

BAB III ANALISIS DAN PEMODELAN

3.1 Analisis

Pengamatan dilakukan terhadap *framework game grid-tile sandbox* yaitu Cubiquity¹, Uniblocks Voxel Terrain², dan ironVoxel³. Berdasarkan pengamatan didapat beberapa hal atau fitur sejenis yang terdapat pada ketiga *framework* tersebut. Berikut adalah tabel perbandingan antara ketiga *framework* dan *framework* yang akan dibuat.

Tabel 3.1 Perbandingan Fitur *Framework*

Fitur	Cubiquity	Uniblocks Voxel Terrain	ironVoxel	<i>Framework</i> yang akan dibuat
<i>Tile</i> dapat berbentuk kubik atau halus	Ada	Ada	Tidak Ada	Tidak Ada
Terdapat block editor	Tidak Ada	Ada	Tidak Ada	Tidak Ada
Pemrosesan dunia <i>game</i> dalam bentuk potongan-potongan	Ada	v	Ada	Ada
Dapat mengubah dunia <i>game</i> saat permainan	Ada	Ada	Ada	Ada
Dapat membuat <i>game multiplayer</i>	Tidak Ada	Ada	Tidak Ada	Tidak Ada
Terdapat generator dunia	Tidak Ada	Ada	Ada	Tidak Ada
Pengecekan kolisi objek	Ada	Ada	Ada	Ada
Terdapat manajemen benda dasar	Tidak Ada	Tidak Ada	Tidak Ada	Ada
<i>Game</i> hasil implementasi berupa dunia 3D atau 2D	3D	3D	3D	2D

Berdasarkan hasil pengamatan, hal penting dalam sebuah *game grid-tile sandbox* adalah cara

¹<https://www.assetstore.unity3d.com/jp/#!/content/12689>

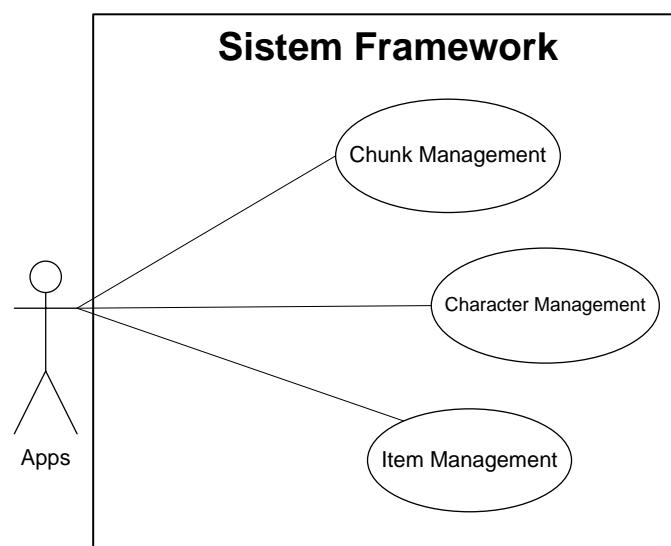
²<https://www.assetstore.unity3d.com/jp/#!/content/14768>

³<http://forum.unity3d.com/threads/ironvoxel-infinite-world-voxel-solution.266947/>

pemrosesan dari dunia *game*. Contoh pemrosesan dari dunia *game* adalah pemrosesan dari sebuah dunia dengan jumlah *tile* sangat banyak. Cara yang dapat digunakan adalah cara yang telah dijelaskan pada subbab 2.4 yaitu dengan memotong peta menjadi kelompok-kelompok yang lebih kecil. Program tidak dituntut untuk memroses seluruh kelompok tetapi hanya dilakukan pada sebagian kecil kelompok saja dengan adanya pembagian ini. Kelompok-kelompok mana saja yang akan diproses ditentukan oleh titik fokus dari layar permainan.

3.2 Arsitektur Framework

Arsitektur *Framework* dalam *framework* menggunakan *Unified Modelling Language* (UML). Diagram yang digunakan untuk pemodelan adalah *use case diagram*, *Sequence diagram*, dan *class diagram*. Pemodelan dilakukan berdasarkan hasil dari subbab 3.1 analisis.



Gambar 3.1 Use Case Diagram Framework

3.2.1 Use Case Diagram Framework

Pada Gambar 3.1 dapat dilihat terdapat fungsi dasar yaitu *Chunk Management* yaitu fungsi

untuk mengatur potongan-potongan dari peta dunia, *Tile Management* yaitu fungsi untuk menambahkan atau mengurangi *tile* pada peta dunia, *Character Management* yaitu fungsi untuk mengatur dan memroses karakter, dan *Item Management* yaitu fungsi untuk mengatur benda dan tas karakter. Pada setiap bagian akan disertakan *activity diagram* dan *sequence diagram*.

3.2.1.1 Chunk Management

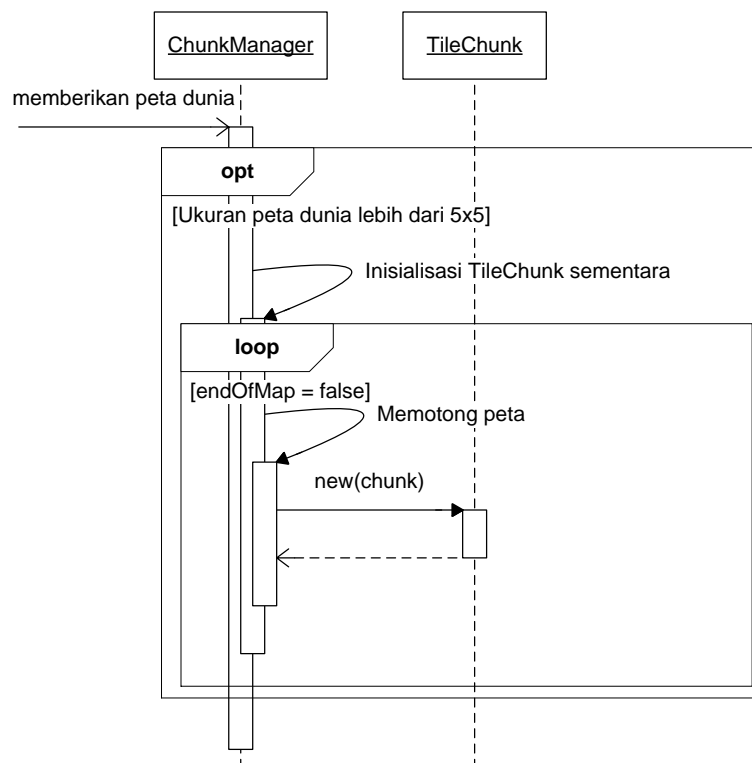
Seperti yang telah dijelaskan di subbab 3.1, pemrosesan peta dunia dilakukan dengan membuat peta dunia yang besar menjadi potongan-potongan yang lebih kecil. Potongan dari peta dunia yang akan diproses untuk diperbarui dipilih dari titik fokus permainan. Proses pembaruan (*update*) dilakukan terhadap potongan peta yang sudah dipilih sehingga proses pembaruan tidak perlu dilakukan terhadap seluruh peta dunia. Proses penambahan dan pengurangan *tile* juga terdapat pada fungsi ini. *Chunk Management* memiliki fungsi-fungsi yaitu pemotongan peta dunia, pemrosesan potongan peta, penambahan *tile*, dan perusakan *tile*.

