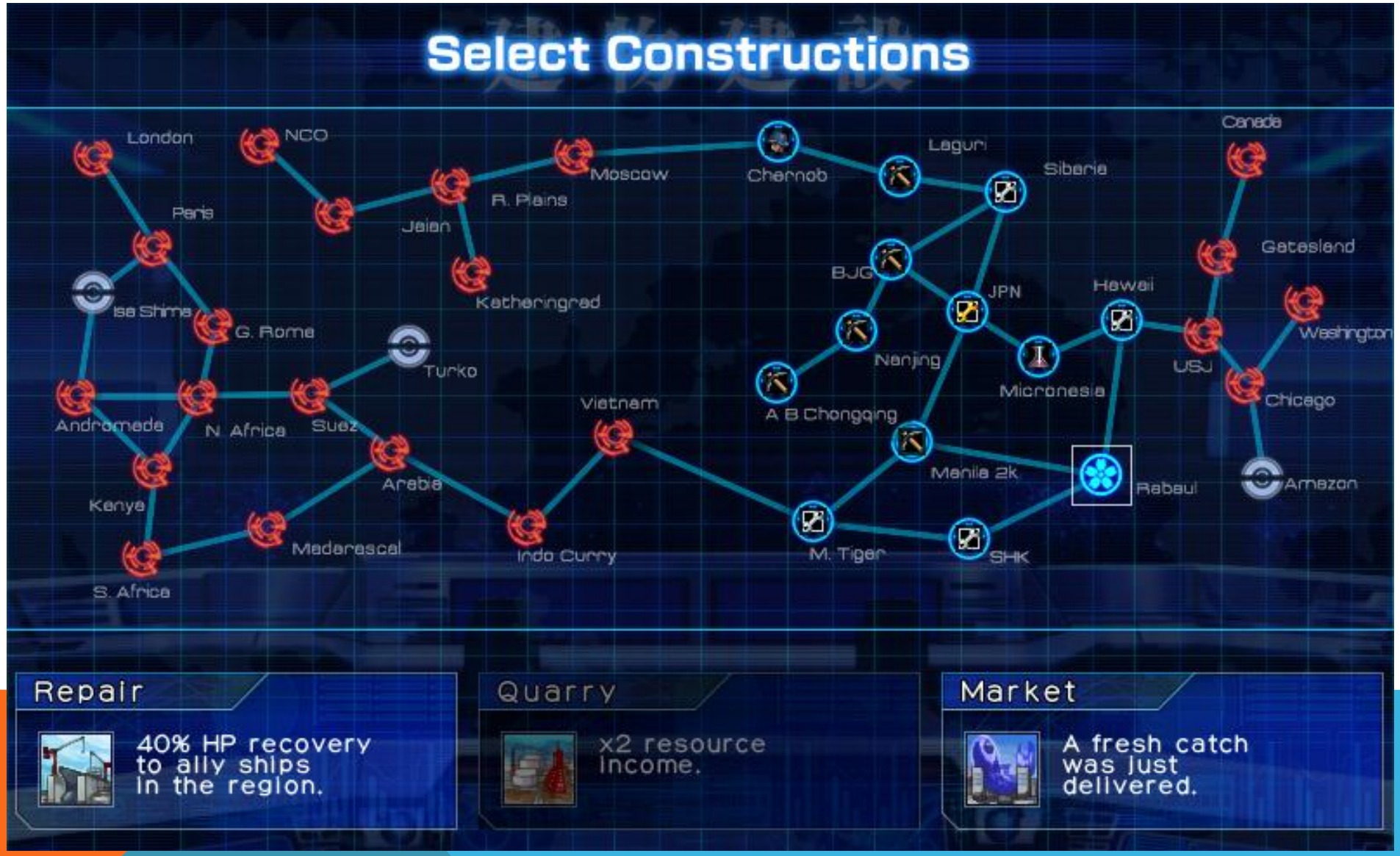
TURN-BASED STRATEGY GAME MENGGUNAKAN ALGORITMA RESOURCE ASSIGNMENT PADA PERANGKAT MOBILE BERBASIS ANDROID

Oleh : Riandanu Madi Utomo

TURN-BASED STRATEGY ?



