

Implementasi Visual Programming untuk Membangun Permainan *Top-Down Shooter* Bug Wars Menggunakan Scratch



Computational Thinking Kelompok 7:

Delfia Nur Anrianti Putri	(21/474882/PA/20438)
Gelora Damayanti Manalu	(21/480851/PA/20914)
Farrel Amroe Azhari	(21/476754/PA/20600)
Rabbani Nur Kumoro	(21/472599/PA/20310)
Wella Wineke Prachinta	(22/506328/NUGM/01018)

A. Latar Belakang

Dalam pembuatan *game*, terdapat banyak aspek yang harus diperhatikan agar dapat mencapai tujuan dan agar pemain mendapatkan *user experience* yang bagus. Oleh karena dibutuhkan suatu tahapan perancangan untuk mengembangkan suatu *game*. Tahapan Perancangan Menurut Eko Nugroho (designer/Co-Founder Kummara), proses pengembangan *game* terbagi secara garis besar menjadi beberapa tahapan sebagai berikut: [1]

1. Tahap Riset dan Penyusunan Konsep Dasar, pada tahap ini ide dasar, objektif, tema, *target audience*, teknologi, media (*platform*), serta berbagai batasan lain dirumuskan. Tahapan riset ini menjadi sebuah tahapan krusial, berbagai elemen dasar dari sebuah *game* yang akan disusun.
2. Perumusan *Gameplay*, pada tahap ini *game designer* merumuskan *gameplay* atau *game mechanic* yang akan digunakan dalam sebuah *game*. *Gameplay* mengatur bagaimana pemain dapat memenuhi objektif dalam *game* dan mendapatkan pengalaman bermain yang menyenangkan.
3. Penyusunan Asset dan Level Design, dalam tahapan ini *game designer* fokus pada penyusunan konsep dari semua karakter serta asset (termasuk suara atau musik) yang diperlukan dan juga level design atau pengelompokan tingkat kesulitan serta berbagai asset yang tepat pada tiap level (jika ada lebih dari satu level) agar *game* dapat menghadirkan pengalaman bermain yang optimal.
4. *Test Play (Prototyping)*, pada tahapan ini sebuah *prototype* atau *dummy* dihadirkan untuk menguji *gameplay* serta berbagai konsep yang telah tersusun, baik dalam tiap level maupun secara keseluruhan, serta melakukan berbagai perbaikan yang diperlukan.
5. *Development* pada tahap ini seluruh konsep (karakter dan asset) yang sebelumnya telah tersusun mulai dikembangkan secara penuh, *game engine* mulai dikembangkan, dan semua elemen mulai dipadukan.
6. *Alpha / Close beta Test (UX - Initial Balancing)*, fokus utama pada tahap ini adalah untuk mengetahui apakah semua komponen utama dari *game* telah mampu memberikan *user experience* seperti yang diharapkan sekaligus juga untuk mendeteksi adanya masalah teknis yang belum terdeteksi pada tahapan sebelumnya.
7. Rilis, pada tahap ini *game* sudah siap untuk dirilis dan diperkenalkan pada target pemainnya. Ketika sebuah *game* telah dirilis untuk publik bukan berarti proses pengembangan selesai, *game* umumnya terus dioptimalkan. Hal ini untuk memastikan bahwa *game* yang dihadirkan benar-benar mampu memberikan pengalaman bermain yang maksimal.

Dalam pengembangan *game* ini, dipilih tipe *game top-down shooter* menggunakan serangga atau *bug* sebagai karakter utamanya. Alasan dipilihnya tipe *game* ini adalah karena tipe *game top-down shooter* mudah untuk dimengerti dan dipahami serta memiliki *gameplay* pertarungan yang sangat menantang untuk

dimainkan karena *player* terus berupaya untuk membunuh lawan agar mendapatkan skor setinggi mungkin. Tema karakter serangga atau *bug* dipilih karena perilakunya yang memiliki tendensi kuat untuk menyerang sesama spesiesnya untuk memenangkan persaingan dalam hal memenuhi kebutuhan bertahan hidupnya, yang mana perilaku ini sesuai dengan konsep dari tipe *game top-down shooter*.

Pada permainan ini, karakter utama dari *game* adalah seekor *ladybug* yang berusaha untuk bertahan hidup dengan cara menghindari dari serangan *bug* lawan, memakan apel untuk menambah tenaga, dan menyerang lawan sebagai mekanisme pertahanan dan perlawanan. Karena *ladybug* terkenal dengan metode pertahanan dan perlawanan menggunakan emisi bau yang ofensif, pada permainan ini, karakter *player* dapat menyerang musuh dengan melemparkan senyawa bebauan tersebut dalam bentuk proyektil.

