

BAB IV

DESKRIPSI PEKERJAAN

Dalam Bab IV ini akan dibahas mengenai deskripsi pekerjaan selama melakukan kerja praktik di CV Sego Njamoer. Pada pelaksanaan kerja praktik, penulis merancang sebuah proyek game Android berjudul Sego Njamoer sebagai media promosi digital bernama *Advergame*. Dalam kerja praktik ini penulis bertugas sebagai programmer dan animator, dengan rincian tugas sebagai

berikut:

1. Merancang sistem permainan dan *gameplay*
2. *Coding* psistem permainan
3. Menyusun *sprite* untuk animasi karakter
4. Mengimport aset *game*
5. Mengekspor dalam bentuk file *.apk

4.1 Analisa Sistem

Kerja praktik yang dilaksanakan adalah sebagai berikut:

Nama Perusahaan : CV. Sego Njamoer

Divisi : *Marketing (Game Developer)*

Tempat : Jl. Klampis Semolo Timur XII No. 2

Kerja praktik dilaksanakan selama enam minggu, dimulai pada 25 Juli 2016 hingga 2 September 2016, dengan alokasi waktu senin sampai jum'at pada pukul 08.00-16.30 WIB.

4.2 Posisi Dalam Instansi

Pada saat pelaksanaan kerja praktik, posisi yang didapat adalah sebagai *game developer* di bidang Marketing, yang memiliki tugas memrogram *game* Sego Njamoer sebagai media promosi.

4.3 Kegiatan Selama Kerja Praktik Di CV Sego Njamoer

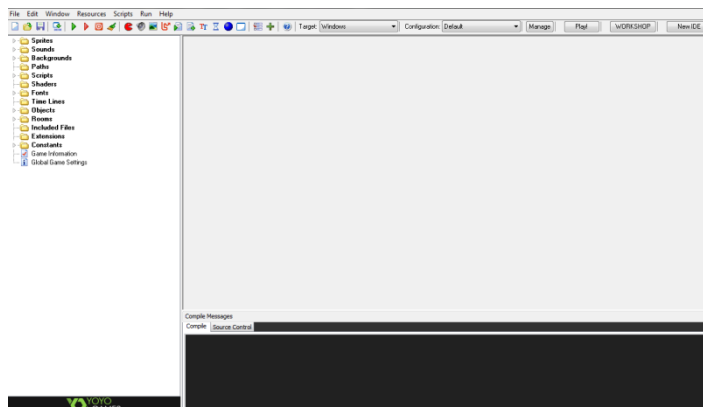
Kegiatan sehari-hari yang dilakukan selama melaksanakan kerja praktik di CV. Sego Njamoer dilaporkan dengan rincian per minggu selama enam minggu. Laporan kegiatan disertai gambar hasil pekerjaan serta keterangan pada tiap gambar.

1. Minggu ke-1

Minggu pertama melakukan kerja praktik di CV.Sego Njamoer diawali dengan koordinasi tim *artist* untuk menentukan jadwal dan perencanaan produksi *game*. Selain itu penentuan konsep cerita dalam *game* juga dilakukan pada minggu pertama. Setelah terkordinasi dan terencana dengan baik, ditentukan secara pasti pembuatan *game* Sego Njamoer diprogram menggunakan aplikasi Game Maker Master Collection dengan logo seperti gambar 4.1 di bawah ini. Tampilan aplikasi Game Maker dapat dilihat pada gambar 4.2 di halaman selanjutnya.



Gambar 4.1 Logo Game Maker
(Sumber: Olahan Penulis)



Gambar 4.2 Tampilan Aplikasi Game Maker
(Sumber: Olahan Penulis)

Setelah mempelajari struktur dasar GML (Game Maker Language) sebagai jenis bahasa pemrograman yang digunakan dalam Game Maker, kemudian dilakukan perancangan karakter *dummy* sebagai aset yang bersifat sementara agar pemrograman dasar *game* dapat dilakukan, desain karakter *dummy* dapat dilihat pada gambar 4.3 di bawah, hal ini diperlukan agar tidak memakan banyak waktu saat menunggu tim *artist* merancang aset *game* yang asli.