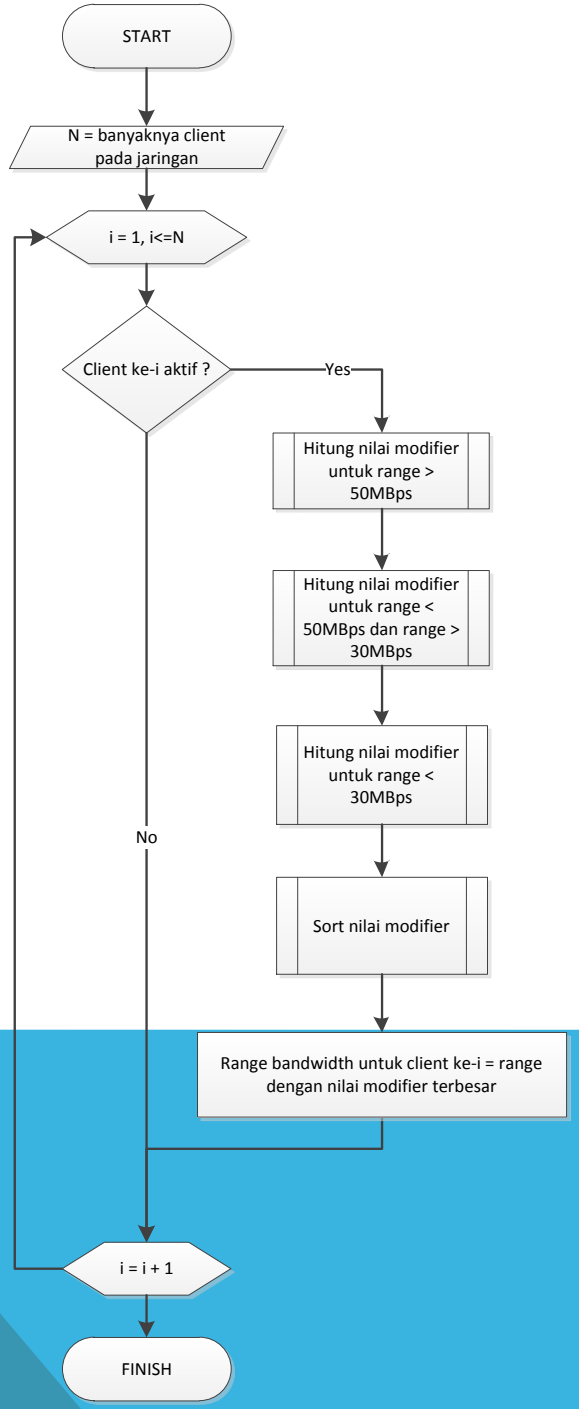

Select Constructions
London
NCO
Moscow
Chernob
Leguri
Siberia
Canada
Paris
R. Plains
Jalan
Gateseland
BJG
JPN
Hawaii
Isa Shima
G. Rome
Katheringrad
Nanjing
Micronesia
USJ
Washington
Turko
A B Chongqing
Chicago
Andromeda
N. Africa
Suez
Vietnam
Manila 2k
Rabaul
Amazon
Arabia
Kenya
Maderascal
Indo Curry
M. Tiger
SHK
S. Africa
Repair
40% HP recovery to ally ships in the region.
Quarry
x2 resource income.
Market
A fresh catch was just delivered.

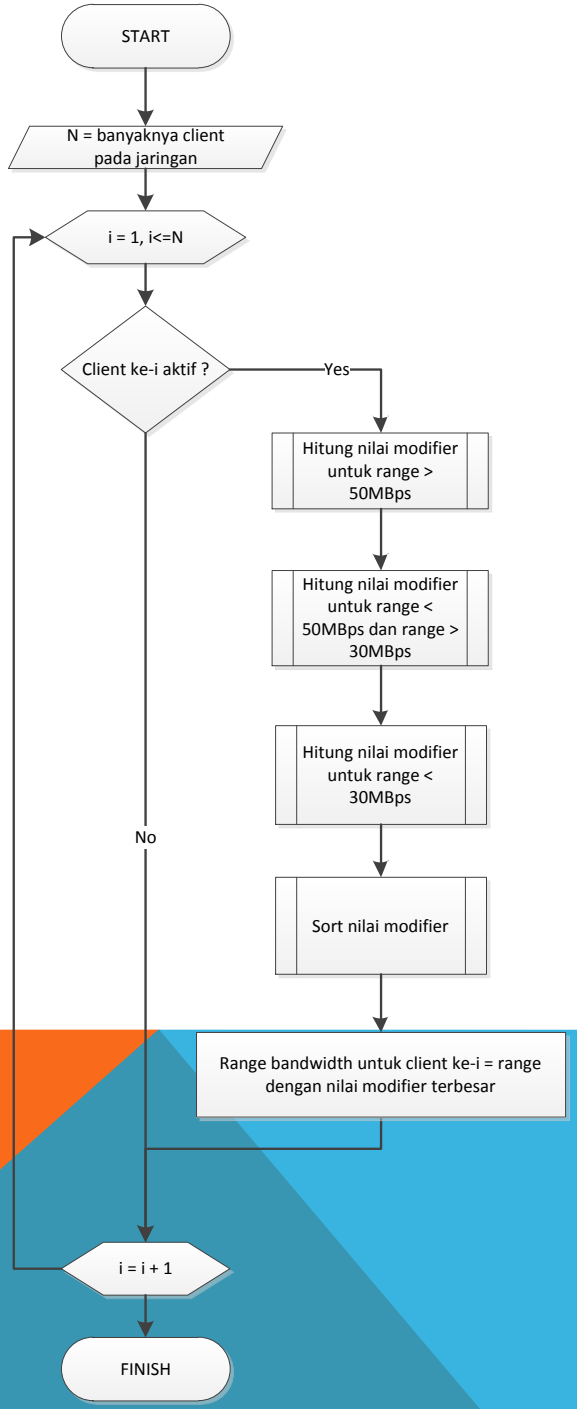
AI HARUS MEMASUKAN KONDISI YANG SEDANG DIALAMINYA DALAM PERHITUNGAN PENGAMBILAN KEPUTUSANNYA