1. Pemotongan Peta Dunia

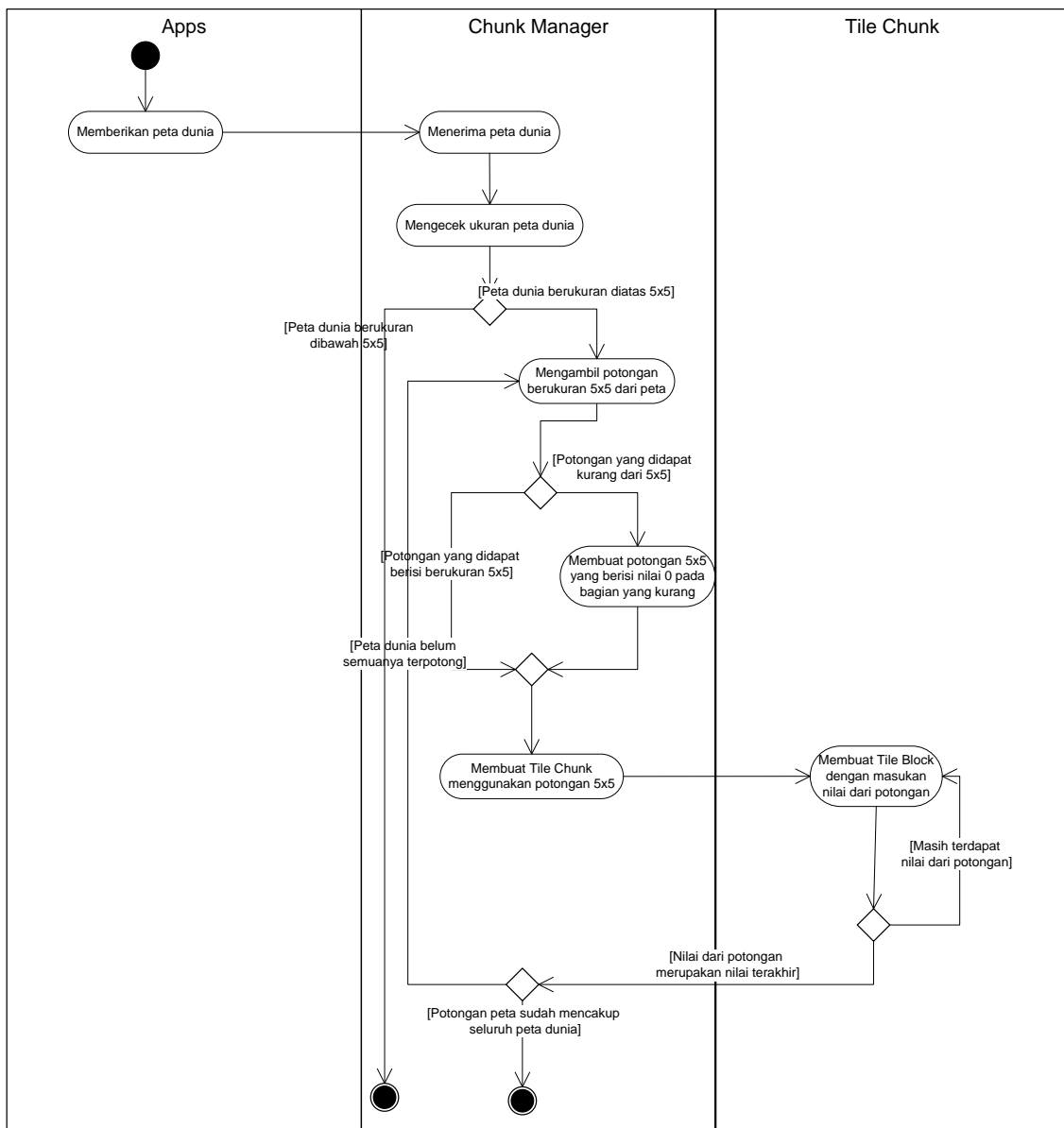
Proses pemotongan peta dunia dilakukan pada saat awal *framework* dijalankan. Pemotongan dilakukan jika peta dunia yang dimasukkan pengguna memiliki besar di atas 5×5 *tile*. Pemotongan peta dunia dilakukan mengikuti aturan besar setiap potongana dalah 5×5 . Jika besar peta dunia masukan dari pengguna tidak berkelipatan 5 maka potongan peta yang memiliki nilai kurang akan diisi dengan *tile* kosong. *Sequence diagram* dan *activity diagram* dari proses pemotongan peta dunia dapat dilihat pada gambar 3.2 dan gambar 3.3.

Pada gambar 3.2 dan 3.3 proses dimulai dengan aplikasi implementasi memberikan peta dunia berbentuk *tile* kepala Chunk Manager untuk diproses menjadi bentuk

potongan-potongan. Setelah menerima peta dunia secara utuh, Chunk Manager akan mengecek apakah ukuran peta dunia lebih kecil atau lebih besar dari besar potongan yang digunakan. Jika ukuran peta dunia diatas ukuran potongan, Chunk Manager akan melakukan pemotongan secara berurutan. Saat mengambil potongan, Chunk Manager akan mengecek apakah hasil dari pemotongan akan didapat potongan 5x5 secara utuh atau tidak. Jika potongan yang didapat tidak utuh, maka penambahan data *tile* kosong akan dilakukan agar potongan menjadi berukuran 5x5. Chunk Manager akan membuat potongan dalam bentuk Tile Chunk menggunakan hasil potongan,. Selanjutnya Tile Chunk akan membuat *tile* sesuai dengan potongan. Proses ini berlangsung sampai seluruh peta terpotong.



Gambar 3.2 Sequence Diagram Pemotongan Peta Dunia



Gambar 3.3 Activity Diagram Pemotongan Peta Dunia

2. Pemrosesan Potongan Peta

Pemrosesan potongan peta atau kelompok *tile* digunakan untuk menentukan kelompok *tile* mana saja yang menjadi fokus untuk di-*update* oleh sistem. Proses ini diawali dengan sistem mengecek apakah nilai baris dan kolom kelompok *tile* yang sedang menjadi fokus sama dengan nilai baris dan kolom yang tersimpan sekarang. Jika terdapat perbedaan pada nilai

baris atau kolom maka kelompok *tile* yang bersebelahan dengan nilai baris atau kolom baru akan ditambahkan. Kelompok *tile* yang tidak bersebelahan dengan nilai baris atau kolom baru akan dibuang sehingga jumlah kelompok menjadi tetap. Penambahan kelompok *tile* baru dilanjutkan dengan penyesuaian posisi kelompok *tile* baru tersebut dengan posisi kelompok *tile* fokus. *Sequence diagram* dan *activity diagram* untuk pemrosesan potongan peta dapat dilihat pada gambar 3.4 dan gambar 3.5.

3. Penambahan *Tile*

Sistem mengecek apakah terdapat *tile* atau tidak pada koordinat yang dipilih. Setelah itu pengecekan apakah nilai masukan bertipe *tile*. *Tile* akan dimasukkan ke dalam kumpulan *tile* yang sesuai dengan tipe apakah *tile* tersebut *tile* dinding atau pijakan. *Sequence diagram* dan *activity diagram* penambahan *tile* diperlihatkan pada Gambar 3.6 dan 3.7

4. Perusakkan *Tile*

Untuk menghancurkan *tile* diperlukan untuk mengurangi kekerasan *tile* terlebih dahulu. Sistem mengecek apakah *tile* yang ditunjuk kosong atau tidak. *Tile* yang terpilih akan dikurangi kekerasannya sejumlah kerusakan yang diberikan. Setelah itu perubahan tekstur *tile* saat kekerasan *tile* 33% dan 66%. *Sequence diagram* dan *activity diagram* perusakkan *tile* diperlihatkan pada Gambar 3.8 dan 3.9

3.2.1.2 Character Management

Character management yang terdapat dalam *framework* adalah fungsi pengecekan kolisi karakter dan fungsi untuk memunculkan karakter baru. Proses pengecekan kolisi karakter terdapat digunakan untuk mengecek pergerakan karakter apakah karakter akan bertabrakan dengan

sesuatu atau tidak. Pada fungsi untuk memunculkan karakter baru, fungsi pengecekan kolisi karakter digunakan untuk mengecek apakah karakter bertabrakan dengan peta atau tidak. Pemunculan karakter dilakukan jika karakter tidak bertabrakan dengan peta.

