Prolog dapat diimplementasikan dengan rekurens. Berikut adalah contoh aplikasinya. Pada contoh di bawah ini, loop akan melakukan `do_something` sebanyak N kali.

```
loop(0) :- !.  
loop(N) :-  
    do_something,  
    N1 is N - 1,  
    loop(N1).
```

BAB III

PROGRAM

3.1 Map

Perintah/Predikat/Rule/Fact	Keterangan
displayMap	Menampilkan gambar peta permainan
getTentara>NamaWilayah, JumlahTentara)	Mendapatkan tentara berdasarkan nama wilayah

3.2 Initiating

Perintah/Predikat/Rule/Fact	Keterangan
startGame	Menampilkan gambar peta permainan
playRound(PlayerList, PlayerNames)	Mendapatkan tentara berdasarkan nama wilayah
initiatePlayer(N, Z, [Z Rest], [Name NamesListRest])	Menginisiasi pemain terkait informasi pemain, keterangan <i>attack</i> , dan <i>risk</i>
rollDice([PlayerName Rest], PlayerNames)	Pemain melemparkan dadu dengan format nama dan jumlah nilai dadu yang didapatkan
initDice(PlayerName, Roll)	inisialisasi pemain terhadap pelemparan dadu
findMaxDice(PlayerList, MaxPlayer, MaxValue)	Mengembalikan jumlah nilai dadu maksimal yang didapatkan pemain
findMaxDiceHelper([PlayerName Rest], CurrMaxPlayer, CurrMaxValue, MaxPlayer, MaxValue)	Mencari jumlah nilai dadu pemain maksimal dan menjadikan pemain tersebut maju langkah pertama
setPlayerOrder(PlayerList, MaxPlayer)	Menentukan urutan pemain
displayPlayerOrder(PlayerNames)	Mencetak urutan pemain
displayPlayerList([Player Rest], PlayerNames)	Mencetak list pemain
findPlayerIndex([_ Rest], Player, Index)	Mencari indeks pemain pada list pemain

rotateList(List, N, RotatedList)	Mengubah urutan pemain dengan menambahkan bagian depan/belakang list
countMaxPlayers([PlayerName Rest], MaxValue, Count)	Menghitung banyaknya pemain yang main dalam gim
displayPlayerNames([PlayerName Rest])	Mencetak nama-nama pemain dalam gim
getPlayerNameByIndex(Index, PlayerNames, NamaPlayer)	Mengembalikan indeks pemain dari list
takeLocation(Location)	Menjalankan fungsi takeLocation dengan bantuan fungsi-fungsi di bawah.
updatePlayerTerritories(Player, Location, NewListWilayah)	Menginsert wilayah yang ditake ke list wilayah per pemain dan mengupdate jumlah tentaranya
updatePlayerBenua(Player, KodeWilayah, NewListBenua)	Menginsert benua tempat wilayah yang ditake ke list wilayah per pemain
handleLocationSelection(Location, KodePemain, Player, PlayerName, Iterate)	Fungsi untuk handle kondisi dimana bila sebuah lokasi sudah diambil oleh player/belum
handleLastLocationSelection(Location, KodePemain, Player, PlayerName)	Fungsi untuk handle kondisi takeLocation terakhir yaitu setelah 24 kali dijalankan dengan benar oleh para pemain
change_iterator	Fungsi untuk mengecek iterasi secara increment 1 untuk mendeteksi iterasi 24
switchTurn	Fungsi untuk berpindah kepada next player
placeTroopsHelper(KodePemain, RemainingTroops, KodeWilayah, JumlahTroopsYangDitaroh)	Fungsi yang dijalankan pada command placeTroops
placeTroops(KodeWilayah, JumlahTroopsYangDitaroh)	Fungsi untuk para pemain set tentara di wilayahnya
change_iterator_troops	Fungsi untuk mengecek iterasi secara increment 1 untuk mendeteksi iterasi sejumlah troops pemain
placeAutoTroops(KodePemain, [Wilayah Rest], SisaTentara, NumWilayah)	Fungsi untuk meletakkan tentara secara otomatis pada beberapa wilayah yang dimiliki pemain
placeTroopsAutoHelper(KodePemain, Wilayah, JumlahTroops)	Fungsi yang dijalankan pada command placeAutomatic
assertTroopsAuto(KodePemain,	Fungsi untuk mengupdate infoplayer tiap pemain yang