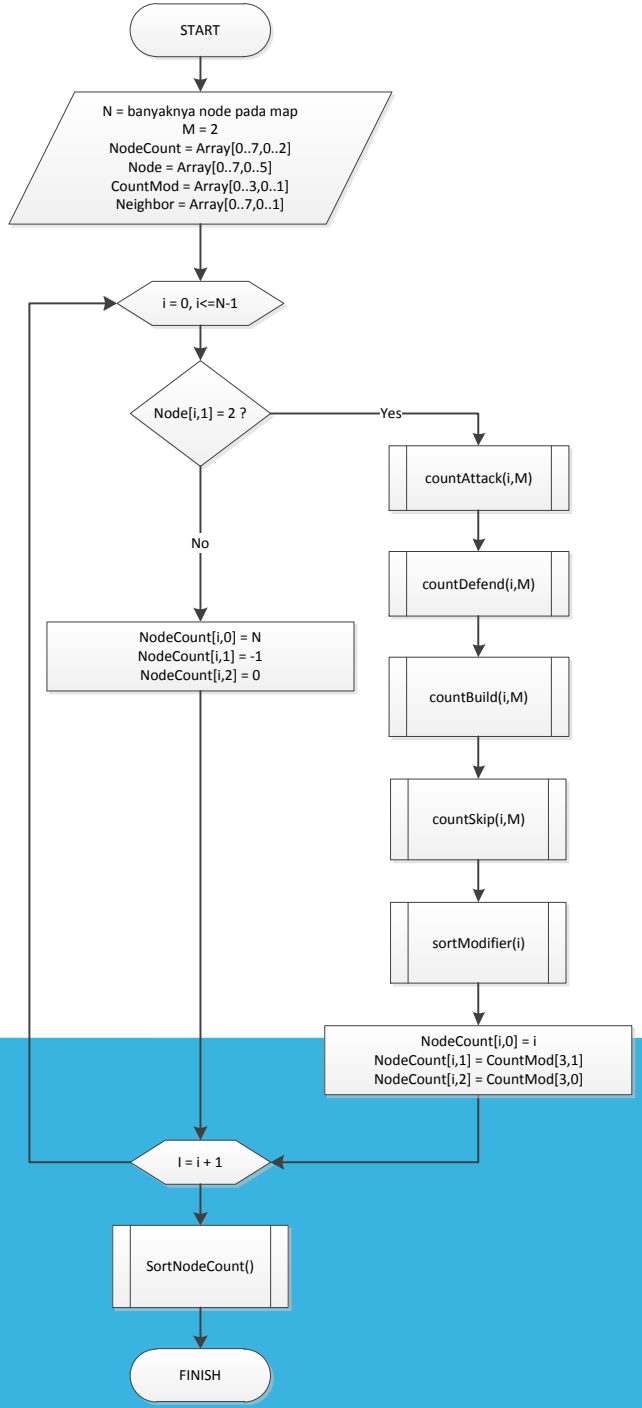
ALGORITMA RESOURCE ASSIGNMENT

ALGORITMA RESOURCE ASSIGNMENT

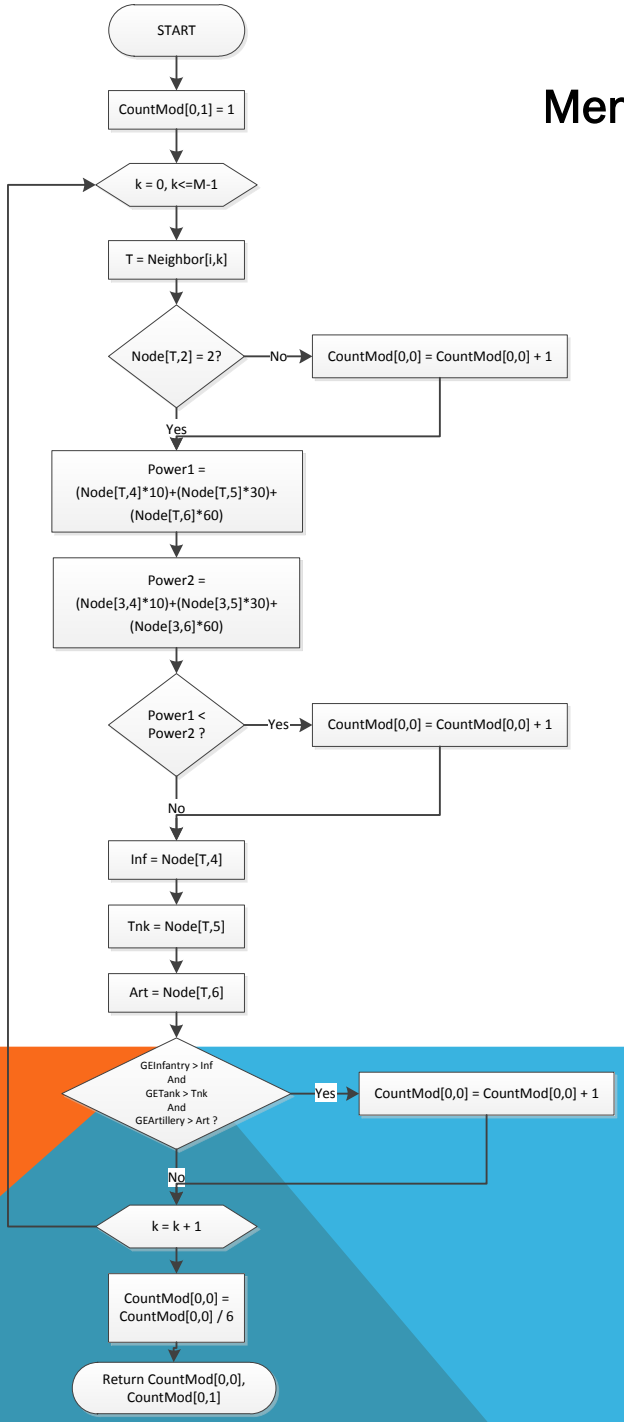
Mempunyai kemampuan untuk menghasilkan sebuah nilai berdasarkan beberapa nilai yang ada dengan diurutkan menggunakan skala prioritas.


START
N = banyaknya client pada jaringan
i = 1, i<=N
Client ke-i aktif ?
Yes
No
Hitung nilai modifier untuk range > 50MBps
Hitung nilai modifier untuk range < 50MBps dan range > 30MBps
Hitung nilai modifier untuk range < 30MBps
Sort nilai modifier
Range bandwidth untuk client ke-i = range dengan nilai modifier terbesar
i = i + 1
FINISH

