



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB II

LANDASAN TEORI

2.1 Game Design

Game memiliki banyak sekali *genre* dengan tujuan yang bermacam-macam. Charlier dan De Fraine (2012) mengatakan penggunaan game untuk edukasi sudah berkembang selama bertahun-tahun dan Razak, Connolly dan Hainey (2012) mengatakan bahwa sebuah game bisa menjadi suatu alat yang efektif untuk belajar banyak mata pelajaran seperti matematika, bahasa dan sains. Eric Zhi Feng Liu dan Po-Kuang Chen (2013) mengatakan bahwa motivasi untuk belajar dan efisiensinya bisa ditingkatkan melalui game dan element terbaru dari game membuat game menjadi sangat populer. Dan cheng (2009) mengatakan bahwa proses pembuatan game itu tersendiri bisa jadi media pembelajaran yang efektif untuk melatih problem solving, reasoning, dan creative thinking. Permasalahanya Penelope Sweetser (2013) mengatakan bahwa game development masih baru di perguruan tinggi sehingga metode masih sangat sedikit. Eric Zhi Feng Liu and Po-Kuang Chen (2013) juga mengatakan bahwa game bukan hanya bisa untuk belajar, tetapi juga membantu untuk bisa memahami abstrak konsep yang lebih dalam.

Glenda A Gunter, Robert F. Kenny dan Erik Henry Vick (2005). mengatakan bahwa game harus di desain sedemikian rupa untuk mencapai tujuan. Penelitian tersebut juga memberikan contoh bahwa game learning saja apabila di desain secara kurang baik bisa membuat tujuan utama dari game tersebut yaitu

untuk proses learning menjadi hilang. Sehingga penelitian tersebut menyarankan design paradigm agar tujuan utama dari sebuah game tidak akan hilang.

2.1.1 Game Design Document

Game design document merupakan salah satu cara agar game yang sedang di develop tidak akan melenceng dari desain awalnya. *Game design document* itu sifatnya sangat dinamis tergantung dari game itu sendiri. Menurut Adam Ardisasmita (2015) semua game design document biasanya dibagi menjadi beberapa bab yaitu *game overview*, *core gameplay*, *game mechanic and interaction*, *wireframe* dan *game module breakdown*.

Game Overview isinya menjelaskan informasi dasar dari game seperti tentang apa, sebagian besar ceritanya apa, genrenya dan sebagainya. Biasanya ditunjukkan dengan *menu flow*.

Core Gameplay berisikan tentang rules-rules dan bagaimana permainan akan di desain. Hilangnya tujuan utama dari sebuah permainan bisa terpengaruh apabila bagian ini tidak di develop dengan baik.

Game Mechanic and Interaction berisi tentang semua peraturan dan komponen mekanik disini. Hal ini adalah yang terpenting dari *game design* karena bagian ini adalah kunci untuk menentukan apakah permainan sesuai dengan tujuan yang ingin kita capai.

Wireframe berisi tentang prototype game berupa user interface dan dasar dari gamenya itu tersendiri.

Game Module Breakdown itu seperti *scheduling*, target apa setiap hari yang harus dicapai dan apa yang harus dibuat selanjutnya.

2.2 Level Designing

Rouse (2004) mengatakan level di game adalah sebuah area distinct yang terpisah baik sisi geografi, naratif, dan gameplaynya. Penelope Sweetser(2013) mengatakan bahwa separasi dan urutan dari level memiliki dampak yang cukup substantial terhadap flow dari game karena orang biasanya bermain satu level per sesi dan mendapatkan pencapaian setelah menyelesaikan satu level. Dan Taylor (2015) mengatakan bahwa komunikasi yang baik seperti lingkaran yang rusak. Pembuat dari game membuat lingkarannya tetapi meninggalkan sedikit bagian yang kosong dimana bagian tersebut akan diisi oleh player sehingga terjadi komunikasi yang baik. Akan tetapi perlu diperhatikan kekosongannya, apabila terlalu besar player akan bingung, jika terlalu kecil orang tidak akan sadar. Beberapa cara untuk membuatnya adalah menggunakan narrative seperti membuat misi, dari environment, dan sebagainya. Di dalam buku Raph Koster (2005), dia menjelaskan tentang bagaimana otak manusia menikmati proses informasi di sekitarnya agar mudah diproses nanti. Tingkat kesenangan player itu dari bagian learning, mechanic, dan sebagainya. Jika player memahami mechanic dan pattern dari game secara mudah akan membuat pemain cepat bosan. Sehingga level design sangatlah krusial.