1. Pengecekan Kolisi Karakter

Fungsi untuk mengecek kolisi karakter dengan peta digunakan untuk mengecek apakah karakter tersebut akan menabrak tile atau karakter memiliki pijakan atau tidak. Pada karakter terdapat 6 kotak kolisi yaitu kanan atas, atas, kiri atas, kanan bawah, bawah, dan kiri bawah. Pemisahan menjadi 6 kotak kolisi dilakukan karena akan terjadi perlakuan yang berbeda untuk setiap kolisi. Contohnya bila kotak kolisi atas kanan atau kiri karakter berkenaan dengan peta, karakter tidak dapat lanjut berjalan. Bila kotak kolisi bawah kanan atau kiri karakter berkenaan dengan peta tetapi seluruh kotak kolisi atas karakter kosong, maka karakter akan berpindah naik ke tile depannya. Pengecekan kotak kolisi dilakukan berdasar arah pergerakan karakter. Bila karakter bergerak secara horizontal, maka kotak kolisi yang digunakan hanya kanan atau kiri atas dan kanan atau kiri bawah. *Sequence diagram* dan *activity diagram* untuk pengecekan kolisi karakter dapat dilihat pada gambar 3.10 dan gambar 3.11.

2. Memunculkan Karakter Baru

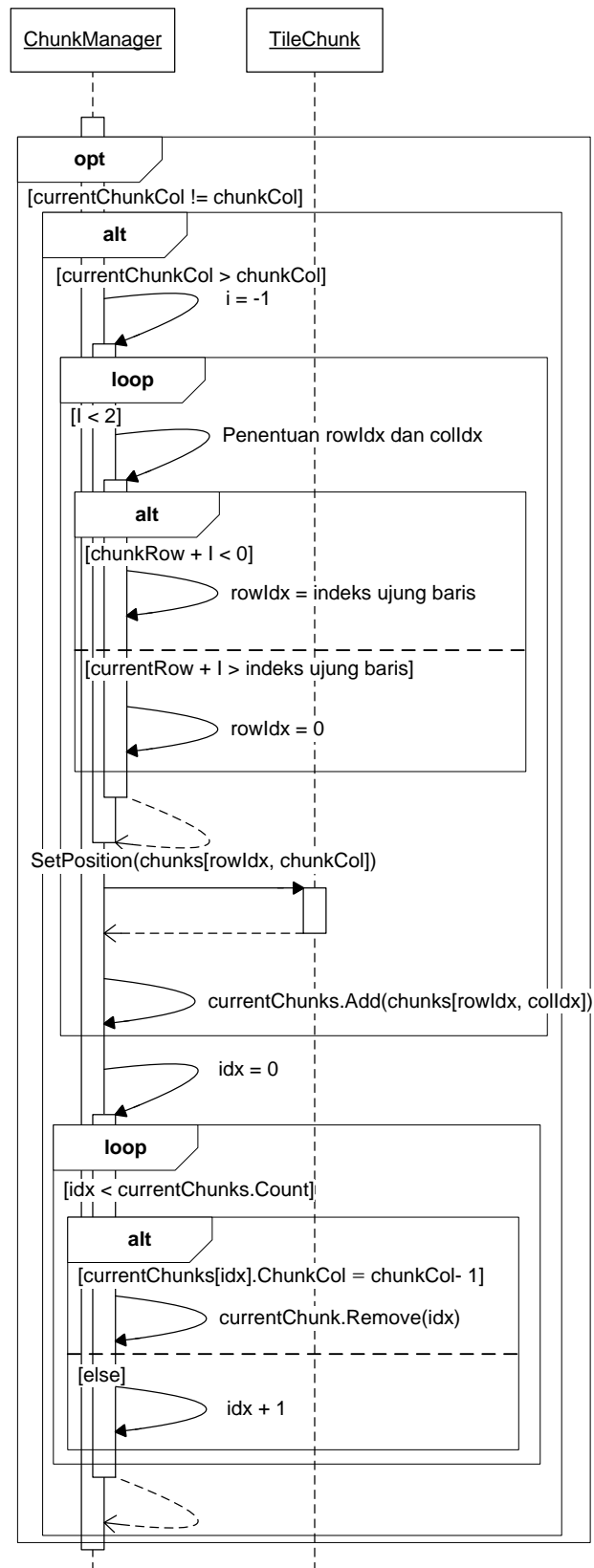
Fungsi untuk memunculkan karakter digunakan untuk memunculkan karakter pada posisi yang ditentukan. Masukan yang diterima untuk memunculkan karakter adalah karakter dalam bentuk kelas *Character*. Sistem akan mengecek apakah karakter baru tersebut akan terhalang atau tidak. *Sequence diagram* dan *activity diagram* untuk *Place Tile* diperlihatkan pada Gambar 3.12 dan 3.13.

3.2.1.3 Item Management

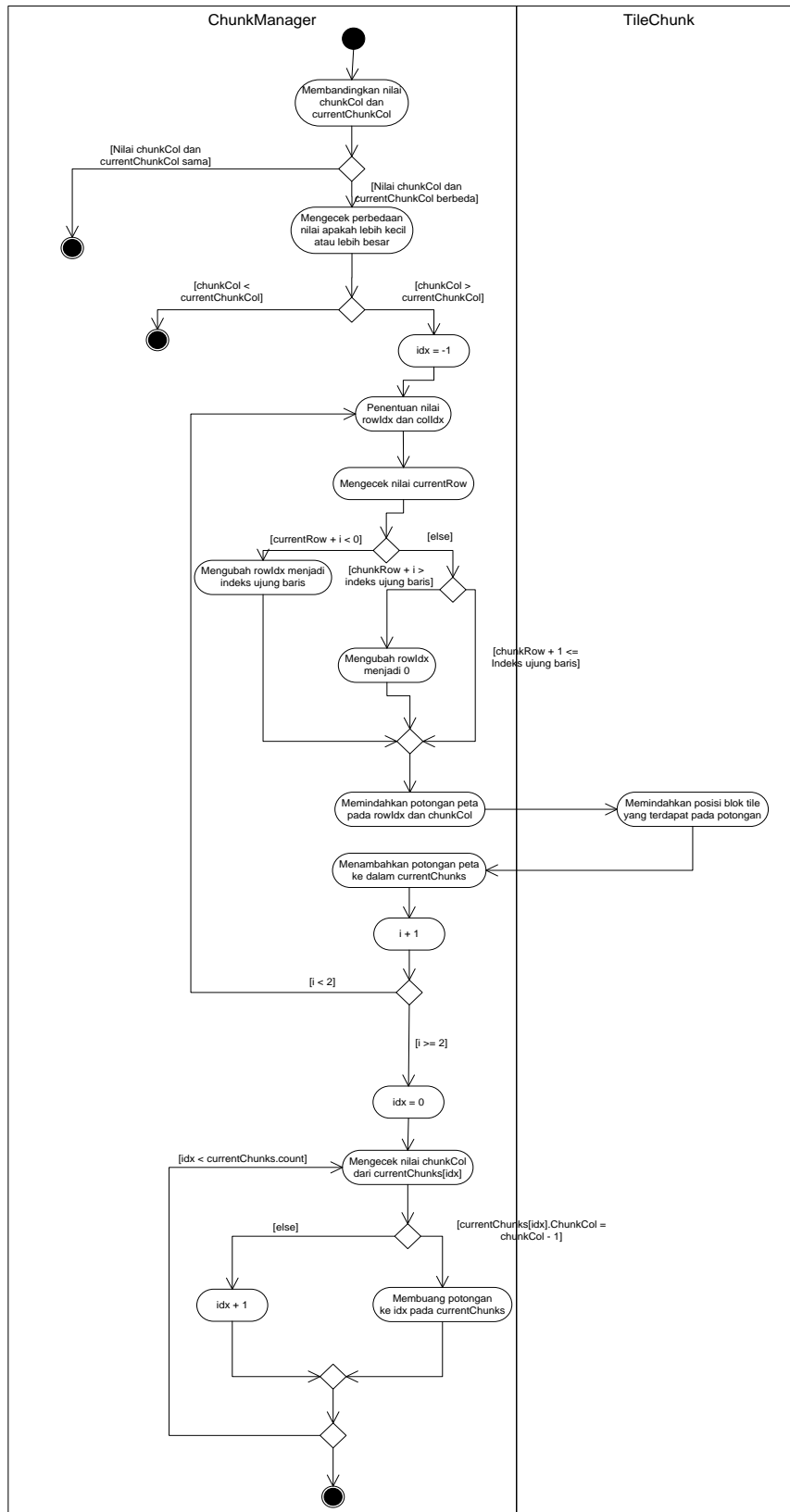
Item management yang terdapat pada *framework* adalah fungsi untuk memunculkan benda pada peta. Untuk memunculkan benda dibutuhkan nilai indeks dan tipe dari benda. Pengecekan terhadap nilai indeks dan tipe benda dilakukan untuk memastikan benda terdapat dalam data. *Sequence diagram* dan *activity diagram* untuk memunculkan benda dapat dilihat pada gambar 3.14.

