Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung

Tugas Besar 1 IF2211 Strategi Algoritma

Semester II tahun 2025/2026

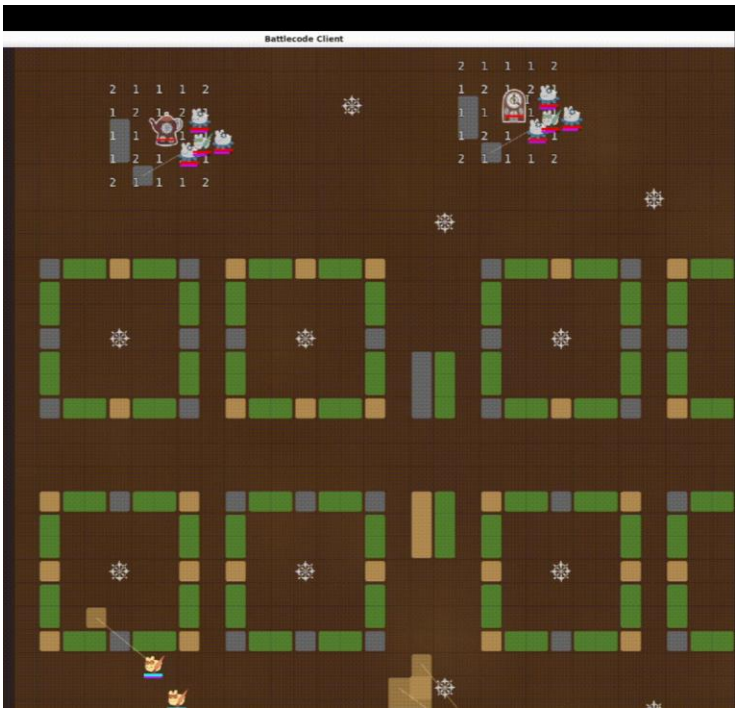
Pemanfaatan Algoritma *Greedy* dalam pembuatan bot permainan Battlecode 2025

Batas pengumpulan : Jumat, 13 Maret pukul 23.59

Arsip pengumpulan :

- Source program yang dapat dijalankan disertai README
- Laporan (soft copy)

Deskripsi Tugas



Gambar 1 : Battlecode 2025

Latar Belakang

Battlecode adalah permainan strategi waktu nyata di mana Anda akan menulis kode untuk pemain otonom. Pemain Anda harus mengelola pasukan robot secara strategis dan mengendalikan cara robot-robot Anda bekerja sama untuk mengalahkan tim lawan. Sebagai peserta, Anda akan belajar menggunakan kecerdasan buatan, penelusuran jalur, algoritma terdistribusi, dan komunikasi untuk membuat pemain Anda sesaing mungkin. Namun, Anda hanya memiliki sumber daya komputasi terbatas per giliran, jadi Anda harus merencanakan strategi, mengoptimalkan, dan berkompromi sesuai kebutuhan.
Pada Tugas Besar pertama Strategi Algoritma ini, mahasiswa diminta untuk membuat sebuah bot yang nantinya akan dipertandingkan satu sama lain. Tentunya mahasiswa harus menggunakan strategi ***greedy*** dalam membuat bot ini.

Komponen Permainan

Komponen-komponen dari permainan ini antara lain:

1. Objectives

   Objective utama adalah mewarnai peta. Tim pertama yang berhasil mewarnai lebih dari 70% petak yang dapat diwarnai di peta akan memenangkan permainan. Tim juga bisa menang dengan cara menghancurkan semua robot dan menara milik tim lawan.

2. Aturan Tiebreaker

   Jika setelah 2000 ronde tidak ada tim yang berhasil mewarnai lebih dari 70% petak yang dapat diwarnai, permainan akan langsung berakhir. Penentuan pemenang dilakukan dengan urutan tiebreaker berikut:

   - Luas area yang sudah diwarnai di peta
   - Jumlah menara sekutu yang masih hidup
   - Total jumlah uang (chips)
   - Jumlah total cat (paint) yang dimiliki semua robot dan menara
   - Jumlah robot sekutu yang masih hidup
   - Jika masih seri, tim dipilih secara acak (uniform random)

3. Peta

   Setiap permainan dimainkan di sebuah peta berbentuk grid persegi panjang 2 dimensi, dengan ukuran antara 20×20 hingga 60×60 petak (inklusif). Titik kiri bawah peta memiliki koordinat (0, 0); koordinat bertambah ke arah Timur (kanan) dan Utara (atas). Lokasi di peta direpresentasikan

dengan objek MapLocation yang menyimpan nilai x dan y. Agar tidak ada peta yang menguntungkan salah satu pemain, dijamin bahwa peta bersifat simetris (baik melalui rotasi maupun refleksi).

4. Jarak Pandang dan Pergerakan

   Robot selalu bisa melihat fitur peta dan robot lain dalam jarak hingga sqrt(20), kira kira 4.47 petak. Unit tidak bisa bergerak melewati menara atau robot lain. Peta mungkin memiliki tembok (walls), yang tidak bisa dilewati atau diwarnai. Tembok maksimal hanya 20% dari seluruh peta. Tembok tidak bisa dihapus atau dibangun oleh pemain. Robot juga tidak bisa melewati atau mewarnai reruntuhan (ruins).