B. Penjelasan Blok Kode

1. Player Bug

Pada *game* ini, *player bug* adalah karakter dalam permainan yang dikendalikan oleh pemain. Pemain dapat memberikan input untuk mengendalikan arah gerak *bug* dengan menekan tombol pada *keyboard* 'W', 'A', 'S', 'D' atau *pointer up*, *left*, *down*, dan *right* untuk menggerakkan ke arah atas, kiri, bawah, dan kanan. Ketika *player bug* bergerak, *background* pada *game* turut bergerak layaknya posisi kamera bergeser mengikuti karakter pemain. Pergerakan dari *player bug* sendiri dimulai dari $x = 0$ dan $y = 0$ serta terbatas pada area permainan, yaitu posisi x di antara -400 hingga 400 dan posisi y di antara -825 hingga 825. Variabel *scroll x* dan *scroll y* digunakan untuk mengindikasikan posisi pemain pada latar belakang, yang mana variabel ini akan digunakan untuk membuat latar belakang mengikuti gerak *player bug*. Pada awal permainan, variabel ini diatur sehingga *SCROLL X* = 100 dan *SCROLL Y* = 100 untuk memberikan efek *scroll horizontal*.

Pemain juga dapat melontarkan proyektil yang dapat mengurangi HP musuh sebanyak 1 setiap proyektil yang mengenai musuh dengan menekan tombol spasi atau mengklik layar *game* dengan *loading time* 0.2 detik. Arah dari proyektil dari pemain didefinisikan berdasarkan posisi *mouse* (*mouse x* dan *mouse y*) yang juga merupakan arah dari *player bug* dan kecepatan proyektil ditentukan berdasarkan jarak ke *pointer mouse*. Pergerakan proyektil ini memanfaatkan posisi x , y , *SCROLL X*, dan *SCROLL Y* dengan menghitung $x - \text{SCROLL X}$ dan $y - \text{SCROLL Y}$ juga mengubah posisi x menjadi $x - (x - \text{SCROLL X})/10$ dan y menjadi $y - (y - \text{SCROLL Y})/10$ untuk memberikan efek proyektil yang semakin pelan seiring waktu. Untuk setiap *enemy bug* yang tewas karena terkena proyektil *player bug*, *score* akan bertambah 1 (akan ditampilkan selama *gameplay* berlangsung).

HP atau *Health Point* dari *player bug* adalah 20 dan akan berkurang 1 setiap kali terkena 2 proyektil dari musuh. Ketika *player bug* terkena proyektil dari musuh, terdapat efek visual yang akan mengindikasikan hal tersebut. HP pada *player bug* ditunjukkan melalui *health bar*. Bar hijau akan berkurang 1 panjangnya apabila pemain terkena proyektil musuh.

Ketika HP *player bug* habis, *player bug* akan perlahan mengecil dan hilang dari arena permainan yang menandakan bahwa *player bug* telah mati, serta seluruh objek dalam permainan saat ini diberhentikan dan layar permainan berakhir ditampilkan beserta teks yang merangkum gameplay yang baru saja dilakukan. Rangkuman *gameplay* terdiri atas lama waktu permainan (yang ditandai dengan variabel *Time Alive* yang merupakan pembulatan dari timer) beserta skor.

2. Enemy

Enemy bug adalah objek yang akan menjadi lawan dari *player* selama *game* berlangsung. Mekanisme *spawn* para *enemy* sendiri adalah setiap *enemy* akan di *spawn* setiap 1.6 detik dengan lokasi yang *random* berdasarkan variabel yang di *generate* secara *random*. Terdapat dua jenis *bug* dengan jumlah HP yang berbeda.

Untuk *spawning bullet* dari *enemy* sendiri, saat di *spawn* akan bergerak ke arah posisi *player* pada saat di *spawn* dengan memanfaatkan posisi *x*, *y*, *SCROLL X*, dan *SCROLL Y* dengan menghitung *x-SCROLL X* dan *y-SCROLL* untuk *spawn* awal sesuai posisi *enemy* itu sendiri. Selain itu, tak lupa peluru dihadapkan ke arah *player* setelah 0.9 detik peluru akan perlahan menghilang. Pergerakan dari *bullet enemy* sendiri di atur oleh *variable aim x* dan *aim y* yang seiring mendekati *player* akan melambat dengan dikalikan nilai 0.9 dan dengan nilai *aim x* dan *aim y* yang dikurangi *variable SCROLL X* dan *Y* dapat membuat peluru yang melewati batas *stage* menjadi tidak ditampilkan. Ketika peluru menyentuh *player*, maka akan menghilang.