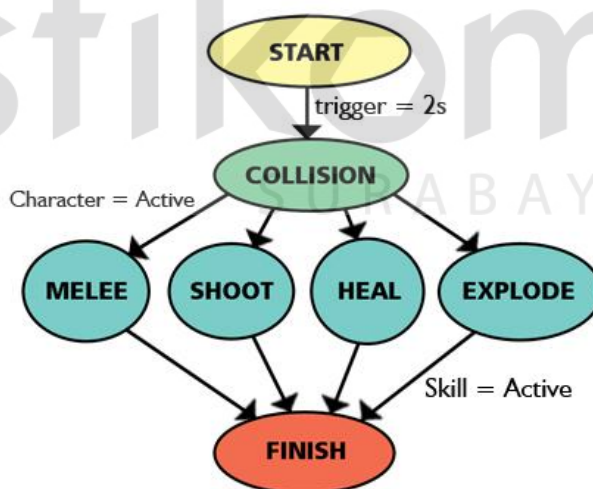
Gambar 4.3 *Dummy* Untuk Karakter Sego Njamoer
(Sumber: Olahan Penulis)

Penyusunan program dasar *game* dimulai pada tahap ini, tepat setelah aset sementara *dummy* selesai dirancang. Dasar sistem permainan yang diprogram untuk *game* Sego Njamoer cukup sederhana, yakni dengan

menekan tombol karakter di pojok kanan atas untuk *summon* (memanggil) produk Sego Njamoer. Secara otomatis karakter produk Sego Njamoer akan menyerang musuh di depan mereka dan menyentuh kubu lawan agar dapat memenangkan permainan. Secara garis besar, mekanisme permainan utama dalam sebuah *game* dapat disebut sebagai *Finite State Machine*, yang terdiri dari:

- a. Konsep *Gameplay*, yakni konsep dan genre yang diterapkan.
- b. Aturan Besar, yang mengatur jalan permainan dan bersifat mutlak.
- c. *Rule Base Character*, kondisi setiap karakter yang didefinisikan.

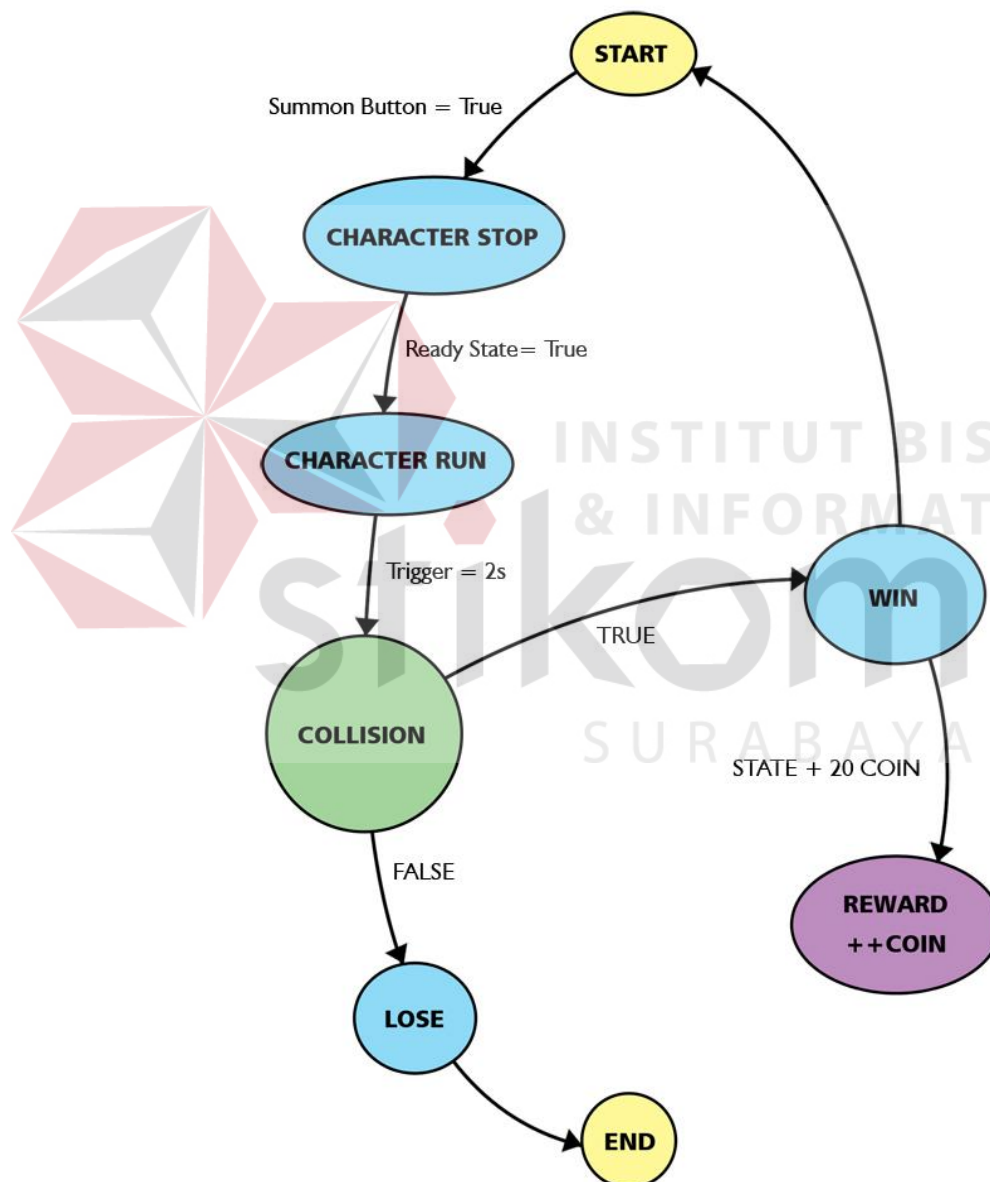
Seluruh *Finite State* tersebut akan dipetakan secara terstruktur untuk memudahkan penulis dalam merancang *game* Sego Njamoer. Gambar 4.4 di bawah merupakan peta dari kondisi karakter berdasarkan kemampuan (*skill*) yang dimiliki.