JumlahTroops)	melakukan troopsautomatic.
---------------	----------------------------

3.3 Turn

Perintah/Predikat/Rule/Fact	Keterangan
endTurn	<p>Mengakhiri giliran pemain <i>current</i> dan melempar giliran tersebut kepada <i>next player</i> dengan memberikan tambahan tentara, yakni berupa (jumlah wilayah DIV 2),</p> <p>dan juga ditambah dengan bonus tentara dari benua hanya jika pemain tersebut menguasai seluruh wilayah di benua tersebut, yakni dengan ketentuan:</p> <p>'AMERIKA UTARA' : 5 'EROPA':5 'ASIA' : 7 'AMERIKA SELATAN' : 2 'AFRIKA' : 3 'AUSTRALIA' : 2</p> <p>Juga menangani pergantian pemain terhadap <i>risk: auxiliary</i> dan <i>supply issue</i>.</p>
draft(Wilayah, Troops)	Meletakkan tentara tambahan pada suatu wilayah dengan syarat bahwa pemain tersebut menguasai wilayah tersebut, jika tidak, draft akan dibatalkan
move(WilayahAsal, WilayahTujuan, JumlahTentaraDipindahkan)	Memindahkan sejumlah tentara dari wilayah asal ke wilayah tujuan dengan syarat pemain tersebut memiliki kedua wilayah tersebut, jika tidak, move akan dibatalkan
attack	Pemain melakukan penyerangan dengan memilih daerah yang ingin diserang dan dapat diserang hanya jika wilayah diserang bertetangga dengan wilayah pemain yang menyerang. Penyerangan ini dilakukan dengan nilai $attack = defend = \text{jumlah nilai dadu yang didapatkan} \times \text{jumlah tentara}$ dan akan dilakukan penyerangan jika nilainya lebih besar dari nilai perlindungan.
writePossibleWilayah(Num, [Wilayah Rest], Max)	Mencetak wilayah yang dapat diserang
randomDiceAttack(Player, NumDice)	Mengeluarkan nilai sejumlah dadu acak untuk penyerangan
rollOneDice(Result, X, Y)	Mengeluarkan nilai dadu acak untuk satu dadu
rollDices(N, [Result Results],X,Y)	Melakukan pelemparan sejumlah dadu

displayDiceRolls(Num, [Roll Rest], Max)	Mencetak hasil nilai pelemparan dadu
risk_scenarios	List yang berisi kumpulan risk, yakni 'CEASEFIRE ORDER', 'SUPER SOLDIER SERUM', 'AUXILIARY TROOPS', 'REBELLION', 'DISEASE OUTBREAK', 'SUPPLY CHAIN ISSUE'
initiateRisk(NextPlayer)	Menginisiasi <i>risk</i> yang dijalankan oleh pemain
resetRisk(NextPlayer)	Mengatur ulang <i>risk</i> pada pemain
risk	Perintah untuk memanggil <i>risk</i> pada pemain
risk(CurrentPlayer)	Melakukan pengacakan jenis <i>risk</i> yang ada
apply_risk('CEASEFIRE ORDER', CurrentPlayer)	Giliran berikutnya, wilayah pemain tidak dapat diserang oleh lawan.
apply_risk('SUPER SOLDIER SERUM', CurrentPlayer)	Semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 6
apply_risk('AUXILIARY TROOPS', CurrentPlayer)	Tentara tambahan yang didapatkan pemain akan bernilai 2 kali lipat.
apply_risk('REBELLION', CurrentPlayer)	Salah satu wilayah acak pemain akan berpindah kekuasaan menjadi milik lawan.
apply_risk('DISEASE OUTBREAK', CurrentPlayer)	Semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 1.
apply_risk('SUPPLY CHAIN ISSUE', CurrentPlayer)	Pada giliran berikutnya, pemain tidak mendapatkan tentara tambahan.
update_facts([wilayah(A,CurrentPlayer, B, C, D, OldNum) Rest], CurrentPlayer, Num)	Memperbarui fakta-fakta terkait wilayah dan <i>player</i> yang sedang giliran
random_member(X, List)	Mengacak urutan member dari list pemain