Untuk pergerakan dari *enemy* sendiri mengacu pada posisi *player* pada saat itu yang didefinisikan dengan *x-SCROLL X* dan *y-SCROLL Y* dengan *loop block code forever* untuk mengatur posisi akhir yang menjadi tujuan pergerakan *enemy*, yaitu *player* sendiri. Untuk kecepatan pergerakan didefinisikan pada blok kode *go to player* dengan pergerakan 1 *pixel* bergantung posisi relatif musuh terhadap *player*.

Saat *enemy* menyentuh peluru *player*, akan dikurangi 1 HP nya dari HP yang ada. Untuk *enemy* berwarna kuning memiliki HP sebesar 3 dan untuk *enemy* berwarna hitam memiliki HP sebesar 5.

3. HP Booster

HP Booster merupakan sebuah *sprite* atau aset yang berbentuk apel berwarna merah. Aset ini muncul pada layar secara acak setiap 3-5 detik. Aset ini memiliki ukuran 12 dan direction 90 yang akan konstan selama permainan berlangsung. HP Booster akan menambah HP player sebanyak 4 (jika hasil penambahan melebihi 10 maka HP akhir pemain adalah 10) ketika player menembakkan projectile sebanyak 5 kali ke arah ikon apel. Ketika sudah ditembak sebanyak 5 kali, *sprite* ini akan hilang dan akan muncul lagi ditempat acak lain.

4. Tree

Tree sangat mirip konsepnya dengan HP Booster. Tree merupakan sebuah aset dengan bentuk pohon berwarna hijau dan memiliki ukuran yang berbeda-beda. Aset ini akan muncul pada tempat random setiap 3-5 detik. Aset ini digunakan untuk tempat perlindungan player. Jika player tertutup oleh aset ini, maka *enemy* tidak dapat menembak *player* sebab peluru akan menghilang ketika menyentuh warna hijau yang berada di bagian paling luar *tree*. Namun ketika aset ini terkena projectile sebanyak 5 kali, maka aset ini akan hilang.

5. Scrolling Background

Scrolling background adalah sebuah aset yang digunakan untuk memfokuskan tampilan pada player ketika player sedang bergerak. Aset ini tidak terlihat karena di-set pada mode tampilan *hidden*. Aset ini akan membuat kamera fokus kepada player kemanapun player bergerak sehingga pemain tidak perlu menggeser layar ketika player bergerak keluar dari tampilan.

6. Text Renderer

Text Renderer berfungsi untuk menampilkan *text* di layar saat *game* telah berakhir atau *game over*. Tampilan yang dihasilkan oleh blok kode ini berisikan lama waktu permainan, skor pemain atau jumlah bug yang telah dikalahkan oleh pemain, beserta instruksi untuk memulai kembali permainan baru (dapat dilakukan dengan menekan tombol space).

Terdapat blok kode *type* yang memiliki argumen sebuah string untuk ditampilkan, posisi X dan posisi Y atau lokasi ditampilkannya *string*, dan ukuran *text*. Ketika blok kode *type* dipanggil, proses yang terjadi antara lain *clear graphic effect*, lalu *brightness game* di set menjadi 90, dan dilakukan iterasi sebanyak jumlah karakter pada string untuk menampilkan karakter pada kostum sesuai dengan karakter input yang pada posisi X dan Y dengan size yang telah ditentukan. Hasilnya, teks yang dihasilkan akan muncul satu-persatu pada layar.

7. Menu

Tampilan menu awal *game* merupakan tampilan awal dari *game* yang dibuat. Pada menu ini pemain harus menekan tombol start untuk menjalankan atau memainkan permainan ini. Setelah menekan tombol start maka pemain akan

masuk ke dalam *game*. Selain tombol start, pemain dapat menekan tombol tutorial untuk melihat sebuah *guideline* yaitu cara bermain.

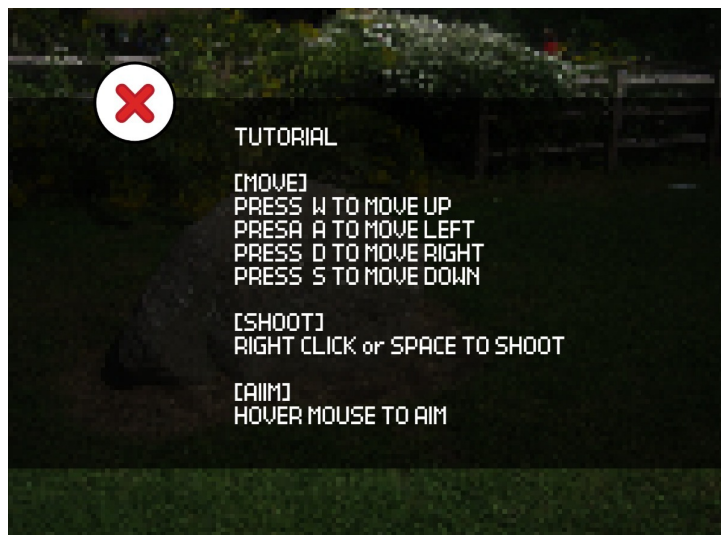
Ketika kursor *mouse* diarahkan ke atas tombol start maupun tutorial, kostum akan di-*update* untuk menampilkan gambar tombol yang seakan sedang ditekan. Saat tombol tutorial ditekan, akan muncul *sprite* yang bernama fade, tutorial-content, dan exit. Tiga *sprite* tersebut menampilkan isi dari halaman tutorial. Selanjutnya, ketika tombol *exit* ditekan, menu awal *game* akan ditampilkan.