5. Tower & Ruins

   Menara juga berfungsi sebagai zona spawn tim dan bisa menyerang unit musuh. Untuk membangun menara, tim harus mewarnai pola khusus di atas reruntuhan (ruins), zona 5×5 yang sudah ditentukan di peta *(lihat section towers untuk selengkapnya)*.
   Setiap permainan dimulai dengan sejumlah reruntuhan yang sudah ada. Reruntuhan tidak bisa dipindah atau dihancurkan. Pusat reruntuhan berjarak minimal 5 petak satu sama lain, dan zona 5×5 di sekitar pusat reruntuhan dijamin bebas tembok. Setiap tim memulai permainan dengan 2 menara: 1 menara cat dan 1 menara uang. Reruntuhan hanya bisa dilihat oleh robot atau menara jika berada dalam radius penglihatan.

   - Money Tower
   - Paint Tower
   - Defense Tower

   Secara default, setiap tim memulai permainan dengan 1 menara penghasil uang dan 1 menara penghasil cat, keduanya sudah di-upgrade ke level 2.

6. Pola Sumber Daya Khusus

   Mewarnai pola khusus tertentu di peta dapat meningkatkan produksi sumber daya tim. Pola ini disebut **Special Resource Pattern (SRP).** Setiap pola aktif yang berhasil diwarnai akan membuat semua menara tambang sekutu mendapatkan +3 sumber daya tambahan per giliran.
   Robot bisa menandai pola ini menggunakan fungsi **markResourcePattern(),** lalu menyelesaikannya dengan **completeResourcePattern()** (biaya 200 chips). Setelah pola selesai dibuat, pola tersebut harus tetap utuh (tidak rusak warnanya) selama 50 ronde agar menjadi aktif. Pola ini sama untuk semua peta dan permainan.

7. Cat

   Cat adalah sumber daya utama permainan. Digunakan untuk mengklaim wilayah dan menghitung skor. Setiap robot dan menara menyimpan cat secara individu, dengan kapasitas maksimal

tergantung tipenya. Robot yang cat-nya sedikit akan mengalami cooldown lebih lama atau bahkan tidak bisa beraksi. Cat bisa diisi ulang dari mopper atau ditarik dari menara penghasil cat. Setiap tipe robot memiliki serangan berbeda yang bisa:

- Menggunakan cat untuk meletakkan warna di petak, atau
- Menghapus cat milik lawan dari petak.

Robot bisa memilih antara warna utama atau warna sekunder tim untuk mewarnai petak. Kedua warna ini dianggap sama untuk keperluan skor, tapi dibedakan saat mengecek penyelesaian pola cat.

8. Chips

Chips digunakan untuk membangun unit & menara. Chips dihasilkan setiap ronde oleh menara penghasil uang, dan bisa digunakan oleh robot sekutu mana saja setelah dihasilkan. Setiap tim memulai permainan dengan 2500 chips.

9. Robot

Semua robot dapat melakukan aksi seperti bergerak, merasakan (sensing), dan berkomunikasi satu sama lain. Dalam setiap pertempuran, robot milikmu akan berhadapan dengan satu tim musuh. Permainan berjalan berbasis giliran (turn-based) dan dibagi menjadi ronde. Pada setiap ronde, setiap robot mendapat giliran untuk menjalankan kode dan melakukan aksi. Kode yang dijalankan robot menghabiskan bytecode (ukuran sumber daya komputasi, dijelaskan lebih lanjut pada section bytecode limit). Setiap robot hanya memiliki jumlah bytecode tertentu per giliran; jika habis, gilirannya langsung berakhir dan akan dilanjutkan pada giliran berikutnya. Jika robot sudah selesai menjalankan aksinya pada giliran tersebut, ia harus memanggil **Clock.yield()** untuk menunggu giliran berikutnya dimulai.
Semua robot memiliki HP (hitpoints / nyawa). Ketika HP mencapai nol, robot langsung dihapus dari permainan. Setiap robot memiliki ID unik yang diacak dan tidak kurang dari 10.000. Robot hanya dapat berinteraksi dengan lingkungan terdekat melalui sensing, pergerakan, dan kemampuan khusus. Setiap robot menjalankan salinan kode Anda secara independen. Robot tidak bisa berbagi variabel statis (masing-masing punya salinannya sendiri) karena dijalankan di JVM terpisah. Dua atau lebih robot tidak boleh berada di petak yang sama. Ketika cooldown pergerakan di bawah 10, robot dapat bergerak ke salah satu dari 8 petak tetangga. Semua robot memiliki cadangan cat (paint) dengan kapasitas maksimum tertentu. Semakin rendah cadangan cat, semakin tinggi

cooldown pergerakan dan aksi yang dihadapi. Aturan penalti cat rendah:

- Jika cadangan cat hanya X% penuh (X < 50), maka cooldown meningkat (100 – 2X)%.
- Jika cadangan cat habis sama sekali selama giliran, robot tidak bisa bergerak atau melakukan aksi apa pun (kecuali self-destruct) sampai mendapat cat lagi, dan akan kehilangan 20 HP per ronde.