3.2.2 Class Diagram

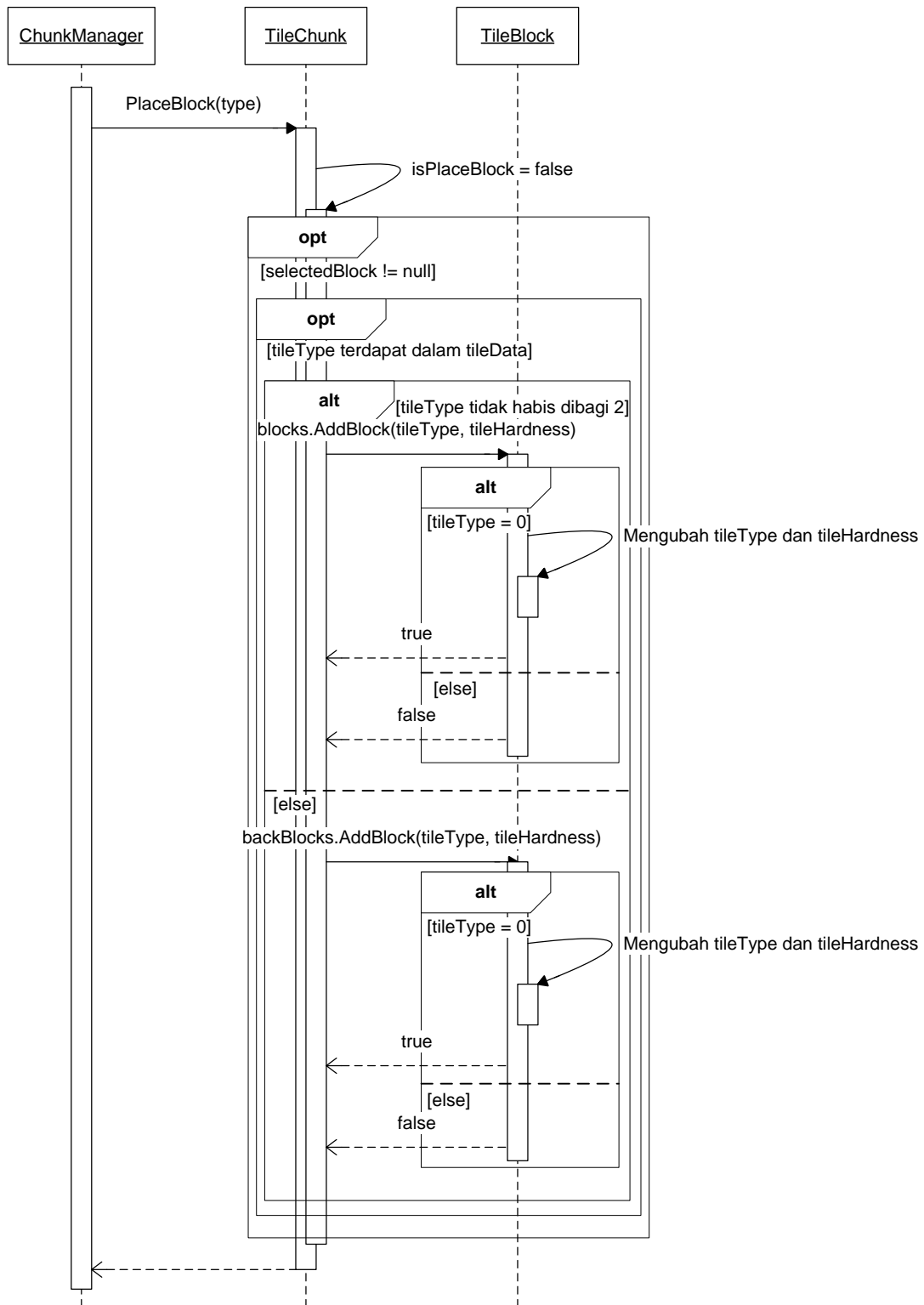
Berdasarkan *sequence diagram* dan *activity diagram* yang telah dibahas sebelumnya, *class diagram* dari *framework* dapat dimodelkan. Selain dengan *class* yang merepresentasikan entitas dari *sequence diagram* dan *activity diagram*, terdapat penambahan *class* untuk tipe dari data penyimpanan dan *class* utama yang menyatukan *class-class* lainnya. Relasi *class diagram* dapat dilihat pada gambar 3.15. Penjelasan mengenai atribut dan *method* pada *class* dapat dilihat pada lampiran A.