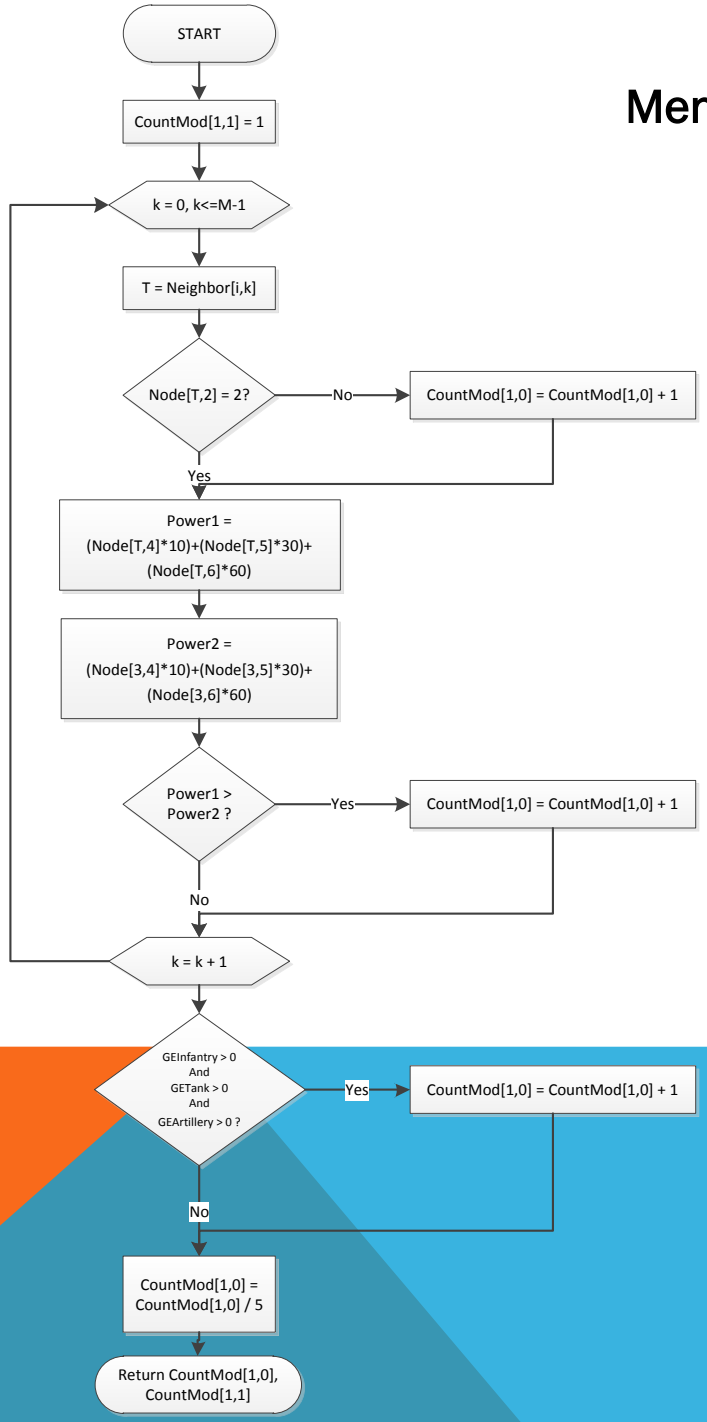

START
N = banyaknya client pada jaringan
i = 1, i<=N
Client ke-i aktif ?
Yes
No
Hitung nilai modifier untuk range > 50MBps
Hitung nilai modifier untuk range < 50MBps dan range > 30MBps
Hitung nilai modifier untuk range < 30MBps
Sort nilai modifier
Range bandwidth untuk client ke-i = range dengan nilai modifier terbesar
i = i + 1
FINISH


START
N = banyaknya node pada map
M = 2
NodeCount = Array[0..7,0..2]
Node = Array[0..7,0..5]
CountMod = Array[0..3,0..1]
Neighbor = Array[0..7,0..1]
i = 0, i<=N-1
Node[i,1] = 2 ?
Yes
No
countAttack(i,M)
countDefend(i,M)
countBuild(i,M)
countSkip(i,M)
sortModifier(i)
NodeCount[i,0] = N
NodeCount[i,1] = -1
NodeCount[i,2] = 0
NodeCount[i,0] = i
NodeCount[i,1] = CountMod[3,1]
NodeCount[i,2] = CountMod[3,0]
I = i + 1
SortNodeCount()
FINISH

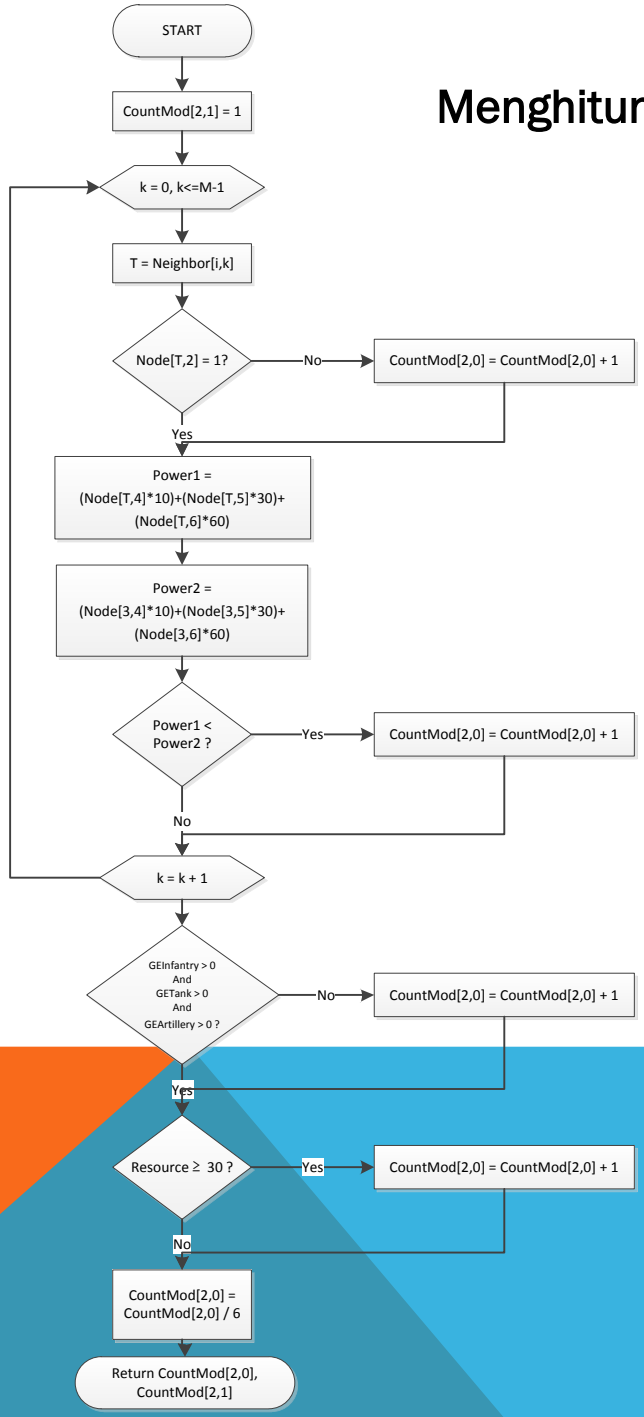
Menghitung modifier untuk keputusan menyerang



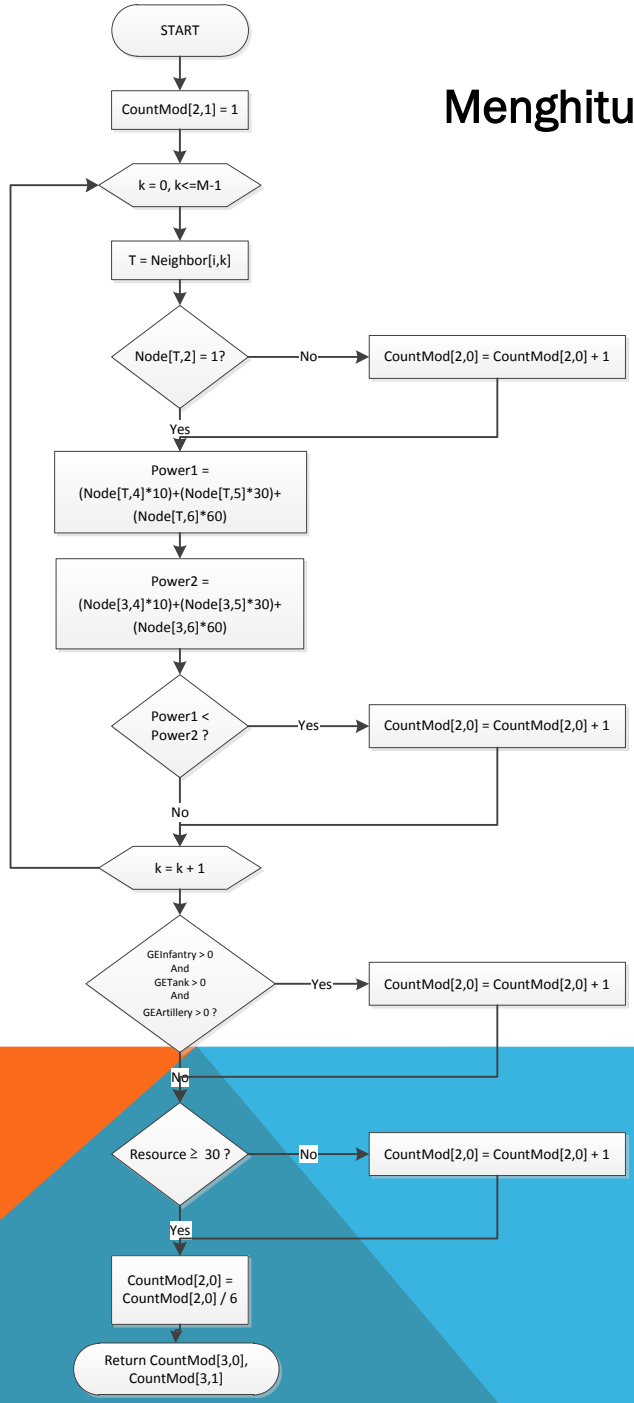
Menghitung modifier untuk keputusan bertahan