3.4 Wilayah

Perintah/Predikat/Rule/Fact	Keterangan
tetanggaList(KodeWilayah,	Menemukan wilayah-wilayah yang bertetangga dalam list

ListTetangga)	
writeTetangga(ListTetangga)	Mencetak wilayah-wilayah yang bertetanggaan
checkLocationDetail(KodeWilayah)	Mengecek informasi wilayah berdasarkan kode, nama, dan total tentara di wilayah

3.5 Player

Perintah/Predikat/Rule/Fact	Keterangan
listBenuaDikuasai(KodePemain, ListBenuaDikuasai)	Mengembalikan list benua yang dikuasai oleh pemain
listWilayahDiBenua(KodePemain, KodeBenua, ListWilayahDiBenua)	Mengembalikan list wilayah
print_list_benua([KodeBenua Tail])	Mencetak list benua
displayWilayahBenuaDimiliki(KodePemain, KodeBenua)	Mencetak wilayah yang dikuasai oleh pemain
print_list_wilayah_benua([KodeWilayah Tail])	Mencetak list benua yang dikuasai oleh pemain
writeBenuaDikuasai(ListBenuaDikuasai)	Mencetak benua yang dikuasai oleh pemain
checkPlayerTerritories(KodePemain)	Mengecek wilayah yang dimiliki oleh pemain
checkPlayerDetail(KodePemain)	Mengecek pemain terkait total wilayah, tentara aktif, dan tentara tambahan
print_bonus_troops_list([KodeBenua Tail])	Mencetak list bonus tentara pada benua
troopsBenuaDikuasai(KodePemain, ListBenuaDikuasai, TotalBonusTroops)	Mencari dan menghitung banyaknya total tentara bonus yang ada di benua
calculateBonusTroops([Benua Rest], TotalBonusTroops)	Menghitung total tentara bonus, yakni bonus tentara yang tersisa dijumlahkan dengan total wilayah yang ada
checkIncomingTroops(KodePemain)	Mengecek jumlah tentara tambahan yang ingin ditambah dari wilayah

BAB IV

UJI COBA PROGRAM

4.1. Map

displayMap	<pre> ?- displayMap. ##### # North America # Europe # Asia # # [NA1(0)]-[NA2(0)] # # # # # # # -----[NA5(0)]-----[E1(0)]-[E2(0)]-----[A1(0)] [A2(0)] [A3(0)]----- # [NA3(0)]-[NA4(0)] # # # # [E3(0)]-[E4(0)] ##### # # ##### ##### ##### # # ##### ##### ##### # [SA1(0)] # [AF2(0)] # [A6(0)]---[A7(0)] # # # # # ---[SA2(0)]-----[AF1(0)]--- # # # # [AF3(0)] # # # # # # [AU1(0)]---[AU2(0)]----- # # # # # # South America # Africa # Australia # ##### (1 ms) yes </pre>
------------	--