Robot spawn dengan cadangan cat 100% penuh dan dapat diisi ulang di menara atau oleh mopper. Kehilangan cat saat mengakhiri giliran:

- Di wilayah netral : kehilangan 1 cat
- Di wilayah musuh : kehilangan 2 cat
- Di wilayah sekutu : tidak kehilangan cat

Penalti cat tambahan setiap ronde adalah 1 × jumlah robot sekutu yang bersebelahan (dalam 8 petak sekitar). Pinalti ini berlipat ganda (×2) saat berada di wilayah musuh. Robot dapat di-spawn dalam radius aksi √4 (≈ 2 petak) dari menara sekutu, dengan biaya cat dan chips sesuai jenisnya.

10. Jenis Robot

Soldier

- HP: 250
- Kapasitas cat maks: 200
- Biaya produksi: 200 cat + 250 chips
- Serangan: Menyerang satu petak (radius aksi 3)
    - Jika petak kosong atau sudah ada cat sekutu → mewarnai petak tersebut
    - Menimbulkan 50 damage ke menara musuh (tidak ke robot)
- Biaya serangan: 5 cat
- Cooldown aksi: 10

Splasher

- HP: 150
- Kapasitas cat maks: 300
- Biaya produksi: 300 cat + 400 chips
- Serangan: Menyerang area melingkar (radius 2 dari titik pusat)
    - Titik pusat serangan maksimal 2 petak dari posisi splasher
    - `Dalam radius √2 (~1.41) dari pusat → bisa menimpa cat musuh`
    - Mewarnai semua petak kosong / sekutu di area
    - Menimbulkan 100 damage ke menara musuh (tidak ke robot)
- Biaya serangan: 50 cat
- Cooldown aksi: 50

Mopper

- HP: 50
- Kapasitas cat maks: 100
- Biaya produksi: 100 cat + 300 chips
- Kemampuan utama:
    1. `Mop petak (radius √2)`
        - Menghapus cat musuh di petak tersebut
        - Jika ada robot musuh di petak → musuh kehilangan 10 cat, mopper mendapat 5 cat
        - Cooldown: 30
    2. `Transfer cat ke robot atau menara sekutu (radius √2)`
        - Mentransfer jumlah cat tertentu dari cadangan sendiri
        - Cooldown: 10
    3. Ayunan mop (swing) ke salah satu dari 4 arah utama (atas, bawah, kiri, kanan)
        - Menghapus 5 cat dari masing-masing robot musuh yang berdekatan dalam garis lurus
        - Serangan diterapkan dua kali di arah yang sama: → 1 langkah ke arah tersebut → 2 langkah ke arah yang sama
        - Maksimal 6 target potensial (3 di langkah 1 + 3 di langkah 2)
        - Cooldown: 20

Catatan tambahan: Mopper mendapat penalti cat ganda saat bergerak atau mengakhiri giliran di wilayah musuh.

Ability dan Statistik Robot

Name	Health	Paint Capacity	Paint Cost	Money Cost	Attack Cost	Attack Radius	Attack Damage	Attack Cooldown
Soldier	250	200	200	250	5	3	50 health to towers	10
Splasher	150	300	300	400	50	2	100 health to towers	50
Mopper	50	100	100	300	0	$\sqrt{2}$	-10 paint to robots	30

Untuk informasi lebih lengkap, silahkan buka dokumentasi Battlecode pada link berikut.

11. Towers
    - Prosedur Pembangunan Menara

      Semua menara dibangun dengan prosedur yang sama. Menara hanya bisa dibangun di atas reruntuhan (ruins), setelah pola cat 5×5 tertentu ditempatkan di sekitar reruntuhan tersebut.

      1. Panggil markTowerPattern() untuk secara otomatis menempatkan penanda di area 5×5 sekitar reruntuhan. Penanda ini menunjukkan posisi mana yang harus diwarnai dengan warna utama (primary color).
         - Biaya: 25 cat
      2. Setelah pola sudah diwarnai, robot bisa memanggil completeTowerPattern() untuk memunculkan menara yang ditentukan di lokasi reruntuhan.
         - Menara akan tetap hidup sampai HP-nya mencapai 0, meskipun pola cat di sekitar reruntuhan kemudian dihapus.
         - Catatan: Pola menara bisa diselesaikan tanpa memanggil markTowerPattern() terlebih dahulu, asalkan pola cat utama dan sekunder sudah cocok.

- Posisi dan Ukuran Menara

Menara yang dihasilkan akan menempati blok 1×1 di tengah bentuk spawn. Bentuk berbeda menghasilkan jenis menara berbeda dengan rentang serangan, laju serang, laju penambangan, dan kesehatan yang berbeda (lihat tabel di bawah).

Level dan Upgrade

- Menara selalu dibangun di level 1.
- Bisa di-upgrade ke level lebih tinggi menggunakan upgradeTower() dalam radius sqrt(2) (~1.41 petak).

Batasan

- Tidak ada tim yang boleh memiliki lebih dari 25 menara sekaligus.
- Setiap menara bisa menyimpan 1000 cat.

Bonus dari Menara Pertahanan

Setiap menara pertahanan yang hidup meningkatkan kekuatan serangan satu blok (single-block attack) dari semua menara sekutu sebesar:

- +5 (level 1)
- +7 (level 2)
- +9 (level 3)

Bonus ini tidak memengaruhi serangan area (AoE) menara sekutu.