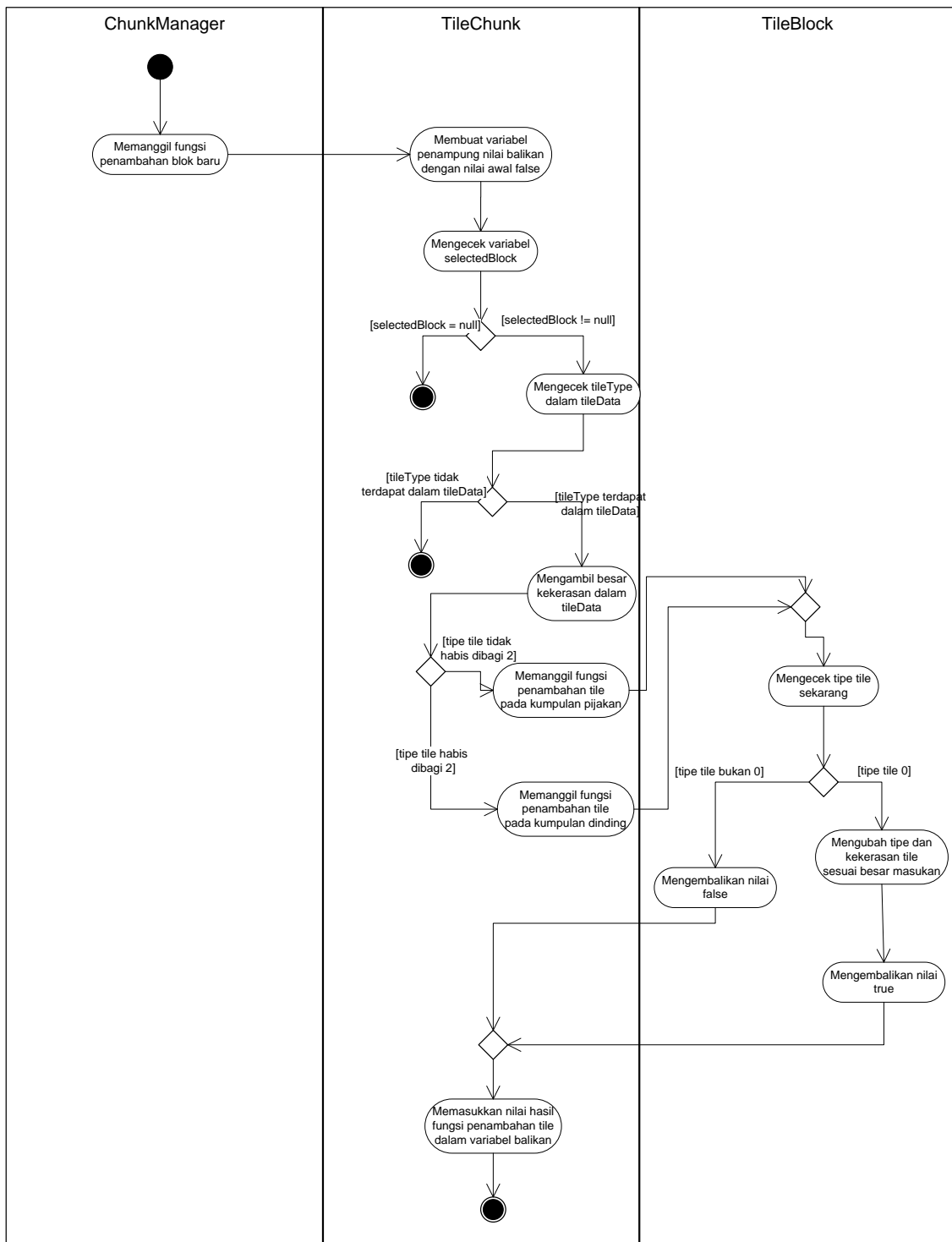
Gambar 3.4 Sequence Diagram Pemrosesan Potongan Peta



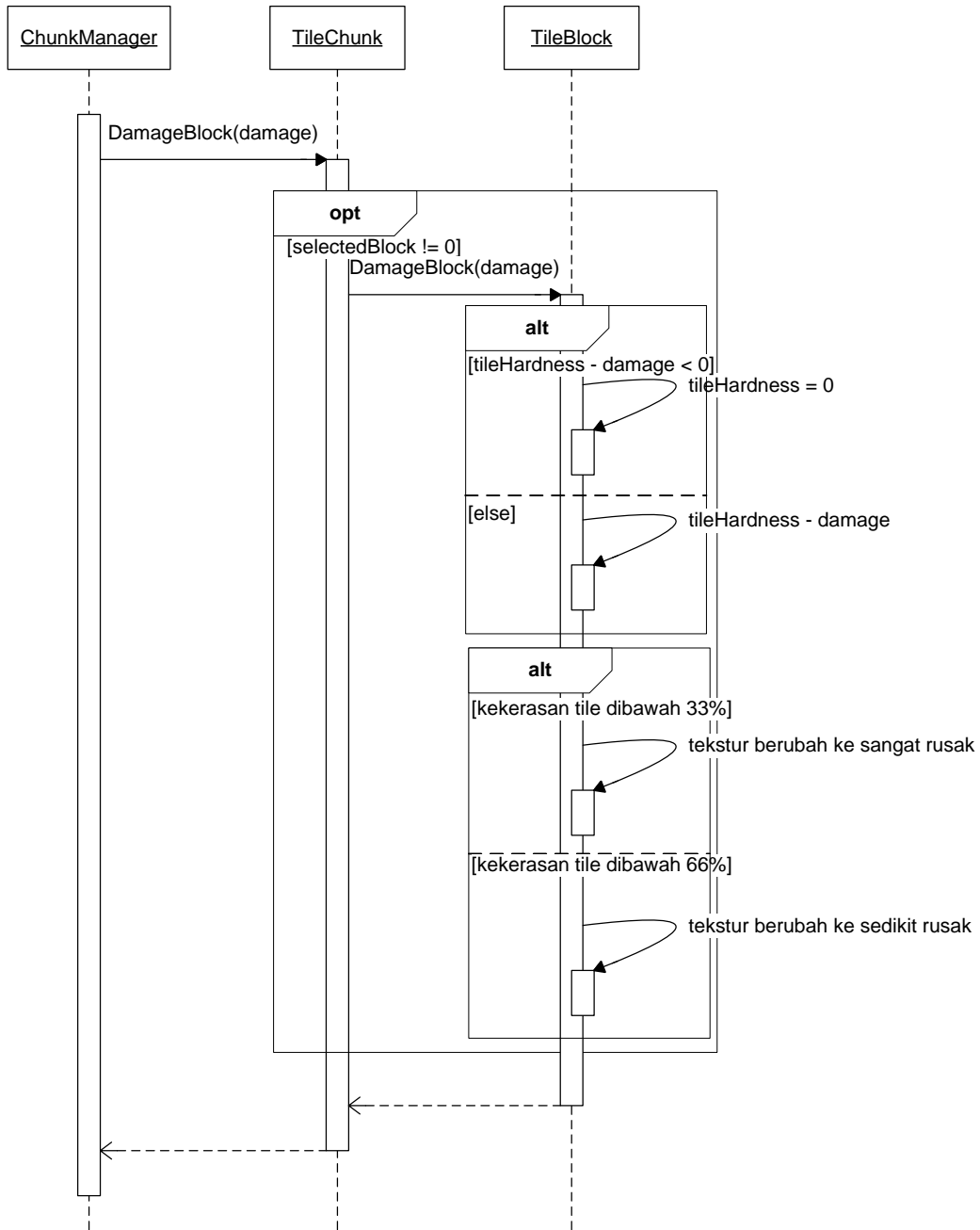
Gambar 3.5 Activity Diagram Pemrosesan Potongan Peta