Gambar 4.4 Mendefinisikan *Collision* Setiap Karakter
(Sumber: Olahan Penulis)

Pada gambar 4.4 di atas telah didefinisikan bahwa setiap karakter dalam *game* Sego Njamoer memiliki kemampuan yang berbeda-beda. Terdapat

kemampuan *Melee* yang ahli menyerang di jarak dekat, *Shoot* untuk menembak jarak jauh, *Heal* adalah kemampuan untuk menyembuhkan, dan *Explode* untuk meledakkan musuh. Gambar 4.5 di bawah merupakan pemetaan *Finite State* yang digunakan untuk mendefinisikan *Rule Base* dari game *Sego Njamoer*.

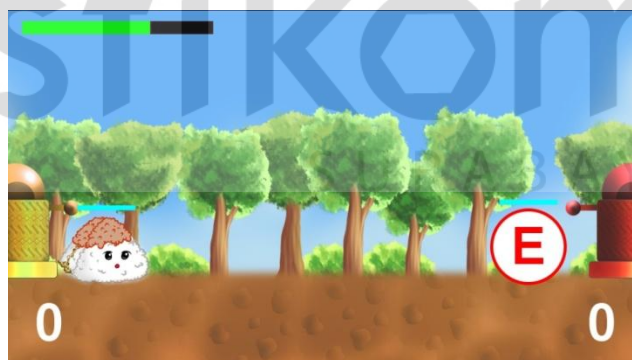


Gambar 4.5 *Finite State Machine*
(Sumber: Olahan Penulis)

Pada gambar 4.5 di halaman sebelumnya terdapat peta algoritma yang mendefinisikan *Finite State Machine* dalam *game* *Sego Njamoer*. Di awal permainan saat *state* tombol *summon* aktif, maka karakter akan keluar dan berhenti sesaat hingga *state ready* aktif selama dua detik. Kemudian karakter akan berlari menuju musuh, jika menang maka karakter akan mendapatkan *reward* berupa penambahan koin sebesar 20. Jika kalah karakter akan langsung menuju room *Game Over*.

2. Minggu ke-2

Pada minggu kedua, penulis melakukan *breakdown* dari pemetaan *Finite State* yang telah didefinisikan sebelumnya. *Breakdown* diperlukan untuk menjabarkan garis besar dan aturan dasar *gameplay* dari *game* *Sego Njamoer* agar penulis dapat melakukan *game programming* dengan mudah dan terstruktur.



Gambar 4.6 *Dummy* *Sego Njamoer* Menghampiri Musuh
(Sumber: Olahan Penulis)

Gambar 4.6 di atas menjelaskan *gameplay* dasar dari *game* *Sego Njamoer*, dimulai dari karakter *Sego Njamoer* dan Musuh yang muncul dari dua sisi (kiri dan kanan). *Bar* berwarna hijau di pojok kiri atas merupakan indikator waktu yang akan terus berkurang.