Name	Build/ Upgrade cost	Attack Range	Single-block Attack Strength	AoE Attack Strength	Base Mining Rate	Health
Money Lv1	1000 chips	3	20	10	20 chips/turn	1000
Lv2	2500 chips	3	20	10	30 chips/turn	1500
Lv3	5000 chips	3	20	10	40 chips/turn	2000
Paint, Lv1	1000 chips	3	20	10	5 paint/turn	1000
Lv2	2500 chips	3	20	10	10 paint/turn	1500
Lv3	5000 chips	3	20	10	15 paint/turn	2000
Defense, Lv1	1000 chips	4	40	20	20 chips/attack*	2000
Lv2	2500 chips	4	50	25	30 chips/attack*	2500
Lv3	5000 chips	4	60	35	40 chips/attack*	3000

12. Serangan

Menara dapat menyerang unit musuh. Menara memiliki dua jenis serangan:

- Serangan satu petak (single-block attack) yang menyerang satu unit musuh di satu petak.
- Serangan area efek (AoE / area of effect) jarak jauh yang memberikan kerusakan kepada semua unit musuh di dalam jangkauannya.

Setiap giliran, satu menara dapat melakukan satu serangan satu petak dan satu serangan area efek.

13. Penambangan

Setiap giliran, menara penghasil uang dan menara penghasil cat akan secara pasif menambang sumber daya.

- Cat disimpan di dalam menara dan dapat diambil oleh robot sekutu.
- Uang (chips) tersedia untuk semua unit sekutu.

Setiap menara memulai dengan 500 cat untuk digunakan spawn berbagai jenis unit. Menara dapat memunculkan robot dalam jarak 2 petak dari posisinya. Jika cat di menara habis, menara tidak akan bisa spawn unit lagi sampai diisi ulang atau menghasilkan cat sendiri. Cat dapat diisi ulang menggunakan cat yang dikumpulkan oleh mopper.

14. Komunikasi

Robot hanya bisa melihat lingkungan di sekitarnya secara langsung dan dikendalikan secara independen oleh salinan kode yang sama, sehingga koordinasi antar-robot sangat sulit. Anda tidak dapat berbagi variabel apa pun di antara mereka; bahkan variabel static pun tidak akan dibagikan karena setiap robot memiliki salinannya sendiri.

Komunikasi dilakukan dengan dua cara:

- Pesan (Messages) Pesan dapat berisi informasi apa saja, tetapi dibatasi hanya 4 byte.
- Penanda (Markers) Penanda menyatakan warna apa yang seharusnya digunakan untuk mewarnai ulang suatu petak, dan penanda ini terlihat oleh semua robot sekutu. Penanda tidak memberikan efek lain apa pun dalam permainan.

Pesan

Robot atau menara yang berada dalam jangkauan satu sama lain dapat saling mengirim pesan. Sebuah robot dianggap “dalam jangkauan” (in range) sebuah menara jika memenuhi dua syarat sekaligus:

- `Jaraknya ke menara tersebut ≤ sqrt(20) (~4.47 petak), dan`
- Robot tersebut terhubung dengan menara melalui jalur cat sekutu (artinya robot bisa bergerak sampai ke menara hanya dengan berjalan di petak-petak yang sudah memiliki cat milik timnya sendiri).

The robots and towers who are connected by paint can talk, but the robot who is not on paint cannot talk to the tower.

Setiap robot dan menara menyimpan semua pesan yang diterima dalam 5 ronde terakhir di dalam buffer-nya. Setelah 5 ronde berlalu, pesan tersebut akan dihapus secara otomatis.

Batas Pengiriman Pesan per Ronde adalah sebagai berikut

- Robot: maksimal 1 pesan per ronde
- Menara: maksimal 20 pesan per ronde

Setiap pesan terdiri dari:

- Satu bilangan bulat 32-bit (integer)
- Nomor ronde saat pesan dikirim

- ID robot yang mengirim pesan

Menara memiliki kemampuan tambahan untuk melakukan broadcast (siaran) pesan ke menara sekutu lain yang berada dalam jarak sqrt(80) (~8.94 petak).

- Menara tidak perlu terhubung melalui jalur cat untuk melakukan broadcast ini.
- Setiap kali broadcast dilakukan, itu dihitung sebagai satu pesan terhadap batas 20 pesan per ronde menara.
- Kemampuan ini memungkinkan menara memperluas jangkauan komunikasi jarak jauh antar-menara.