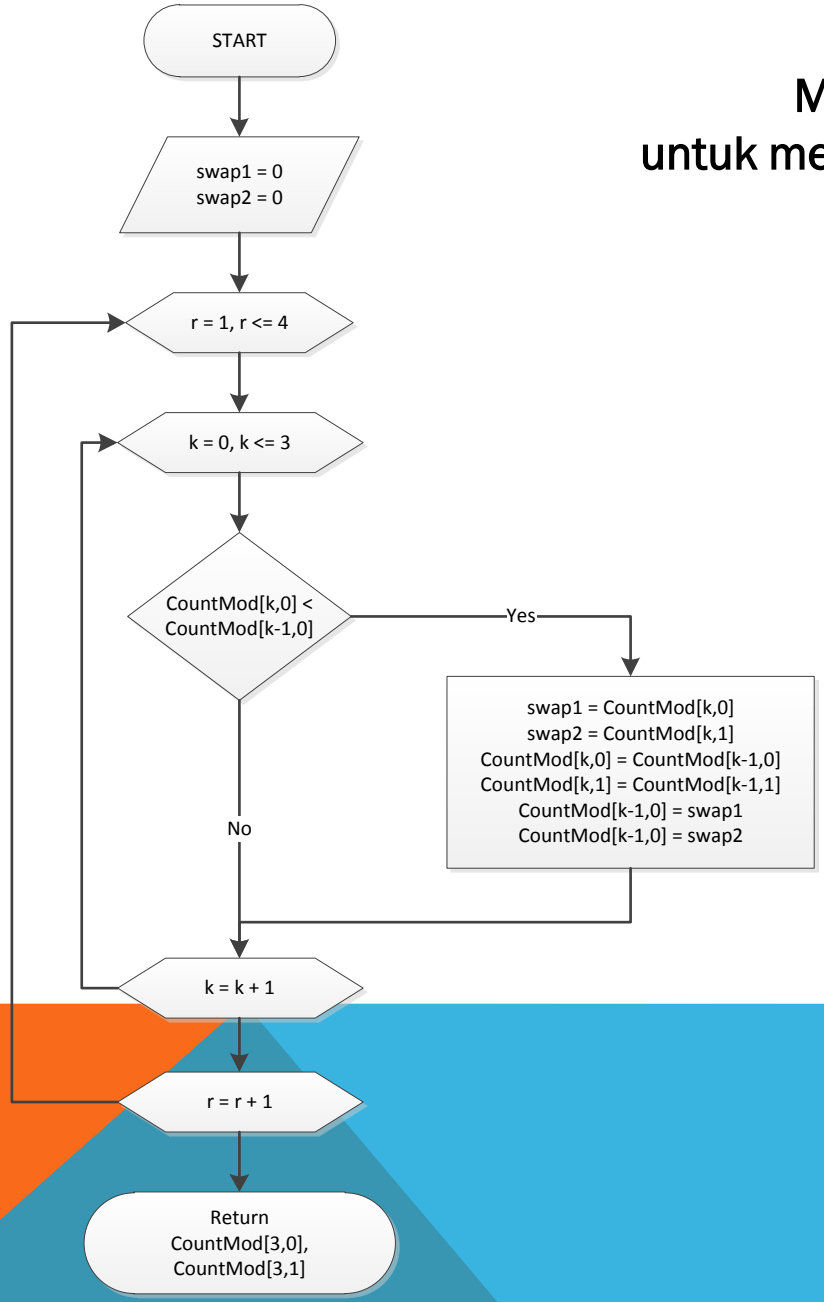
Menghitung modifier untuk keputusan memproduksi unit