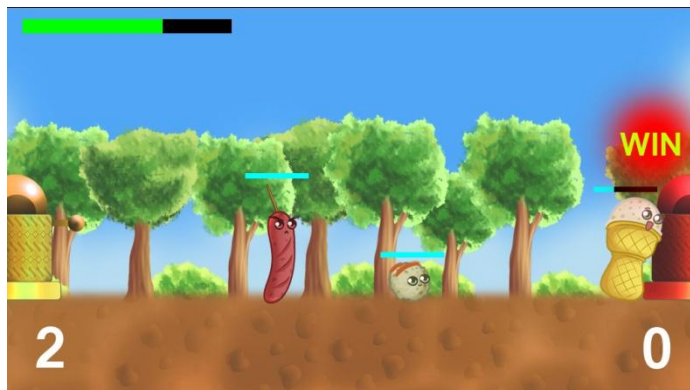
Gambar 4.7 *Dummy* Sego Njamoer Melawan Musuh
(Sumber: Olahan Penulis)

Karakter Sego Njamoer dapat dipanggil terus menerus selama *delay* telah terpenuhi, begitu pula dengan musuh, karakter musuh akan terus keluar untuk menyerang pasukan Sego Njamoer. Gambar 4.7 di atas merupakan ilustrasi saat pasukan Sego Njamoer menyerang pasukan musuh.



Gambar 4.8 *Dummy* Sego Njamoer Mengalahkan Musuh
(Sumber: Olahan Penulis)

Gambar 4.8 di atas menggambarkan karakter Sego Njamoer yang berhasil mengalahkan pasukan musuh. Saat pasukan musuh dikalahkan, *player* akan menerima tambahan koin yang dapat digunakan untuk memanggil tambahan karakter Sego Njamoer.

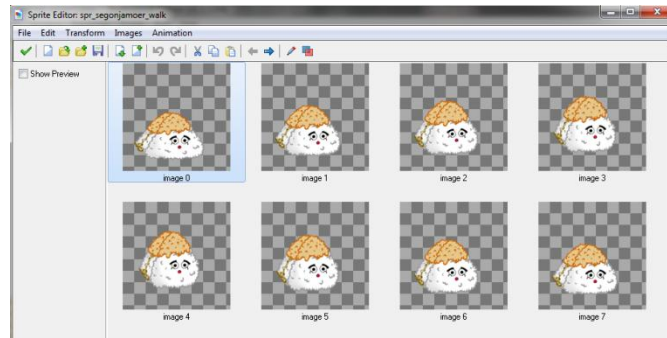


Gambar 4.9 *Dummy* Sego Njamoer Menang
(Sumber: Olahan Penulis)

Setelah mengalahkan beberapa musuh, pasukan Sego Njamoer secara otomatis melangkah maju hingga ke kubu lawan, saat kubu lawan berhasil disentuh pasukan Sego Njamoer, maka *player* memenangkan permainan dan otomatis akan melaju ke level selanjutnya. Gambar 4.9 di atas merupakan ilustrasi *dummy* Sego Njamoer memenangkan permainan.

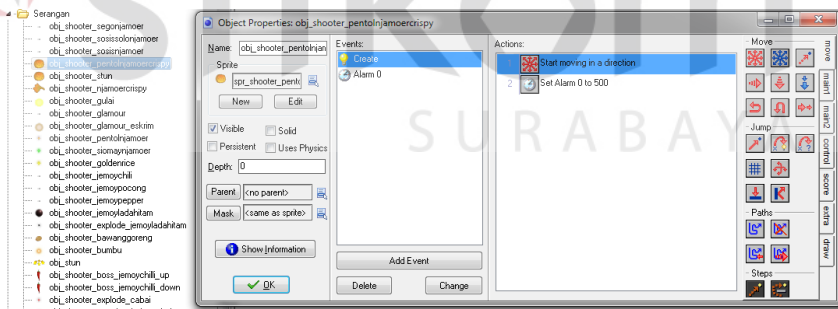
3. Minggu ke-3

Pada Minggu Ketiga tim *asrtist* menyelesaikan aset *sprite* untuk karakter Sego Njamoer dan musuh di dalam *game*. Penulis menyusun *sprite* tersebut dalam bentuk animasi di dalam Game Maker. Terdapat fitur *sprite combine* dalam Game Maker sehingga memudahkan penulis untuk membuat animasi atraktif dalam mendukung grafis *game* Sego Njamoer. Cara mengkombinasikan *sprite* cukup mudah, yakni dengan mengimport *sprite* sesuai urutan animasi. Terdapat dua jenis karakter yang akan diimport dalam bentuk *sprite*, yakni karakter untuk pasukan Sego Njamoer, dan karakter untuk pasukan Musuh. Untuk karakter Sego Njamoer terdapat dua belas karakter, dan karakter musuh terdapat sepuluh karakter termasuk *boss*.