Penanda

Penanda adalah informasi yang bisa ditempatkan atau dihapus di peta oleh robot. Penanda hanya memiliki dua jenis:

- Menandai dengan warna sekunder
- Menandai dengan warna utama

Penanda hanya terlihat oleh robot sekutu dan akan tetap ada di petak tersebut sampai dihapus atau diganti dengan penanda lain oleh sekutu. Selain itu, penanda hanya dapat ditempatkan pada ubin yang dapat dicat

Penanda dapat diletakkan dengan cara

- `Secara manual: robot dapat menempatkan penanda di petak yang berjarak ≤ sqrt(2) (~1.41 petak) dari posisi robot saat ini. Akan dijelaskan lebih lanjut pada section Action`
- Secara otomatis melalui fungsi pola:
    - markTowerPattern() → menandai pola 5×5 di sekitar reruntuhan dengan penanda yang sesuai untuk membangun menara
    - markResourcePattern() → menandai pola 5×5 untuk Special Resource Pattern (SRP)

15. Action and Cooldown

Robot melakukan berbagai aksi untuk berinteraksi dengan dunia permainan. Beberapa aksi tidak bisa dilakukan berkali-kali dalam satu giliran atau dalam waktu singkat karena terikat cooldown. Berikut daftar aksi utama:

- Menyerang/mewarnai

  Menyerang dan mewarnai dilakukan dengan fungsi yang sama. Aksi ini hanya bisa dilakukan jika action cooldown robot < 10. Action cooldown robot dapat dicek dengan

fungsi isActionReady(). Dilakukan dengan memanggil metode attack() yang akan menaargetkan satu petak yang dapat dilihat oleh robot tersebut. Efeknya tergantung tipe robot (seperti yang dijelaskan di bagian Robot): memberikan damage atau mewarnai petak. Setelah dilakukan, action cooldown robot bertambah sesuai nilai yang ditentukan oleh tipe robot.

- Bergerak (moving)

  Hanya bisa dilakukan jika movement cooldown < 10. Movement cooldown dapat dicek dengan isMovementReady(). Petak tujuan harus kosong (tidak ditempati unit lain) dan bisa dilewati (passable). Dilakukan dengan memanggil move() dengan arah yang diinginkan. Setelah bergerak, movement cooldown bertambah 10. Untuk memeriksa apakah gerakan valid, gunakan canMove().

- Menandai (marking)

  Semua robot bisa menandai petak tempat robot sedang berdiri, atau Menghapus penanda sekutu dari petak tempat robot berdiri. Biaya dari aksi ini adalah 1 cat. Aksi ini tidak menambah action cooldown sama sekali. Aksi dapat dilakukan dengan mark() atau removeMark(). Penanda tidak memengaruhi kepemilikan wilayah (tidak membuat petak menjadi wilayah sekutu). Penanda hanya bisa dirasakan oleh robot sekutu melalui senseNearbyMapInfos(). Penanda secara umum tersebut hanya boleh ditempatkan pada petak yang dapat diwarnai (tidak ada tembok atau reruntuhan). Selain itu, terdapat Fungsi khusus penanda, yakni markTowerPattern() dan markResourcePattern()menggunakan 25 cat dan menandai area 5×5 dengan pola yang sesuai.

- Mentransfer Cat (Transferring)

  Hanya dapat dilakukan oleh robot mopper, dan hanya dapat dilakukan jika action cooldown < 10 (dapat dicek dengan isActionReady()). Aksi ini dapat dilakukan dengan memanggil prosedur transferPaint(). Jarak maksimal: √2 (~1.41 petak). Mentransfer jumlah cat tertentu ke target (robot atau menara sekutu). Jika jumlah yang diberikan negatif, maka mopper justru mengambil cat dari target. Setelah dilakukan, action cooldown bertambah sesuai nilai yang disebutkan di deskripsi mopper.

- Menarik cat dari menara (withdrawing)

  Semua robot bisa menarik cat dari menara sekutu untuk mengisi ulang cadangan cat mereka. Memerlukan action cooldown < 10.

Setelah dilakukan, action cooldown bertambah 10. Jarak maksimal: √2 (~1.41 petak). Dilakukan dengan transferPaint() menggunakan nilai negatif untuk menunjukkan jumlah yang ingin ditarik.

- Spawn Robot (spawning)

  Hanya bisa dilakukan jika action cooldown menara < 10 (dicek dengan isActionReady()). Dilakukan dengan buildRobot(). Mengurangi biaya robot (cat + chips) dari stok menara. Memunculkan satu robot tipe tertentu di petak yang ditentukan. Setelah dilakukan, action cooldown menara bertambah 10.

- Berkomunikasi (communitating)

  Robot menggunakan sendMessage() untuk mengirim pesan ke menara sekutu dalam jangkauan. Menara menggunakan sendMessage() untuk mengirim pesan ke robot sekutu dalam jangkauan. Robot menggunakan readMessages() untuk membaca pesan baru yang diterima dalam 5 ronde terakhir dari buffer. Kedua metode ini tidak menambah action cooldown. Syarat komunikasi: kedua unit harus berada di petak yang terhubung melalui jalur cat sekutu.

- Menghancurkan Diri Sendiri (Disintegrating)

  Robot apa pun bisa memanggil disintegrate(). Menghancurkan robot tersebut secara instan. Tidak ada cooldown khusus untuk aksi ini.

Batasan Program

- Bytecode

Jumlah komputasi yang boleh dilakukan oleh robot pada setiap giliran sangat terbatas. Bytecode menjadi ukuran yang praktis untuk menghitung komputasi di bahasa pemrograman seperti Java, di mana satu bytecode Java kira-kira setara dengan satu operasi dasar seperti pengurangan atau pengambilan field, dan satu baris kode biasanya mengandung beberapa bytecode (untuk detail lebih lanjut bisa dilihat di tautan tersebut). Karena bytecode merupakan bagian dari kode yang sudah dikompilasi, program yang sama akan selalu menghasilkan bytecode yang identik dan karena itu membutuhkan jumlah komputasi yang persis sama pada input yang sama. Keunggulan ini sangat membantu karena kita bisa menghindari penggunaan waktu sebagai ukuran komputasi, yang sering menimbulkan masalah seperti perilaku tidak deterministik. Dengan adanya batas bytecode, menjalankan ulang pertandingan yang sama antara bot yang sama akan selalu menghasilkan hasil yang persis identik, fitur yang sangat berguna untuk proses debugging.