Rouse(2004) mengatakan role dari level design itu sangat kompleks. Membuat satu level berarti menciptakan map layout secara fisik, menciptakan narasi, menaruh object, mengintegrasikan nilai estetika, problem solving, dan sebagainya. Karena tingkat kompleksnya dari membuat level, level design

termasuk bagian yang cukup sulit untuk diciptakan agar bisa menarik sekaligus menantang bagi para pemain. Level designing bisa dilakukan dengan cara manual atau dengan procedural content generation. Untuk menciptakan level yang dinamis, setiap pembuatan konten harus di generate dengan komputer dengan rules-rules tertentu sehingga tetap menarik dan tetap bisa dimainkan.

2.3 Role Playing Game

Role Playing Game menurut Jennifer Grouling (2010) itu berarti menandakan bahwa selama permainan, kita berperan sebagai salah satu pemain imajinatif dimana pemain bertanggung jawab untuk melakukan pemilihan keputusan tentang apa yang akan dilakukan dan apa yang akan terjadi dengan karakter imajinatif tersebut.

2.4 Procedural Content Generation

Julian Togelius, Noor Shaker, dan Mark J. Nelson (2016) mengatakan bahwa procedural content generation merupakan pembuatan konten dari game secara algoritmik dengan input yang terbatas atau input tidak secara langsung. Dengan kata lain procedural content generation adalah sebuah computer software yang bisa menciptakan sebuah game content dengan sendirinya atau bersama-sama dengan banyak player dan desainer.

Daniel Hoohsyar, Moslem Yousefi dan Heuseok Lim (2017) mengatakan bahwa konten yang di generate secara otomatis sedang menjadi topik yang hangat untuk diteliti karena pembuatan secara manual dinilai menghabiskan waktu dan

mengeluarkan uang yang tidak sedikit. Bahkan Daniel Hoohsyar, Moslem Yousefi dan Heuseok Lim (2017) mengatakan procedural content generation sangat penting untuk game development dan dianggap menjadi sangat penting di masa depan nanti. Mengapa bisa penting? Karena manusia itu mahal dan lambat dan saat mendvelop game kita membutuhkan banyak manusia. Bahkan game sekarang dianggap biasa jika dikerjakan oleh ratusan orang dalam waktu bertahun-tahun membuat profit dari game berkurang dan jumlah dari developer yang sanggup untuk membayar tenaga manusia menjadi berkurang. Julian Togelius, Noor Shaker, dan Mark J. Nelson (2016) bahkan mengatakan developer yang bisa mengganti beberapa artist dan designer bisa mendapatkan keuntungan yang sangat kompetitif karena game bisa di develop lebih cepat tanpa mengurangi kualitas dari game

Topik Procedural Content Generation dinilai masih baru. Julian Togelius, Alex J. Champandard, Pier Luca Lanzi, Michael Mateas, Ana Paiva, Mike Preuss, dan Kenneth O. Stanley (2016). mengatakan bahwa komunitas riset PCG baru dibentuk beberapa tahun lalu dan workshop pertama yang seluruhnya menjelaskan tentang PCG baru dibuka 2011 lalu. Para periset yang meneliti tentang procedural content generation juga lebih berfokus terhadap bagaimana menciptakan algoritma yang bisa diadaptasi dan bisa di kontrol karena sebagian besar algoritma yang digunakan sifatnya cepat, simple dan sulit untuk dikontrol. Dengan algoritma yang di riset ini diharapkan bisa menciptakan sebuah tipe game baru dan menciptakan cara baru untuk mendvelop game.