Gambar 4.10 Mengimport *Sprite* Karakter Sego Njamoer
(Sumber: Olahan Penulis)

Gambar 4.10 di atas merupakan tampilan fitur *sprite import* di aplikasi Game Maker. Setelah dianimasikan, *sprite* yang terkumpul satu per satu diekspor dalam bentuk *object* agar dapat diprogram menggunakan GML. Menyusun *object* harus menyesuaikan dengan nama *sprite* agar saat *programming* dapat dilakukan dengan terstruktur dan mudah untuk mencegah kesalahan *letters*. Tampilan *object* dapat dilihat pada gambar 4.11 di bawah ini.



Gambar 4.11 Kumpulan *Object*
(Sumber: Olahan Penulis)

Gambar 4.10 di atas merupakan tampilan fitur *sprite import* di aplikasi Game Maker. Setelah dianimasikan, *sprite* yang terkumpul satu per satu diekspor dalam bentuk *object* agar dapat diprogram menggunakan GML.

Menyusun *object* harus menyesuaikan dengan nama *sprite* agar saat memasukkan kode pemrograman tidak dianggap sebagai *object* yang sama.

```

1
2 with (argument0) draw_healthbar ( x+30, y-120, x-30, y-1
3
4 if (!pause)
5 {
6     pause = 1;
7     instance_deactivate_all(true);
8     instance_create(0,0,obj_blackalpha);
9     instance_create(1024/2,576/2-30,obj_pause);
10    instance_create(1024/2-100,350,obj_btn_replay);
11    instance_create(1024/2+100,350,obj_btn_back);
12 }
13 else
14 {
15     pause = 0;
16     instance_activate_all();
17     instance_deactivate_object(obj_pause);
18     instance_deactivate_object(obj_blackalpha);
19     instance_deactivate_object(obj_btn_replay);
20     instance_deactivate_object(obj_btn_back);
21 }
22

```

Gambar 4.12 *Script* Dalam Game Maker
(Sumber: Olahan Penulis)

Gambar 4.12 di atas merupakan *script* dalam Game Maker yang digunakan untuk menambahkan fitur dalam permainan, seperti fitur *pause*, *start*, dan *replay*. Selain itu, untuk mengindikasikan kekuatan karakter Sego Njamoer dan Musuh, digunakan fitur *health bar* yang muncul di atas kepala setiap karakter sehingga *player* dapat mengerti kekuatan yang tersisa masing-masing karakter dalam *game* Sego Njamoer. Selama melakukan *game programming*, digunakan sistem *global variable* yang akan berfungsi sebagai *variable* utama yang mengatur seluruh komponen parameter dalam *game* Sego Njamoer, seperti level, koin, dan penambahan pasukan baru. Secara garis besar, eksistensi *global variable* sangat memudahkan seorang *game developer* dalam menangani jumlah *variable* yang sangat tinggi,

sehingga hanya dengan memanggil satu *global variable* saja maka seluruh *variable* dalam kumpulan *script* akan aktif secara otomatis. Gambar 4.13 di bawah merupakan contoh *global variable* yang digunakan dalam *game* *Sego Njamoer*.