4.2. Initiating

startGame	jumlah pemain tidak valid	<pre> ?- startGame. [Masukkan jumlah pemain: 5. Mohon masukkan angka antara 2 - 4. [Masukkan jumlah pemain: 1. Mohon masukkan angka antara 2 - 4. Masukkan jumlah pemain: █ </pre>
	jumlah pemain valid	<pre> [Masukkan jumlah pemain: 4. [Masukkan nama pemain 1: 'Nuel'. [Masukkan nama pemain 2: 'Densu'. [Masukkan nama pemain 3: 'Wiga'. [Masukkan nama pemain 4: 'Amel'. Nuel melempar dadu dan mendapatkan 7. Densu melempar dadu dan mendapatkan 4. Wiga melempar dadu dan mendapatkan 2. Amel melempar dadu dan mendapatkan 9. Urutan pemain: Amel - Nuel - Densu - Wiga Amel dapat mulai terlebih dahulu. Setiap pemain mendapatkan 12 tentara. Giliran Amel untuk memilih wilayahnya. (2 ms) yes </pre>

takeLocation	nama wilayah valid dan belum diambil	<pre>[?- takeLocation(a1). Amel mengambil wilayah A1. Giliran Nuel untuk memilih wilayahnya. yes [?- takeLocation(a2). Nuel mengambil wilayah A2. Giliran Densu untuk memilih wilayahnya. yes [?- takeLocation(a3). Densu mengambil wilayah A3. Giliran Wiga untuk memilih wilayahnya. yes [?- takeLocation(a4). Wiga mengambil wilayah A4. Giliran Amel untuk memilih wilayahnya. (1 ms) yes _</pre>
	nama wilayah valid dan sudah diambil	<pre>[?- takeLocation(a4). Wilayah A4 sudah dikuasai. Tidak bisa mengambil. Giliran Nuel untuk memilih wilayahnya. yes _</pre>
	nama wilayah tidak valid	<pre>[?- takeLocation(c3). no _</pre>
	semua wilayah sudah selesai diambil	<pre>[?- takeLocation(sa2). Amel mengambil wilayah SA2. Seluruh wilayah telah diambil pemain. Memulai pembagian sisa tentara. Giliran Nuel untuk meletakkan tentaranya. (1 ms) yes _</pre>
placeTroops	wilayah dimiliki oleh pemain lain	<pre>[?- placeTroops(au2, 4). Wilayah tersebut dimiliki pemain lain. Silakan pilih wilayah lain. Giliran Wiga untuk meletakkan tentaranya. yes _</pre>

	jumlah tentara yang dimiliki pemain tidak mencukupi	<pre> [] ?- placeTroops(e2, 100). Jumlah tentara yang dimiliki Wiga tidak cukup. yes </pre>
	jumlah tentara yang dimiliki cukup dan wilayah merupakan milik pemain	<pre> [] ?- placeTroops(e2, 3). Player Wiga meletakkan 3 tentara di e2. Terdapat 3 tentara yang tersisa. yes </pre>
placeAutomatic		

4.3. Turn

endTurn	sudah inisiasi pemain	<pre> [] ?- endTurn. Player WIGA mengakhiri giliran. Sekarang giliran Player AMEL! Player AMEL mendapatkan 8 tentara tambahan. yes [] ?- endTurn. Player AMEL mengakhiri giliran. Sekarang giliran Player WIGA! Player WIGA mendapatkan AUXILIARY TROOPS! Player WIGA mendapatkan 2 tentara tambahan. yes [] ?- endTurn. Player AMEL mengakhiri giliran. Sekarang giliran Player WIGA! Player WIGA terdampak SUPPLY CHAIN ISSUE! Player WIGA tidak mendapatkan tentara tambahan. yes </pre>
---------	-----------------------	---