Setiap ronde, setiap robot mengambil gilirannya secara berurutan. Jika sebuah robot mencoba melebihi batas bytecode-nya (biasanya terjadi tanpa disengaja, misalnya karena ada loop yang terlalu besar atau hal serupa), komputasinya akan langsung dihentikan sementara dan kemudian dilanjutkan kembali tepat dari titik itu pada giliran ronde berikutnya. Kode akan berjalan lagi dengan normal, tetapi hal ini bisa menimbulkan masalah. Misalnya, robot sempat memeriksa apakah sebuah petak masih kosong, lalu komputasinya terpotong, robot lain bergerak dulu, dan ketika giliran robot itu dilanjutkan ia mencoba bergerak ke petak yang sekarang sudah ditempati orang lain. Untuk menghindari situasi seperti ini, gunakan fungsi Clock.yield() untuk mengakhiri giliran robot secara sengaja. Fungsi ini akan menghentikan komputasi tepat di tempat yang kamu inginkan, lalu melanjutkannya dari baris kode berikutnya pada giliran ronde selanjutnya.

Batas bytecode untuk semua robot adalah 17500, sedangkan batas bytecode untuk semua menara adalah 20000.

- Package

Pemain dapat menggunakan kelas dari paket mana pun yang tercantum dalam AllowedPackages.txt, kecuali kelas yang tercantum dalam DisallowedPackages.txt. File-file ini dapat ditemukan di sini.
Selain itu, pembatasan berikut juga berlaku:

- `Object.wait, Object.notify, Object.notifyAll, Class.forName,` dan `String.intern` tidak diperbolehkan.
- `java.lang.System` hanya mendukung `out, arraycopy, dan getProperty.`
- Selain itu, `getProperty` hanya boleh digunakan untuk mengambil properti dengan nama yang diawali dengan "`bc.testing.`".
- `java.io.PrintStream` tidak boleh digunakan untuk membuka file.

Perlu diperhatikan bahwa pelanggaran terhadap salah satu pembatasan di atas akan menyebabkan robot meledak saat dijalankan, meskipun file sumber berhasil dikompilasi tanpa masalah.

- Memory

Robot harus menjaga penggunaan memori tetap wajar. Jika sebuah robot menggunakan lebih dari 8 MB heap space selama turnamen atau pertandingan scrimmage, robot tersebut dapat meledak.

- Execution Time

Robot harus menjaga waktu eksekusinya tetap wajar. Walaupun penghitungan bytecode memang dimaksudkan untuk tujuan ini, beberapa penggunaan Java yang sangat canggih dapat menyebabkan robot berjalan sangat lambat meskipun masih berada dalam batas bytecode.

Untuk mengurangi beban pada server, diberlakukan batas total waktu eksekusi per tim. Waktu ini dihitung secara kumulatif untuk semua robot dalam satu pertandingan, dan akan di-reset antar pertandingan dalam satu game.

Jika sebuah tim melebihi batas ini, tim tersebut akan otomatis menyerah dan kalah dalam game. Nilai batas ini dapat diakses melalui konstanta game, dan Anda dapat mengecek berapa banyak waktu yang telah berlalu melalui kelas Clock.

Hal ini seharusnya tidak mempengaruhi sebagian besar pemain. Jika Anda mendapati bahwa Anda mencapai batas tersebut, kemungkinan ada optimasi yang perlu dilakukan pada kode Anda.

Petunjuk Development

Untuk tugas besar ini, game engine yang akan digunakan ***sudah dimodifikasi*** oleh asisten. Source Code untuk game engine dan template bot telah disediakan pada tautan berikut.

Berikut adalah panduan untuk mengerjakan tugas ini.

1. Clone repository yang disediakan asisten dengan command berikut

```
git clone https://github.com/Fariz36/STIMA-battle
cd STIMA-battle
```

2. Bot akan di develop pada directory tersebut
3. Jalankan command berikut

```
./gradlew build
cd client
```

4. Nantinya akan ada aplikasi yang dapat di run. Silahkan run aplikasi tersebut
5. Setelah aplikasi di-run, pilihlah direktori "STIMA-battle" sebagai directory root **(bukan STIMA-battle/src)**

Catatan: pastikan telah menginstall gradle, untuk dokumentasi terkait hal tersebut dapat diakses pada link berikut.