C. Cuplikan

Tampilan awal ketika ingin memulai *game*. Terdapat 2 pilihan, yaitu start dan tutorial.



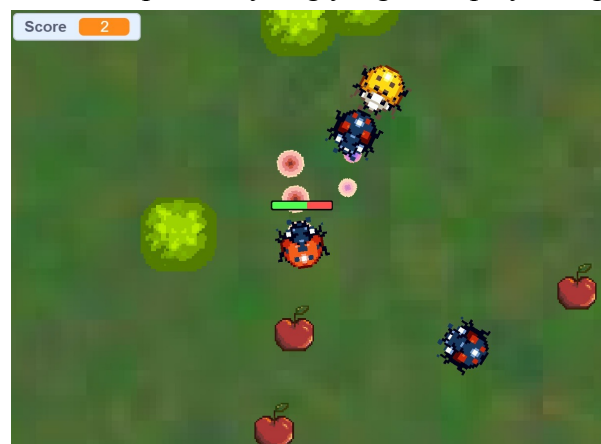
Tampilan ketika *player* memilih *button* tutorial, terdapat beberapa aturan yang digunakan dalam permainan. Jika *player* telah membaca seluruh aturan permainan, maka dapat menekan button “X” pada kiri atas.



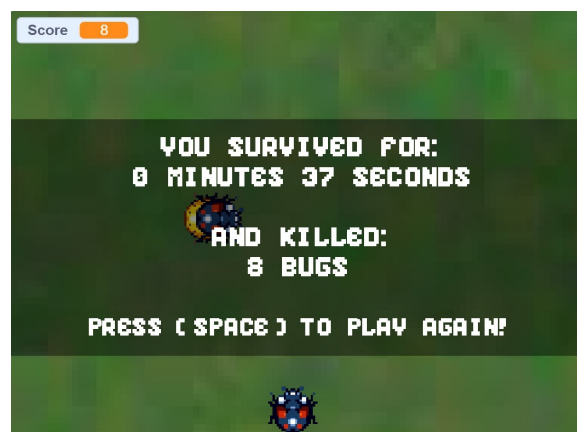
Tampilan ketika *player* memilih button start, terdapat karakter player bug dan enemy bug. *Player* juga dapat melihat nyawa dari karakter player bug melalui kapasitas hp yang ada di bagian atas player bug.



Tampilan ketika hp karakter player bug sudah mulai berkurang, dan terdapat tampilan score yang memberi tahu berapa enemy bug yang telah player bug bunuh.



Tampilan ketika *game over* saat HP player bug sudah habis, terdapat beberapa informasi seperti waktu lamanya player dapat bertahan dalam permainan dan juga berapa jumlah enemy bug yang dibunuh. Juga terdapat informasi untuk memulai kembali permainan.



D. Saran dan Pengembangan

Beberapa hal yang mungkin dapat berguna untuk pengembangan *game* ini ke depannya adalah sebagai berikut:

- Penambahan jumlah stage dan variasi *gameplay* di setiap stagenya.
- Penambahan jenis senjata dan efek tembakan yang berbeda.
- Penambahan bonus object, misalnya objek/benda melayang yang jika ditembak akan membuat laju musuh menjadi lebih lambat dan sebagainya.
- *Displaying* petunjuk permainan dilakukan setiap pemain akan memulai *gameplay* dan bukan hanya di awal permainan.
- Pemberian tombol untuk *pause game* dan *exit game* ketika *player* berada dalam *gameplay*.
- Penambahan papan skor untuk menyimpan skor dari pemain.

Diharapkan saran-saran pengembangan di atas dapat membuat *game* ini memberikan pengalaman permainan yang lebih optimal untuk pemain. Selain itu, diharapkan *game* ini dapat menjadi inspirasi dan mengembangkan *game* serupa yang lebih baik ke depannya.

E. Source Code

<https://scratch.mit.edu/projects/767916601/>

F. Referensi

[1] Nugroho, Eko. 2013. 7 Tahap Pengembangan Game. Kompas. Tersedia pada:
<http://tekno.kompas.com/read/2013/08/21/1226508/7.Tahap.Pengembangan.Game>