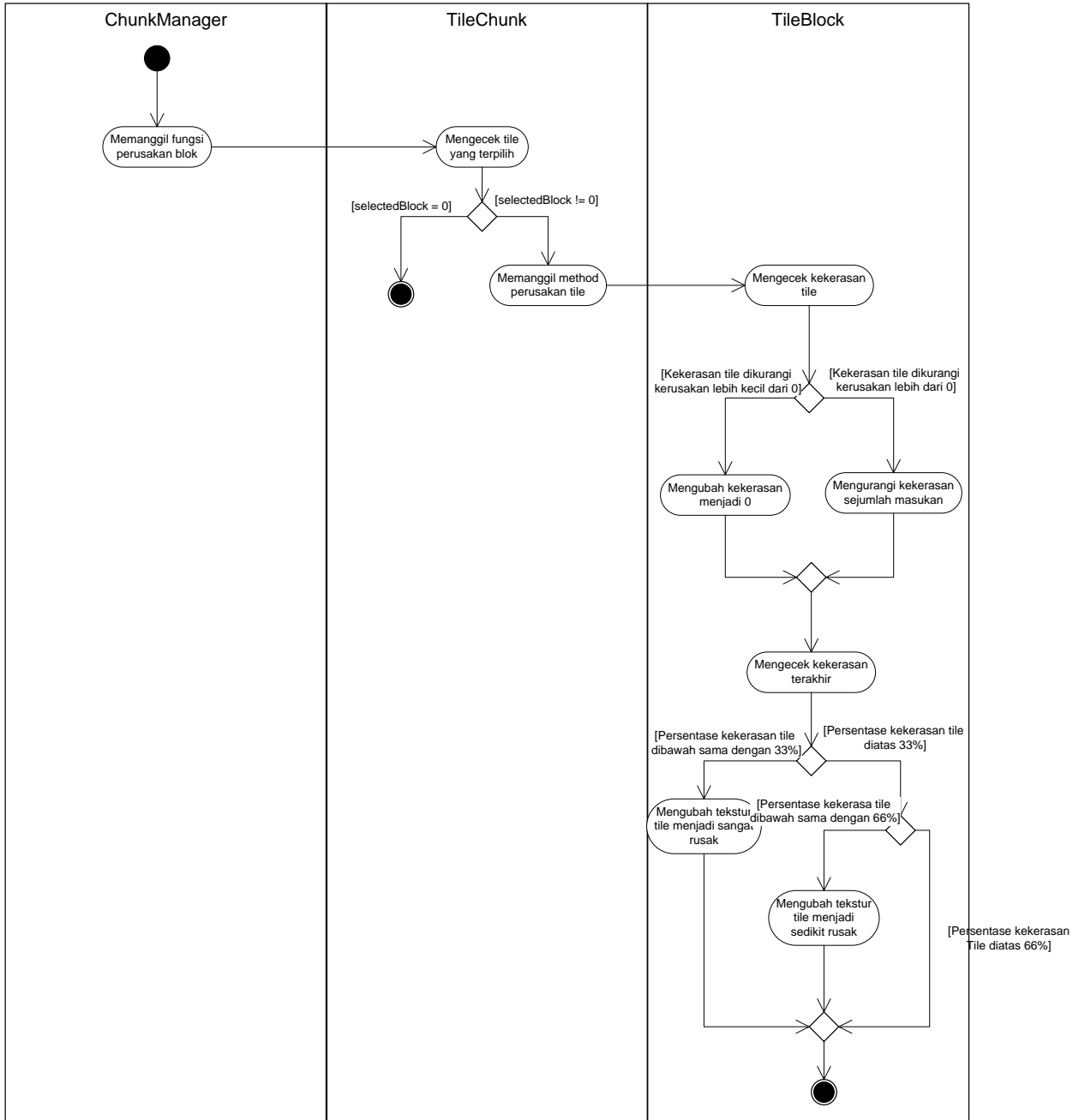
Gambar 3.6 Sequence Diagram Penaruhan Tile



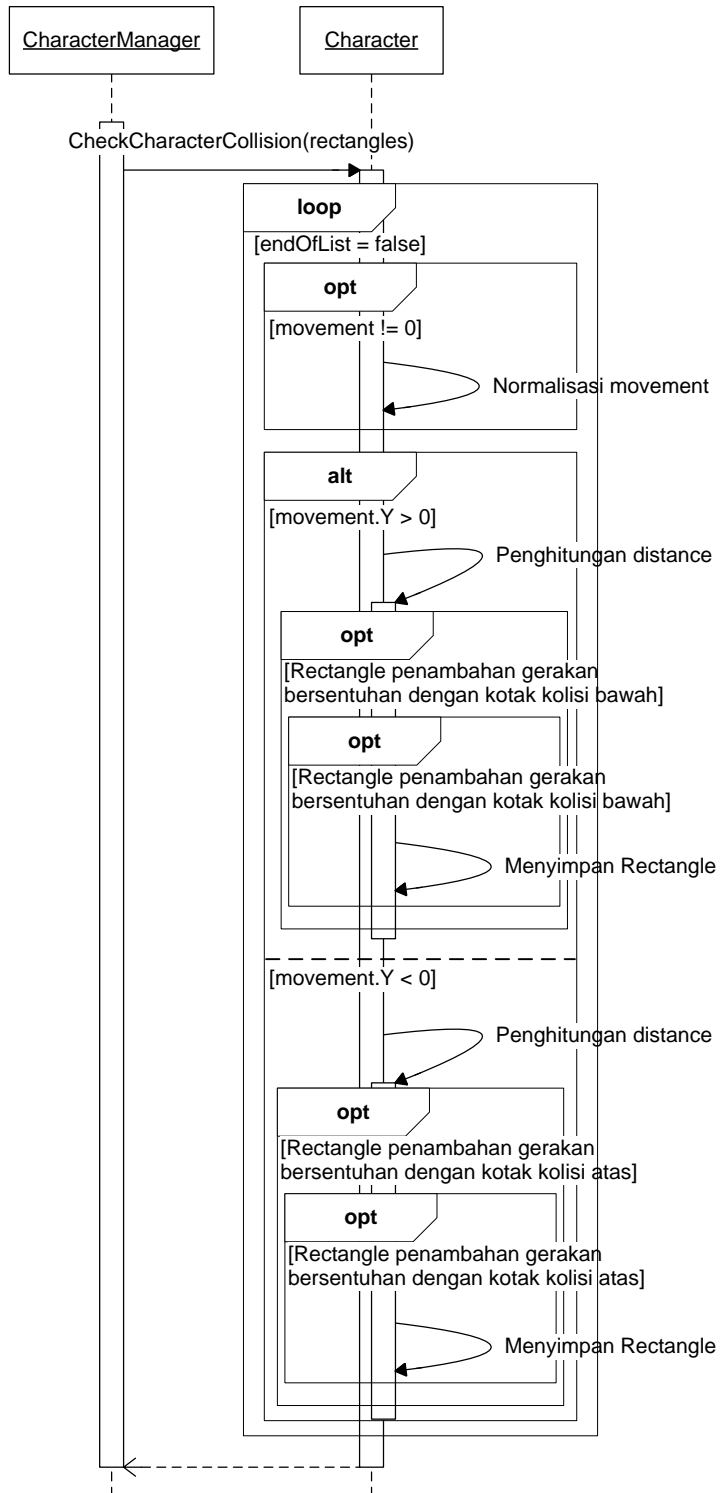
Gambar 3.7 Activity Diagram Penaruhan Tile