draft	sudah inisiasi pemain	<pre> ?- draft(na1,5). Player AMEL meletakkan 5 tentara tambahan di na1. Tentara total di na1: 26 Jumlah Pasukan Tambahan Player AMEL: 1 yes ?- draft(na1,1). Player AMEL meletakkan 1 tentara tambahan di na1. Tentara total di na1: 27 Jumlah Pasukan Tambahan Player AMEL: 0 yes ?- draft(na9,1). AMEL tidak memiliki wilayah na9. draft dibatalkan. yes</pre>
move	sudah inisiasi pemain	<pre> ?- move(na1,na2,3). AMEL memindahkan 3 tentara dari na1 ke na2. Jumlah tentara di na1: 7 Jumlah tentara di na2: 13 yes ?- move(na3,na1,3). AMEL memindahkan 3 tentara dari na3 ke na1. Jumlah tentara di na3: 7 Jumlah tentara di na1: 10 yes ?- move(na1,na2,3). AMEL memindahkan 3 tentara dari na1 ke na2. Jumlah tentara di na1: 7 Jumlah tentara di na2: 16 yes ?- move(na2,na1,3). Player AMEL sudah tidak dapat MOVE lagi. Pemindahan dibatalkan. yes</pre>
attack	daerah yang dipilih bukan milik pemain	<pre> ?- attack. Sekarang giliran player Denise Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan: au1. Daerah yang Anda pilih bukan daerah Anda! Silahkan pilih daerah lain. Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan: █</pre>
	daerah yang dipilih milik pemain	<pre>Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan: a2. Player Denise ingin memulai penyerangan dari daerah A2. Dalam daerah a2 Anda memiliki sebanyak 11 tentara Masukkan banyak tentara yang akan bertempur: █</pre>
	tentara tidak mencukupi	<pre>Masukkan banyak tentara yang akan bertempur: 100. Jumlah tentara yang Anda masukkan tidak valid! Silahkan masukkan jumlah tentara yang valid. Masukkan banyak tentara yang akan bertempur: █</pre>

	tentara mencukupi	<p>Masukkan banyak tentara yang akan bertempur: 8.</p> <p>Player Denise mengirim sebanyak 8 tentara untuk berperang. Pilihlah daerah yang ingin Anda serang.</p> <p>1. a5</p> <p>Pilih: <input type="text"/></p>
	penyerangan berhasil	<p>Pilih: 1.</p> <p>Perang terhadap wilayah A5 telah dimulai.</p> <p>Player Denise Dadu 1: 6. Dadu 2: 2. Dadu 3: 3. Dadu 4: 5. Dadu 5: 3. Dadu 6: 2. Dadu 7: 2. Dadu 8: 3. Total: 26.</p> <p>Player Nuel Dadu 1: 1. Total: 1.</p> <p>Player Denise menang! Wilayah a5 sekarang dikuasai oleh Player Denise.</p> <p>Silahkan tentukan banyaknya tentara yang menetap di wilayah a5 : <input type="text"/></p>
	masukan tentara yang menetap tidak valid	<p>Silahkan tentukan banyaknya tentara yang menetap di wilayah a5 : 100.</p> <p>Pasukan harus lebih dari atau sama dengan 1 dan kurang dari atau sama dengan yang berperang!</p> <p>Silahkan tentukan banyaknya tentara yang menetap di wilayah a5 : <input type="text"/></p>
	penyerangan selesai	<p>Silahkan tentukan banyaknya tentara yang menetap di wilayah a5 : 5.</p> <p>Tentara di wilayah a2 : 6. Tentara di wilayah a5 : 5.</p> <p>(2 ms) yes <input type="text"/></p>
	sudah menggunakan giliran attack	<p>[?- attack. Anda sudah menyerang!</p> <p>no <input type="text"/></p>