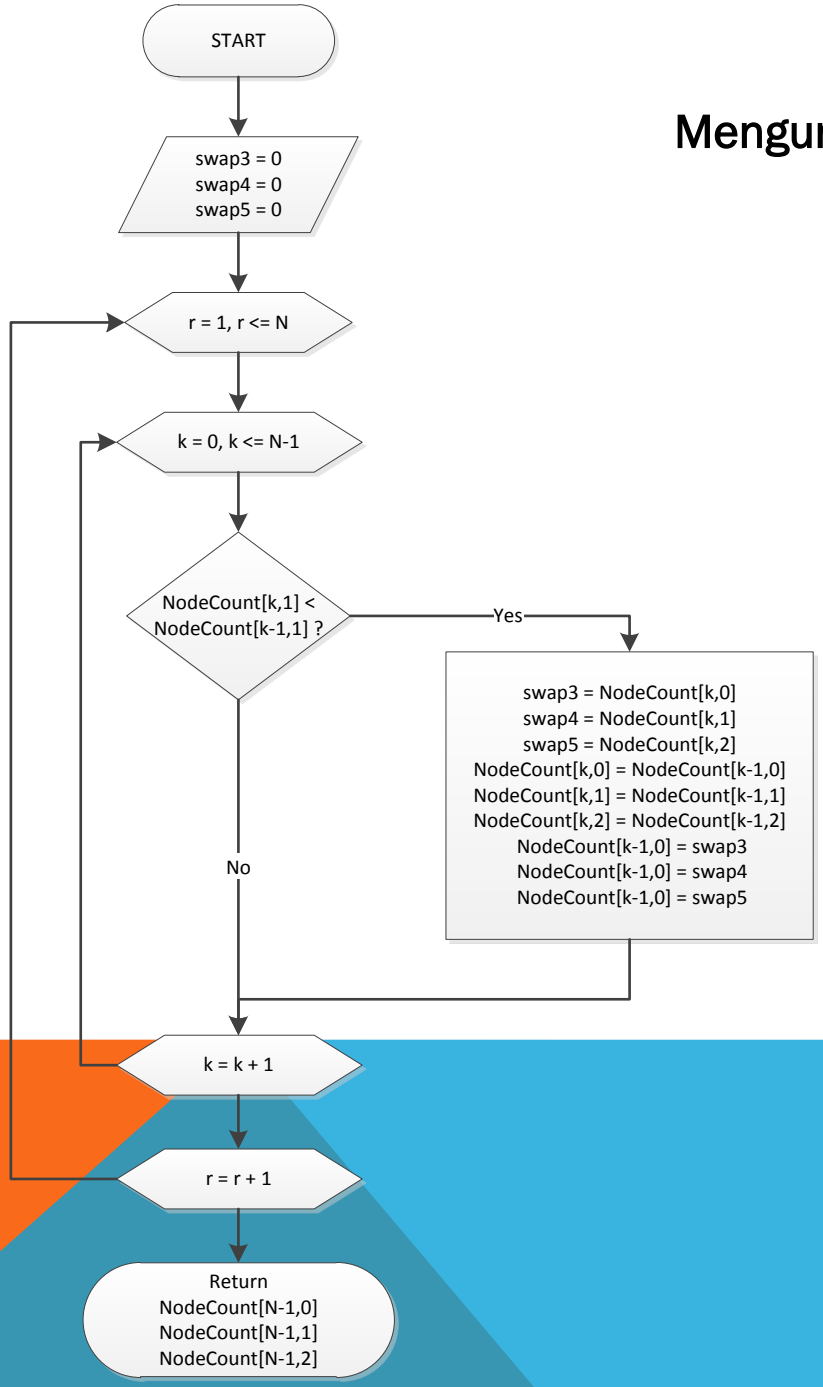
Menghitung modifier untuk keputusan melewatkan giliran



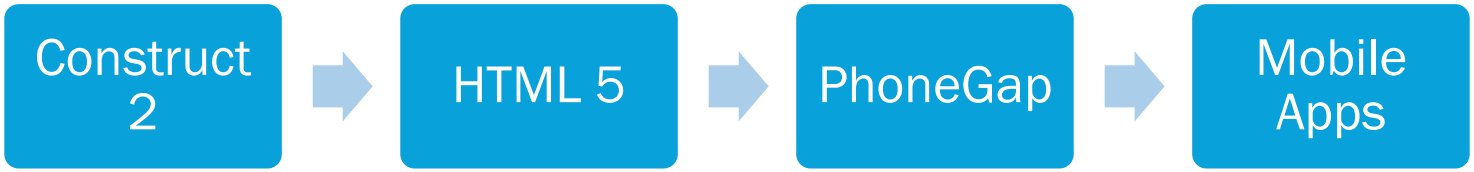
Mengurutkan setiap nilai modifier untuk menentukan keputusan pada sebuah node