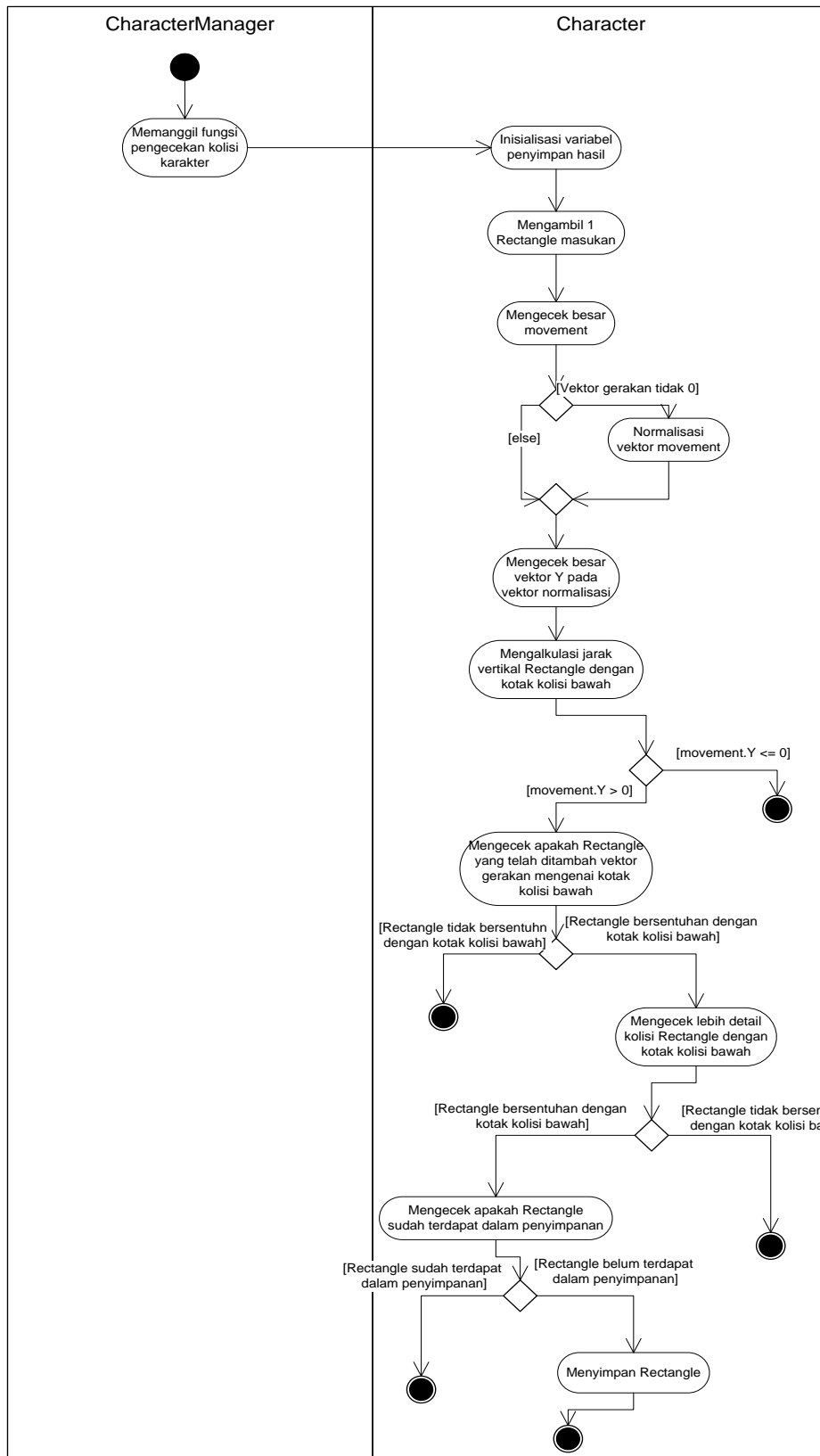
Gambar 3.8 Sequence Diagram Perusakan Tile



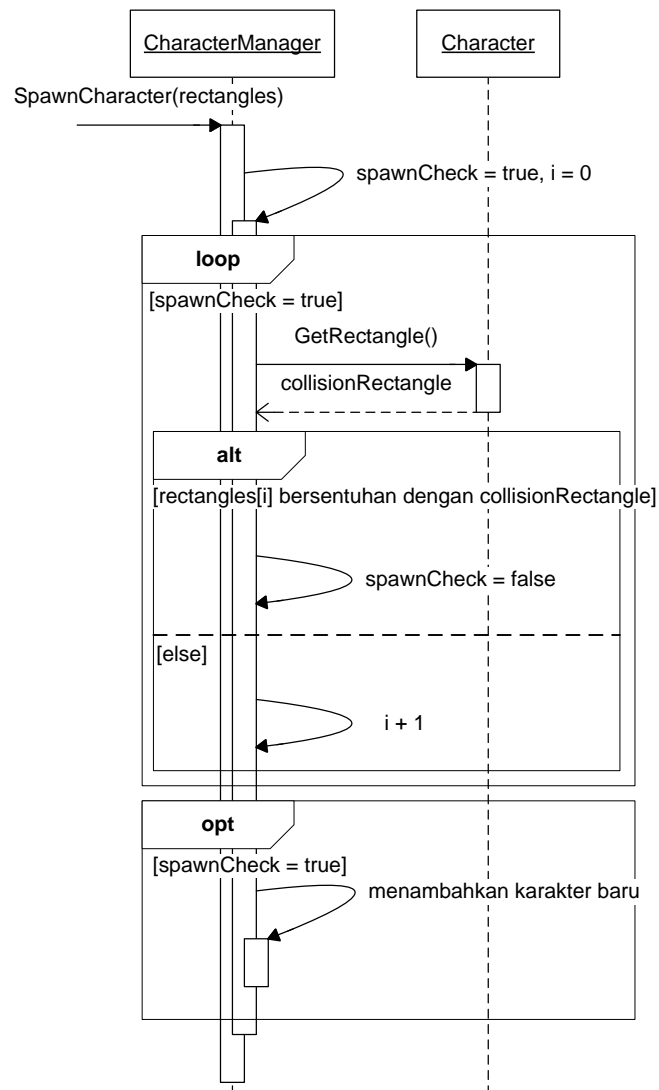
Gambar 3.9 Activity Diagram Perusakan Tile



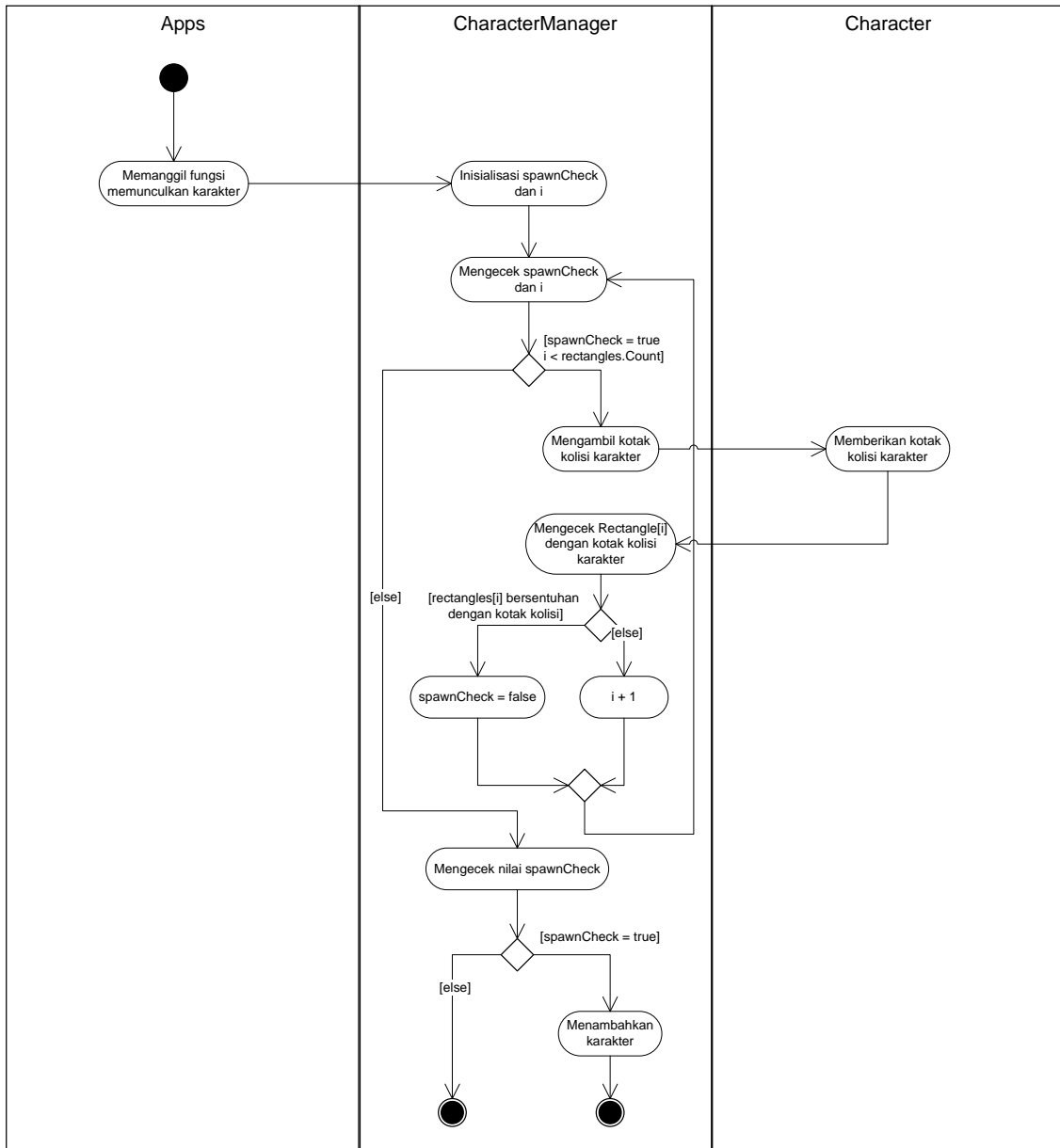
Gambar 3.10 Sequence diagram Pengecekan Kolisi Karakter