	penyerangan gagal	<p>Pilihlah daerah yang ingin Anda mulai untuk melakukan penyerangan: a1.</p> <p>Player a ingin memulai penyerangan dari daerah A1. Dalam daerah a1 Anda memiliki sebanyak 4 tentara</p> <p>Masukkan banyak tentara yang akan bertempur: 3.</p> <p>Player a mengirim sebanyak 3 tentara untuk berperang. Pilihlah daerah yang ingin Anda serang. 1. e2 2. a4</p> <p>Pilih: 2.</p> <p>Perang terhadap wilayah A4 telah dimulai.</p> <p>Player a Dadu 1: 2. Dadu 2: 6. Dadu 3: 5. Total: 13.</p> <p>Player b Dadu 1: 5. Dadu 2: 4. Dadu 3: 1. Dadu 4: 4. Dadu 5: 6. Total: 20.</p> <p>Player b menang! Sayang sekali penyerangan anda gagal :((Tentara di wilayah a1 : 1. Tentara di wilayah a4 : 5.</p> <p>(1 ms) yes</p>
risk	CEASEFIRE ORDER	<p>[-] ?- risk. Player AMEL mendapatkan risk card CEASEFIRE ORDER Hingga giliran berikutnya, wilayah pemain tidak dapat diserang oleh lawan.</p> <p>yes</p>
	SUPER SOLDIER SERUM	<p>[-] ?- risk. Player AMEL mendapatkan risk card SUPER SOLDIER SERUM Hingga giliran berikutnya, semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 6.</p> <p>yes</p>
	AUXILIARY TROOPS	<p>[-] ?- risk. Player AMEL mendapatkan risk card AUXILIARY TROOPS Pada giliran berikutnya, Tentara tambahan yang didapatkan pemain akan bernilai 2 kali lipat.</p> <p>yes</p>
	REBELLION	<p>[+] ?- risk. Player AMEL mendapatkan risk card REBELLION Salah satu wilayah acak pemain akan berpindah kekuasaan menjadi milik lawan.</p> <p>Wilayah na4 sekarang dikuasai oleh Player WIGA .</p> <p>yes _</p>
	DISEASE OUTBREAK	<p>[-] ?- risk. Player AMEL mendapatkan risk card DISEASE OUTBREAK Hingga giliran berikutnya, semua hasil lemparan dadu saat penyerangan dan pertahanan akan bernilai 1.</p> <p>yes _</p>

	SUPPLY CHAIN ISSUE	<pre> ?- risk. Player AMEL mendapatkan risk card SUPPLY CHAIN ISSUE Pada giliran berikutnya, pemain tidak mendapatkan tentara tambahan. yes </pre>
--	-----------------------	--

4.4. Wilayah

checkLocation Detail	saat sudah terdapat tentara	<pre> ?- checkLocationDetail(e2). Kode: E2 Nama: Swedia Total Tentara: 4 Tetangga: Islandia, Rusia, China yes _ </pre>
	Saat belum terdapat tentara	<pre> ?- checkLocationDetail(na1). Kode: NA1 Nama: Greenland Total Tentara: 0 Tetangga: Canada, Amerika Serikat yes _ </pre>
	Saat kode wilayah tidak valid	<pre> ?- checkLocationDetail(c5). no _ </pre>

4.5. Player

checkPlayerDetail(KodePemain)	saat kode pemain valid	<pre> ?- checkPlayerDetail(1). Nama: Densu Total Wilayah: 6 Benua: Pemain ini tidak menguasai benua manapun. Total Tentara Aktif: 6 Total Tentara Tambahan: 0 yes _ </pre>
-------------------------------	---------------------------	---

	saat kode pemain tidak valid	<pre> ?- checkPlayerDetail(6). no </pre>
checkPlayerTerritories(KodePemain)	saat kode pemain valid	<pre> ?- checkPlayerTerritories(1) . Nama: amel Benua AMERIKA UTARA (2/5). NA4 Nama : Mexico. Jumlah Tentara : 1. NA2 Nama : Canada. Jumlah Tentara : 1. Benua AMERIKA SELATAN (1/2). SA1 Nama : Brazil. Jumlah Tentara : 1. Benua EROPA (3/5). E5 Nama : Ukraina. Jumlah Tentara : 1. E3 Nama : Finlandia. Jumlah Tentara : 1. E1 Nama : Islandia. Jumlah Tentara : 1. Benua ASIA (4/7). A7 Nama : Indonesia. Jumlah Tentara : 1. A5 Nama : Taiwan. Jumlah Tentara : 1. A3 Nama : Jepang. Jumlah Tentara : 1. A1 Nama : China. Jumlah Tentara : 1. Benua AFRIKA (1/3). AF2 Nama : Sudan. Jumlah Tentara : 1. Benua AUSTRALIA (1/2). AU1 Nama : New South Wales. Jumlah Tentara : 1. yes </pre>