Spesifikasi Wajib

- Buatlah 3 bot (1 utama dan 2 alternatif) dalam bahasa Java yang mengimplementasikan algoritma Greedy pada bot permainan battlecode dengan tujuan memenangkan permainan

- Strategi greedy yang diimplementasikan setiap kelompok harus dikaitkan dengan fungsi objektif dari permainan ini, yaitu memenangkan permainan dengan mewarnai peta sebanyak mungkin. Buatlah strategi greedy terbaik, karena setiap bot dari masing-masing kelompok akan diadu dalam kompetisi Tubes 1.
- Strategi greedy yang diimplementasikan harus berbeda untuk setiap bot yang diimplementasikan dan setiap strategi greedy harus menggunakan heuristic yang berbeda.
- **Bot yang dibuat TIDAK BOLEH sama dengan SAMPEL yang diberikan sebagai CONTOH ataupun bot yang telah digunakan di kompetisi asli.**
- Tugas dikerjakan **berkelompok** dengan anggota **3 orang,** *boleh lintas kelas maupun kampus*. Setiap kelompok harap mengisi nama kelompok dan anggotanya pada link berikut, dengan maksimal tanggal **Minggu, 1 Maret 2026. Nama kelompok dilarang mengandung unsur SARA, penghinaan, provokasi, atau hal-hal lain yang bertentangan dengan norma hukum.**
- Perlu diperhatikan bahwa kelompok tubes untuk mata kuliah ini tidak boleh sama. Jadi jika berkelompok dengan X pada tubes ini, maka pada tubes selanjutnya anda tidak diperkenankan berkelompok dengan X.

Spesifikasi Bonus

- (maks 6) **[Video]** Membuat video tentang aplikasi greedy pada bot serta simulasinya pada game kemudian mengunggahnya di Youtube. Video dibuat harus memiliki audio dan menampilkan wajah dari setiap anggota kelompok. Untuk contoh video tubes stima tahun-tahun sebelumnya dapat dilihat di Youtube dengan kata kunci "Tubes Stima", "strategi algoritma", "Tugas besar stima", dll. Semakin menarik video, maka semakin banyak poin yang diberikan.
- (maks 10) [**The Best Bot]** Beberapa kelompok pemenang lomba kompetisi akan mendapatkan nilai tambahan berdasarkan posisi yang diraih.
- (maks 7) [***Post Mortem Analysis***] Setelah kompetisi selesai, silahkan membuat *post mortem analysis (an examination of a dead body to determine the cause of death)*. Poin dapat menjadi semakin tinggi apabila analisis yang dilakukan semakin mendetail. Analisis dapat mencakup kesalahan apa yang menyebabkan kekalahan, kesalahan bot decision making flaw, kesalahan non-teknis (version control, kerja sama, etc), solusi pengembangan bot kedepannya, apa yang dapat diperbaiki, lesson learned, dan lain sebagainya.

Petunjuk Pengerjaan

- Program disimpan dalam repository dengan nama **Tubes1_NamaKelompok** dengan nama kelompok sesuai pada sheet kelompok tertera. Dengan struktur seperti berikut.

```
├── src/
│   ├── main-bot/…
│   └── alternative-bots-1/
│   └── alternative-bots-2/
├── doc/
│   └── laporan.pdf
└── README.md
```

1. Folder src berisi semua source code
2. Folder doc berisi laporan tugas besar dengan format NamaKelompok.pdf dengan ketentuan isi pada isi laporan
3. README untuk tata cara penggunaan yang minimal berisi:
    - ❖ Penjelasan singkat algoritma greedy yang diimplementasikan untuk setiap bot yang dibuat
    - ❖ Requirement program dan instalasi tertentu bila ada
    - ❖ Command atau langkah-langkah dalam meng-compile atau build program
    - ❖ Author (identitas pembuat)

- Sangat disarankan untuk menggunakan **semantic commit.** Buatlah release dengan format v1.x dengan x adalah nomor versi dimulai dari revisi ke 0. Contoh : v1.0 untuk release pertama, v1.1 release kedua dan selanjutnya.
- Pastikan untuk membuat repository bersifat Public paling lambat H+1 deadline (1 hari setelah deadline). Sebelum deadline repository harus bersifat Private.
- Laporan dan tautan rilis repository dikumpulkan Jumat, 13 Maret pada alamat Google Form **berikut** paling lambat pukul 23.59 WIB:

PERINGATAN: Keterlambatan akan mengurangi nilai sebanyak 1 poin untuk setiap menit keterlambatan.

- Adapun pertanyaan terkait tugas besar ini bisa disampaikan melalui QnA **berikut**.

Ketentuan Laporan

- **Cover:** Cover laporan ada foto anggota kelompok (foto bertiga). Foto ini menggantikan logo "gajah" ganesha.
- **Bab 1:** Deskripsi tugas (dapat menyalin spesifikasi tugas ini)
- **Bab 2:** Landasan Teori.
    - **Dasar teori** (algoritma **greedy**) secara umum