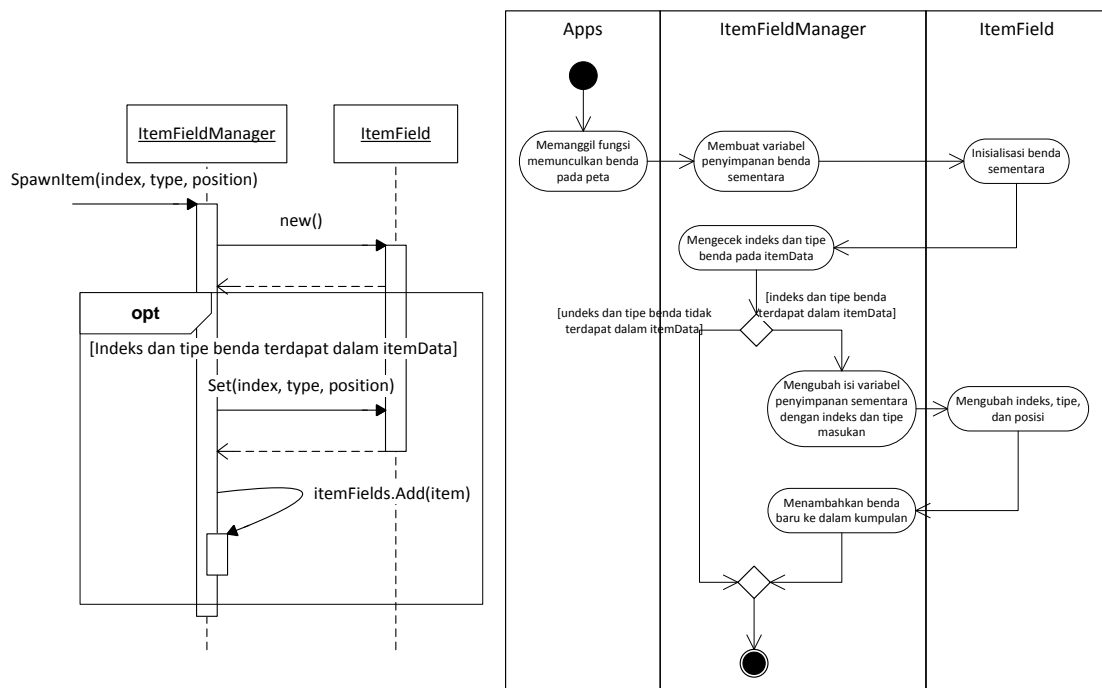
Gambar 3.11 Activity Diagram Pengecekan Kolisi Karakter



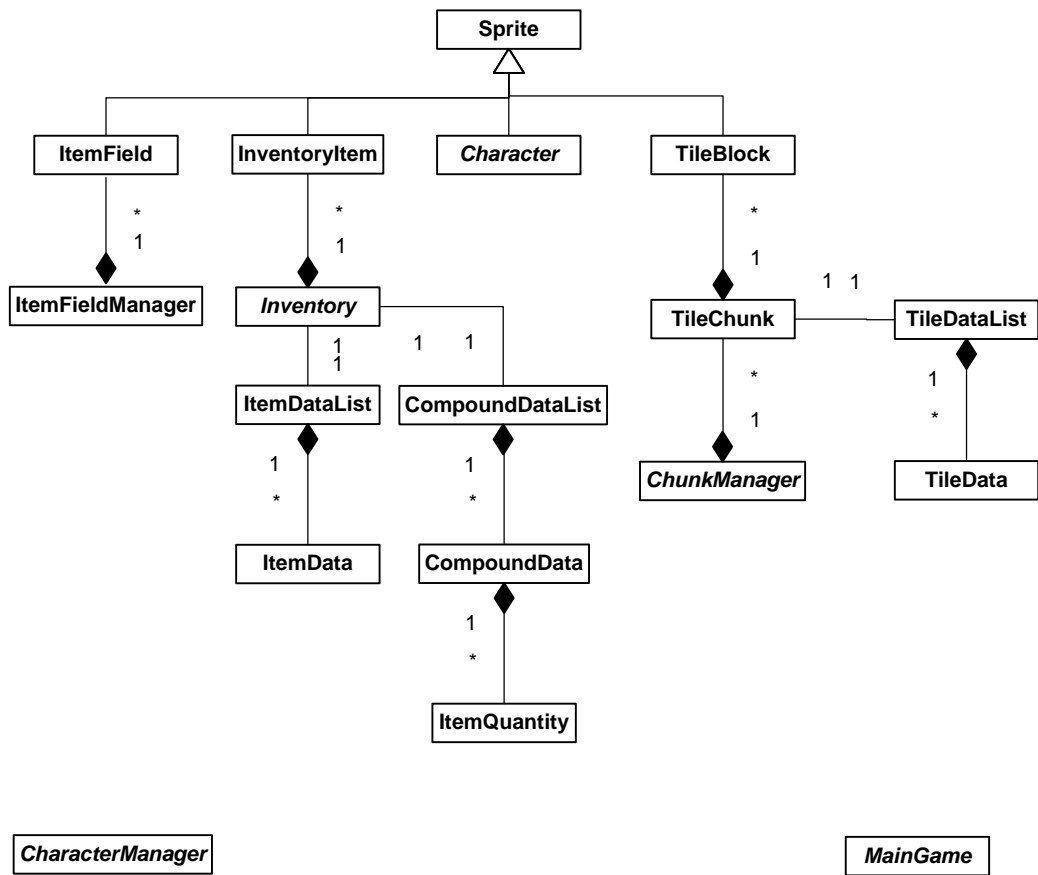
Gambar 3.12 Sequence Diagram Memunculkan Karakter



Gambar 3.13 Activity Diagram Memunculkan Karakter



Gambar 3.14 Diagram Memunculkan Benda(kiri sequence diagram kanan activity diagram)



Gambar 3.15 Relasi Class Diagram Framework