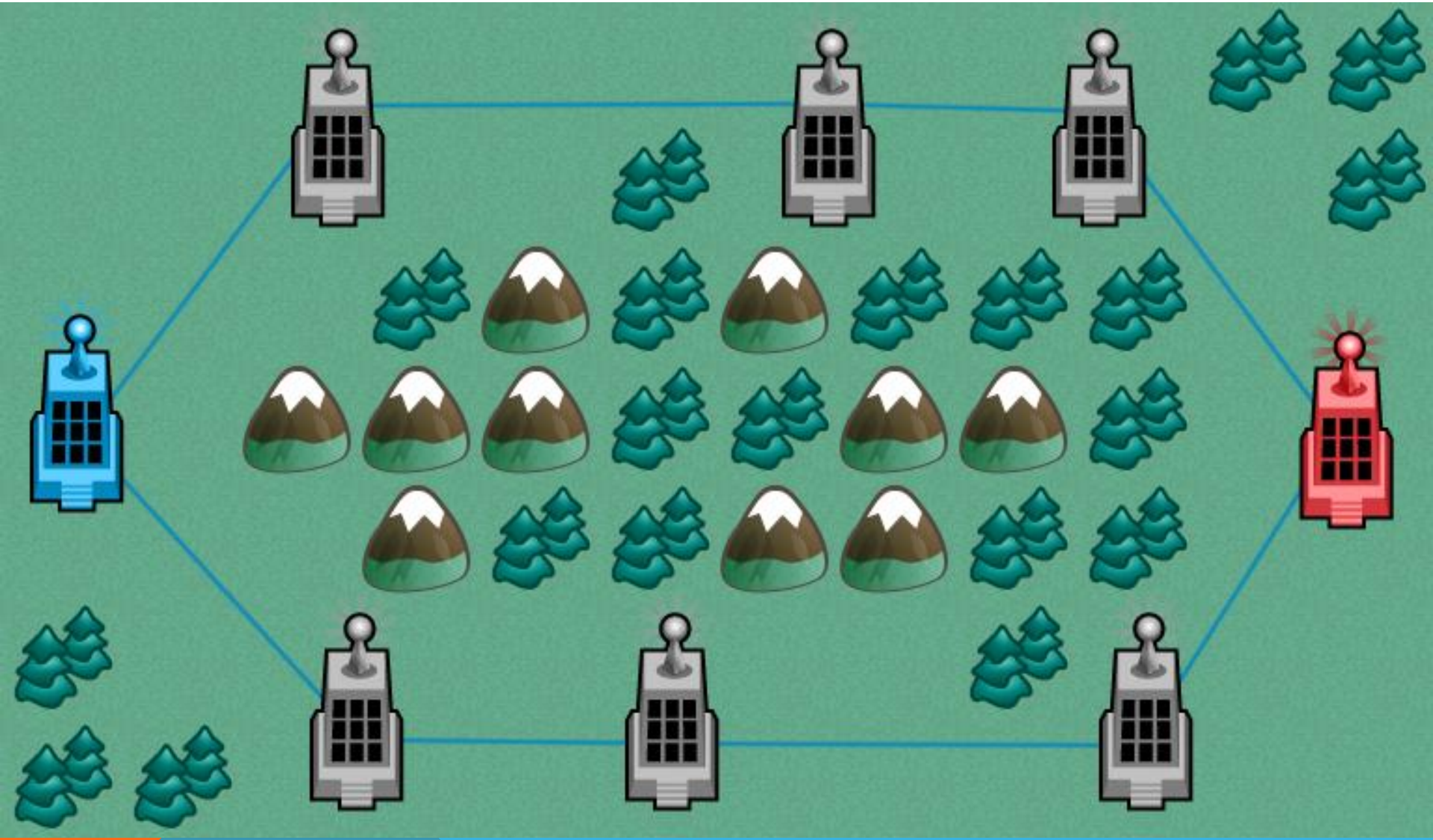


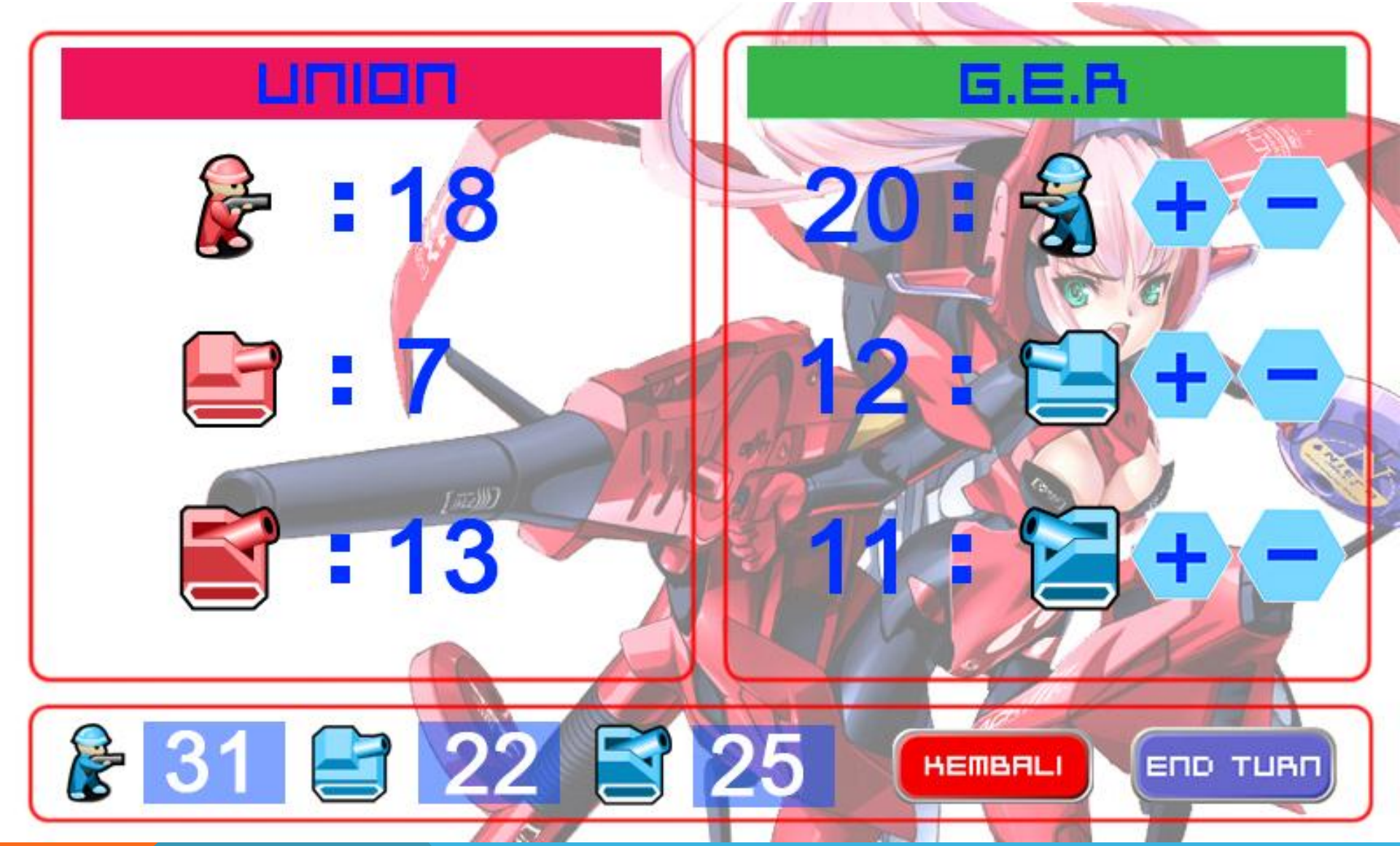
Mengurutkan nilai modifier pada setiap node