	saat kode pemain tidak valid	?- checkPlayerTerritories(5). no _ _ _
checkIncoming Troops(KodePe main)	saat kode pemain valid	?- checkIncomingTroops(1). Nama: amel Total Wilayah: 12 Jumlah tentara tambahan dari wilayah: 6 Total Tentara Tambahan: 6 yes _ _
	saat kode pemain tidak valid	?- checkIncomingTroops(5). no _ _ _

BAB V

PENUTUP

5.1. Kesimpulan

Pada Tugas Besar IF2121 Logika Komputasional, kami diminta untuk membuat suatu program permainan strategi dengan menggunakan GNU Prolog. Program yang dibuat harus mengimplementasikan rekurens, list, cut, fail, dan loop. Permainan, bernama “Global Conquest: Battle for Supremacy,” ini melibatkan 2 hingga 4 pemain yang bersaing untuk menguasai 24 wilayah di 6 benua. Pemain menggunakan perintah seperti Draft, Move, Risk, Attack, dan EndTurn untuk mengelola tentara, menyerang wilayah, dan memenangkan permainan.

Alur permainan mencakup inisiasi pemain, inisiasi map permainan, lemparan dadu untuk menentukan urutan pemain, distribusi tentara awal, dan pembagian wilayah. Pemain dapat memilih secara manual atau otomatis untuk mendistribusikan tentara. Permainan dimulai setelah semua pemain menyelesaikan pembagian tentara.

5.2. Saran

Beberapa saran yang harus di refleksi oleh kelompok tugas besar IF2121 Logika Komputasional ‘Denghis Ghan’ antara lain:

1. Perlu adanya *management* waktu yang lebih baik lagi untuk kelompok Denghis Ghan agar pekerjaan dapat lebih cepat selesai dan tidak terburu-buru di dekat deadline.
2. Perlu adanya management kerja yang lebih baik lagi untuk kelompok Denghis Ghan agar semua program dapat berjalan dengan baik.
3. Meningkatkan pemahaman pada bahasa Prolog agar dapat mempermudah dan mempercepat proses pengerjaan.

5.3. Github Repository

Link Github Repository kelompok Denghis Ghan dapat diakses pada tautan berikut:

<https://github.com/GAIB21/tugas-besar-if2121-logika-komputasional-2023-denghis-ghan>

BAB VI

PEMBAGIAN KERJA

NIM	NAMA	TUGAS
13522013	Denise Felicia Tiowanni	<ul style="list-style-type: none"> - Mengerjakan fungsi startGame - Mengerjakan inisialisasi Map - Membuat Facts - Membuat laporan bab 1, bab 2, bab 3, bab 4, bab 5
13522042	Amalia Putri	<ul style="list-style-type: none"> - Mengerjakan End Turn, Move, Draft - Membuat utils functions - Membuat Facts - Membuat laporan bab 3, bab 4, bab 5
13522053	Erdianti Wiga Putri Andini	<ul style="list-style-type: none"> - Mengerjakan Wilayah, Troops, Player - Mengerjakan inisialisasi Map - Membuat utils functions - Membuat Facts - Membuat laporan bab 1, bab 2, bab 3, bab 4, bab 5
13522058	Imanuel Sebastian Girsang	<ul style="list-style-type: none"> - Mengerjakan Attack, Risk - Membuat Facts - Membuat laporan bab 4