Julian Togelius, Noor Shaker, dan Mark J. Nelson (2016) juga menekankan bahwa kata kunci dari Procedural Content Generation adalah konten dimana konten merupakan sebagian besar dari sebuah game seperti maps, levels, game rules, textures, stories, items, quests, music, weapons, vehicles, etc. Julian Togelius, Noor Shaker, dan Mark J. Nelson (2016) juga menekankan bahwa game engine bukanlah konten, *NPC Artificial Intelligence* juga bukan konten karena AI biasanya dipakai untuk belajar memainkan permainan tersebut sedangkan procedural content generation sifatnya lebih fuzzy, hanya mengenerate sesuatu, tidak perlu adanya proses learning.

Julian Togelius, Noor Shaker, dan Mark J. Nelson (2016) juga mengatakan bahwa pengertian dari kata *procedural* dan *generation* artinya kita berhadapan dengan computer procedures atau algoritma yang menciptakan sesuatu yang artinya sebuah PCG method pasti akan memberikan output. System akan dianggap sebagai sistem PCG jika sistem tersebut menggunakan PCG method untuk mengenerate sesuatu di salah satu bagiannya. Sebagai contoh game *The Legend Of Zelda*, tanpa adanya input setiap kali tool nya di run sebuah level akan tercipta. Contoh lain adalah sebuah software yang *generate* sebuah board game yang komplit, seimbang dan bisa dimainkan.

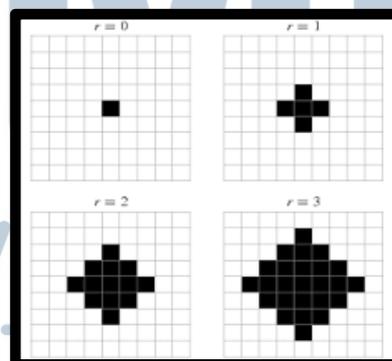
2.5 Cellular Automata

Cellular automata dibuat oleh Von Neumann pada tahun 1940 untuk model *biological self-reproduction*. Cellular automata sudah lama digunakan di bidang *computation, physical, dan biological application*.

Louis D'Alotto (2011) mengatakan bahwa sistem cellular automata tidak bekerja seperti sistem seperti biasanya, yang pasti berevolusi menjadi maksimal, cellular automata itu merupakan sebuah sistem yang sudah dikenal baik untuk modelling dan bisa bekerja sendiri sehingga apabila dimulai dengan data yang berantakan, evolusi dari cellular automata dapat men-generate struktur yang terorganisir.

Penelitian John Von Neumann (1966) mengatakan *cellular automata* memiliki element yang disebut sebagai sel. Setiap sel terhubung dengan tetangganya sehingga terhubung menjadi satu atau lebih dimensi grid. Setiap titik sel berubah sesuai dengan waktu dan menghasilkan nilai yang baru sesuai dengan tetangga yang terhubung menggunakan rules yang ditetapkan.

Cara untuk memproses cellular automata ialah pertama diberikan *set of rules* beserta *infinite sheet of papers*. Setiap kotak memiliki statenya tersendiri dan *cell* sebelahnya merupakan *neighborhood* yang bisa direpresentasikan dengan *von neumann neighborhood*. Setiap sel dibandingkan dengan *sets of rules* dan diganti *state*-nya sesuai rules-nya.

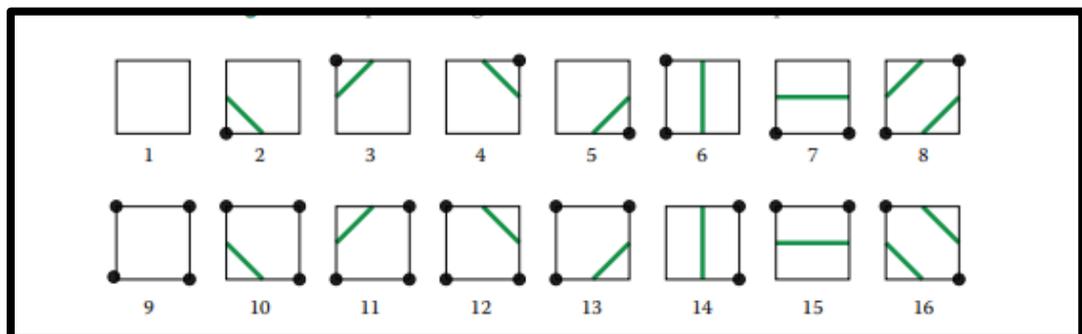


Gambar 2.1 Von Neuman Neighborhood.
diambil dari : <http://mathworld.wolfram.com>

Daniel Shiffman (2012) mengatakan *cellular automata* seperti *pattern* di dalam *grid of cells*. Dengan *set of rules* yang simpel dapat membuat *pattern* menciptakan salinan dari dirinya di *grid* tersebut. Hal tersebut menyebabkan *cellular automata* dapat digunakan di dalam implementasi dari *procedural content generation*.