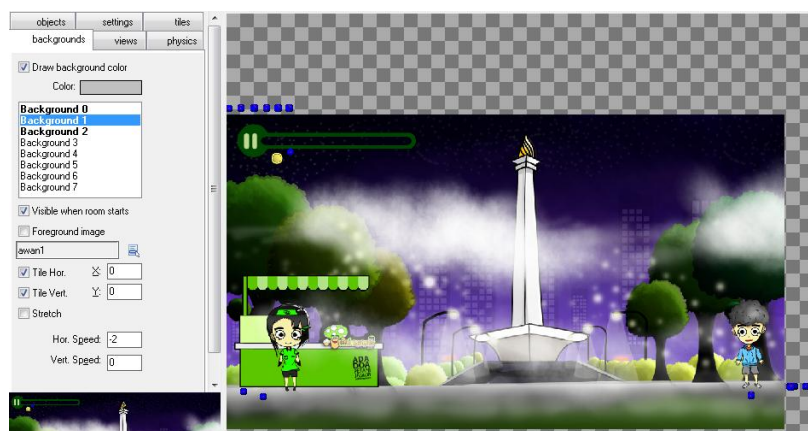
```

1 global.level = 1;
2 global.maplevel= 30;
3 global.tips = 0;
4 global.pasukan = 0;
5 global.tamat = 0;

```

Gambar 4.13 *Global Variable*
(Sumber: Olahan Penulis)

Setelah melakukan *game programming*, langkah selanjutnya adalah menyusun komponen-komponen *object* dalam *room* yang telah disediakan oleh aplikasi *Game Maker*. Fungsi fitur *room* adalah untuk menyatukan seluruh *object* agar dapat muncul di layar sehingga *player* dapat memainkan *game* *Sego Njamoer*. Dalam fitur *room* juga dapat digunakan untuk menggerakkan *background* agar tercipta efek yang mendukung atmosfer permainan, contohnya seperti efek awan atau kabur. Gambar 4.14 di bawah merupakan contoh fitur *room* yang digunakan.



Gambar 4.14 *Global Variable*
(Sumber: Olahan Penulis)

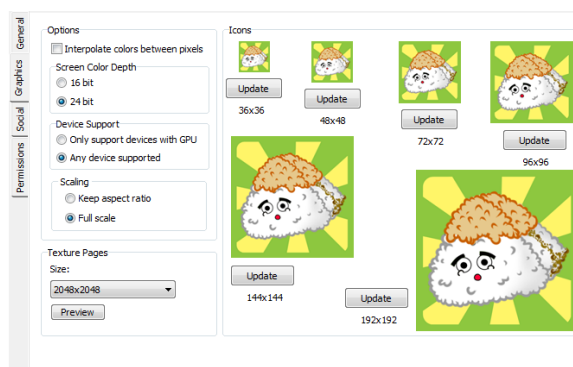
4. Minggu ke-4

Minggu keempat masuk dalam tahap *game testing*, seluruh komponen *object* yang telah diprogram dan telah disusun dalam *room* kemudian dikonversi menjadi aplikasi Android berekstensi *.apk . Konversi dilakukan dengan menggunakan Android SDK-NDK dan Java JDK yang ditracking ke dalam aplikasi Game Maker. Gambar 4.15 di bawah merupakan kolom untuk melakukan *tracking* Android SDK-NDK ke dalam Game Maker.



Gambar 4.15 Android SDK-NDK
(Sumber: Olahan Penulis)

Setelah itu masuk dalam tahap mengatur *icon* dan *splash screen* dalam kolom *global setting*. *Icon* diperlukan untuk mengidentifikasi *game* Sego Njamoer secara visual yang tampil pada *desktop* Android, sedangkan *splash screen* berisi identitas perusahaan Sego Njamoer. Gambar 4.16 di bawah merupakan kolom *global setting* untuk mengatur *icon game* Sego Njamoer.



Gambar 4.16 Game Icon
(Sumber: Olahan Penulis)

Setelah mengatur *icon* dan *splash screen*, dilakukan kompresi data menjadi *.apk agar dapat dioperasikan di *smartphone* dengan sistem operasi Android. Pada tahap ini masuk ke dalam *game testing* untuk melakukan *recheck* jalannya permainan dari awal hingga akhir. Pada gambar 4.17 di bawah merupakan *game testing* untuk melakukan *recheck* terhadap animasi *sprite*.



Gambar 4.17 *Game Testing*
(Sumber: Olahan Penulis)

Selain animasi *sprite*, dilakukan juga *game testing* untuk pengolahan tim pasukan Sego Njamoer, karena pada komponen tersebut memiliki komponen *script* yang rumit dan sering terjadi *bug* saat merubah pilihan tim. Terdapat dua tim, tim A dan tim B, setiap tim memiliki kemampuan yang berbeda-beda. Gambar 4.18 di bawah merupakan cuplikasi dari tim A.



Gambar 4.18 Tim A
(Sumber: Olahan Penulis)