- Bagaimana cara kerja program secara umum (bagaimana bot melakukan aksinya, bagaimana mengimplementasikan algoritma **greedy** ke dalam bot, bagaimana menjalankan bot, dll).
- **Bab 3:** Aplikasi strategi **greedy**
- Proses mapping persoalan *Battlecode 2025* menjadi elemen-elemen algoritma **greedy** *(himpunan kandidat, himpunan solusi, fungsi solusi, fungsi seleksi, fungsi kelayakan, fungsi objektif)*
- Eksplorasi 3 alternatif solusi greedy yang mungkin dipilih dalam persoalan battlecode. (setiap alternatif harus memiliki heuristic yang berbeda)
- Analisis efisiensi dan efektivitas dari kumpulan alternatif solusi **greedy** yang dirumuskan
- Strategi **greedy** yang dipilih dari 3 alternatif solusi **greedy** (yang akan diimplementasikan dalam program) beserta alasan dan pertimbangan pemilihan strategi tersebut.
- **Bab 4:** Implementasi dan pengujian.
- Implementasi 3 alternatif solusi (pseudocode yang cukup detail sehingga dapat dimengerti dengan mudah. Berikan komen pada pseudocode jika merasa perlu)
- Penjelasan struktur data, fungsi, dan prosedur yang digunakan pada bot dengan solusi greedy yang dipilih.
- Pengujian **greedy** minimal sebanyak 3 kali berupa pertandingan antara bot utama (yang dipilih) dengan bot alternatif (format 1 vs 1). Pengujian yang dicantumkan sebaiknya mencakup semua kejadian yang unik.
- Analisis hasil dari pengujian. Misalnya adalah apakah strategi greedy berhasil mendapatkan nilai optimal, lalu jika tidak, dalam kondisi seperti apa strategi **greedy** tidak berhasil mendapatkan nilai optimal, dsb.
- **Bab 5:** Kesimpulan dan saran.
- **Lampiran:** Tautan repository GitHub dan video (jika membuat)
- **Daftar Pustaka**

Keterangan laporan:

1. Laporan ditulis dalam bahasa Indonesia yang baik dan benar.
2. Laporan mengikuti format pada section “Isi laporan” dengan baik dan benar.
3. Identitas per halaman harus jelas (misalnya : halaman, kode kuliah).
4. Pastikan memasukkan pernyataan tersebut pada laporan, yaitu **Pernyataan tidak melakukan kecurangan yang di tandatangani** dengan format sebagai berikut:

Tugas ini disusun sepenuhnya tanpa bantuan kecerdasan buatan (*Generative AI*), melainkan hasil pemikiran dan analisis mandiri.

[Tanda tangan mahasiswa]
[Nama mahasiswa]

[Tanda tangan mahasiswa]
[Nama mahasiswa]

[Tanda tangan mahasiswa]
[Nama mahasiswa]

Penilaian

1. **Bagian 1: Desain solusi algoritma greedy ditulis dalam laporan (45%)**

a. Pemahaman tugas besar (5%)

b. Mapping persoalan battlecode ke elemen-elemen algoritma greedy (10%)

c. Analisis efisiensi dan efektivitas secara teoritis dari alternatif solusi persoalan (5%)

d. Analisis dari Desain solusi Algoritma Greedy yang diusulkan untuk diimplementasikan dalam program/coding (25%)

2. **Bagian 2: Implementasi program dan demo (55%)**

a. Kesesuaian strategi Greedy yang dituliskan dengan implementasi dan saat demo (20%)

b. Kualitas algoritma Greedy yang diimplementasikan (10%)

c. Modularitas/keterbacaan penulisan program (5%)

d. Demo pemahaman program (20%)

3. **Bagian 3: Kompetisi dan komponen lainnya**

a. Beberapa kelompok pemenang (akan ditentukan jumlahnya nanti) pada saat kompetisi akan mendapatkan bonus nilai (bonus maksimal 10 poin)

b. Bonus dalam membuat video kelompok (bonus maksimal 6 poin)

c. Pembuatan ***Post Mortem Analysis*** (bonus maksimal 7 poin)

Perhatian

- Dilarang keras copy paste program dari internet, AI, repository lain, ataupun program milik teman. Program yang dikerjakan menggunakan agent (Opus, Sonnet, dan lain-lain) akan sangat terlihat dan akan memiliki kemiripan dengan mahasiswa yang menggunakan hal serupa. Segala bentuk kecurangan akan diberikan sanksi berat yaitu nilai tugas menjadi nol. Kami tidak ingin hal itu terjadi dan kami ingin Anda untuk dapat memikirkan solusi algoritma dari hasil pemikiran Anda sendiri.

- Pastikan program **dapat dikompilasi setidaknya** pada windows dan linux.
- Apabila program **tidak dapat dijalankan** maka tidak akan dinilai oleh asisten.

Referensi

- **Battlecode 2025**

https://battlecode.org/

- **Spesifikasi mendetail dari API-API yang disediakan**

https://releases.battlecode.org/javadoc/battlecode25/3.1.0/battlecode/common/package-summary.html

--- 2 Tipe mahasiswa ketika tubes bikin bot ---

--- Fariz---

“baru 1 tucil ama 1 tubes, perjalanan masi panjang”
--- Hakim---

“Melamban bukanlah hal yang tabu”
--- Radhi---

“😐”

--- Lukas---

“Dor dor semangat membantai semester 4 nya!”
--- Carlo---

“Ketika akhirnya menduduki takhta Firaun, Mulhotep segera melakukan *rebranding*. Ia merasa nama lamanya kurang menjual untuk visi global. Ia mengganti namanya menjadi **Firaun Ngutangkhamun Akehtenan III (Periode)** ”
--- Orvin---

“Good luck jangan sampai bikin stres parah”
--- Rizal---

“Stand proud, you are strong”
--- Theo---