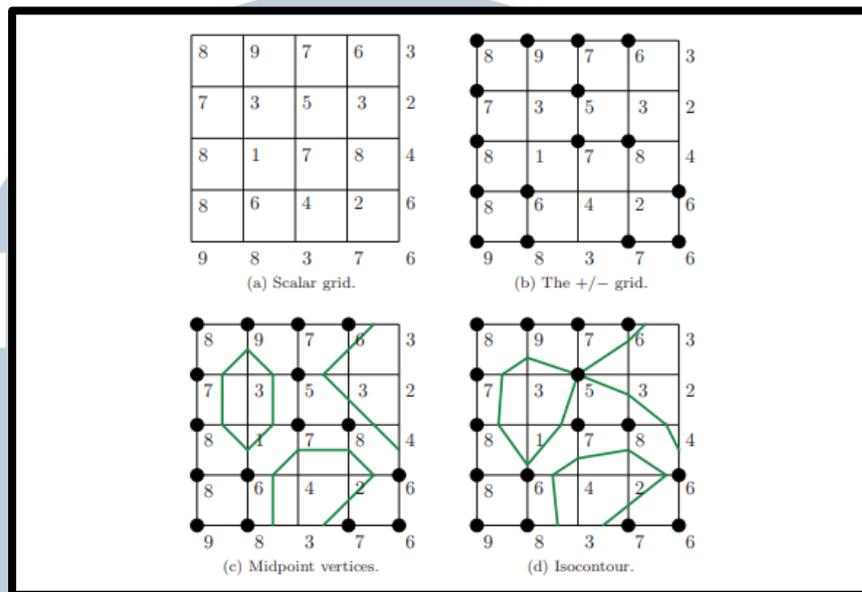
2.6 Marching Square

Menurut Rephael Wenger (2013), algoritma *Marching square* merupakan sebuah algoritma *isovalue* grafik yang mempresentasikan sesuatu agar menjadi sebuah nilai yang nantinya nilai tersebut berupa *vertices* grid 2D. *Marching square* memiliki 3 tahap untuk dijalankan. Tahap yang pertama adalah melihat tabel *isocountour*, lalu setiap grid nya lihat *edge* di dalam *isocontour*, lalu yang terakhir adalah menghitung koordinat *vertex* menggunakan *linear interpolation*.



Gambar 2.2 Bentuk tiap nilai binary dari marching square.
diambil dari : buku isosurface

Setiap nilai true di dalam setiap *grid* akan menambah nilai binary yang nantinya setiap angka yang dihasilkan dari binary akan menghasilkan bentuk yang berbeda-beda.



Gambar 2.3 Contoh algoritma marching square. diambil dari buku isosurface

2.7 Game Experience Questionnaire

Game experience questionnaire (GEQ) merupakan salah satu metode untuk melakukan evaluasi terhadap permainan yang dikembangkan. IJsselsteijn, W.A., de Kort, Y.A.W. dan Poels, K. (2013) mengatakan bahwa GEQ struktur modular yang isinya *core questionnaire*, *social presence module*, *post-game module*. Ketiga module tersebut diisi setelah pemain menyelesaikan sesi bermain permainan.

Core questionnaire merupakan inti dari GEQ. *Core questionnaire* menilai permainan dari 7 komponen yaitu *immersion*, *flow*, *competence*, *positive and negative effect*, *tension*, dan *challenge*. *Social presence module* menilai psikologis dan tindakan pemain dengan entitas lain baik virtual maupun pemain lain. *Post-game module* menilai apa yang pemain rasakan setelah selesai memainkan permainan.