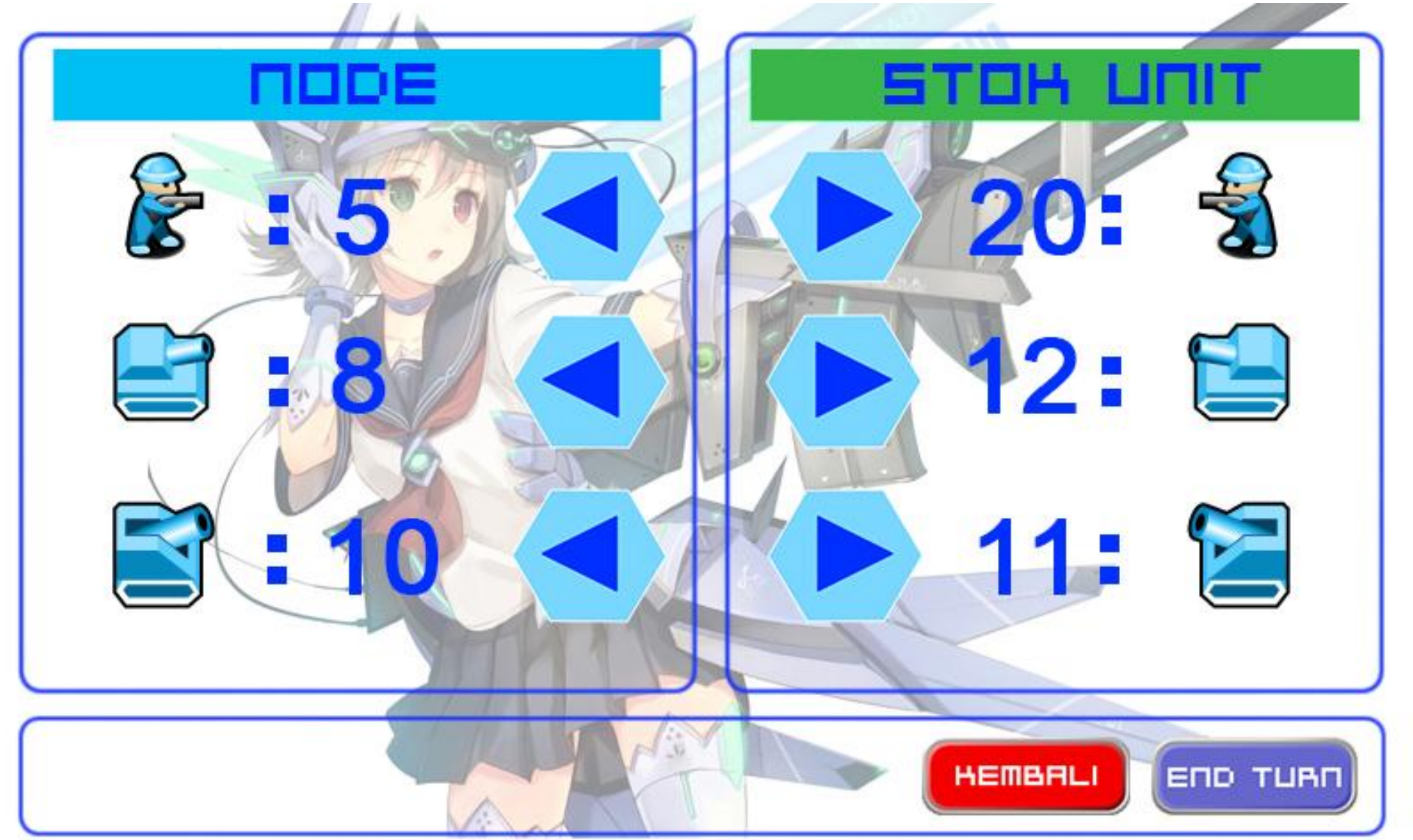


Implementasi pada perangkat mobile

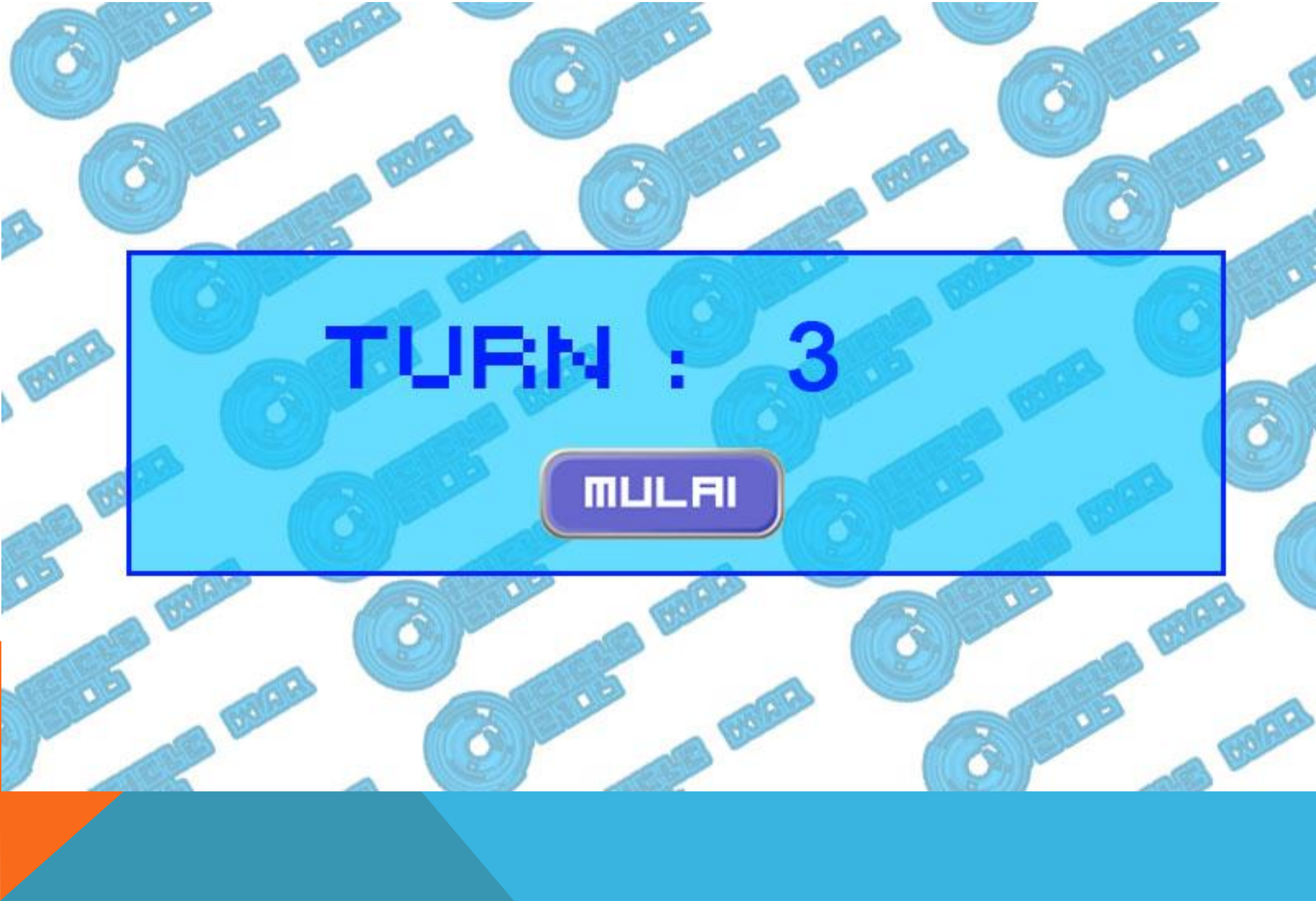





UNION
: 18
: 7
: 13
G.E.A
20 :
12 :
11 :
+
−
+
−
+
−
31
22
25
KEMBALI
END TURN


NODE
: 5
: 8
: 10
STOK UNIT
20 :
12 :
11 :
KEMBALI
END TURN


PRODUKSI
STOK UNIT
22 :
25 :
18 :
RESOURCE : 110
KEMBALI
END TURN


TURN : 3
MULAI

KALAH

SKOR ANDA :

 3

 2

 0

$: 40

KELUAR

MENANG !!!

SKOR ANDA :

 32  28  31

$: 270

KELUAR

DEMO PROGRAM

