

BAB IV

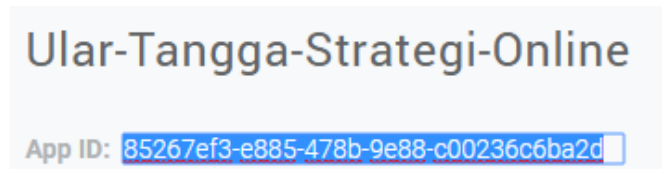
HASIL DAN PEMBAHASAN

4.1. *Production*

4.1.1. Pembangunan *Backend* Permainan

Pembangunan permainan *online multiplayer* ular tangga bergenre strategi ini diawali dengan pembuatan sebuah proyek aplikasi pada layanan *backend*. Pembuatan proyek ini dilakukan agar aplikasi dapat terhubung dengan layanan *backend* dan dapat menggunakan fitur-fitur di dalamnya. Pembuatan proyek ini dilakukan pada *browser* Google Chrome. Setelah membuat proyek, maka layanan *backend* akan memberikan sebuah kode pengenal yang digunakan untuk menghubungkan aplikasi ke layanan *backend*.

Kode pengenal pada layanan *backend* Photon Cloud dapat ditemukan pada halaman *dashboard* Photon Cloud. Kode pengenal pada layanan *backend* Photon Cloud berupa ID aplikasi yang dapat dilihat pada gambar 4.1.



Gambar 4. 1 ID Aplikasi pada Photon Cloud

ID aplikasi yang telah didapatkan kemudian dapat dipasang pada *plugin* Photon yang sebelumnya telah dipasang pada proyek permainan *online multiplayer* ular tangga bergenre strategi di Construct 2. Gambar hasil pemasangan ID aplikasi pada *plugin* Construct 2 dapat dilihat pada gambar 4.2. Pada gambar tersebut, ditunjukkan selain memasang ID aplikasi, penulis juga mengatur versi aplikasi (*AppVersion*), protokol (*Protocol*), wilayah (*Region*) dan tipe *hosting* (*HostType*). Pada properti versi aplikasi, penulis memasukkan versi aplikasi yang telah mencapai versi 14.2. Untuk protokol, penulis memilih *wss://* sebagai transmisi yang aman untuk melakukan konfigurasi jaringan ke server. Pada properti wilayah, dipilihlah *asia* karena wilayah tersebut merupakan *master server* yang terdekat dengan aplikasi yang dibangun. Yang terakhir penulis menggunakan Photon Cloud

sebagai server yang akan menangani permainan *online multiplayer* ular tangga bergenre strategi.

Properties	
AppId	85267ef3-e885-478b-9e88-c00236c6ba2d
AppVersion	14.2
Protocol	wss
Region	asia
HostType	Photon Cloud

Gambar 4. 2 Properti pada *Plugin Photon Cloud*

Berbeda dengan Photon Cloud, layanan *backend* Firebase tidak hanya memberikan kode pengenalan berupa kunci API, tetapi juga kode inisialisasi lainnya seperti *authentication domain* sebagai identitas dalam menggunakan fitur *authentication*, *database URL* sebagai identitas dalam menggunakan fitur *real-time database*, *project ID* sebagai identitas proyek yang dibangun, *storage bucket* sebagai identitas dalam menggunakan fitur *cloud storage*, dan *messaging sender ID* sebagai identitas pengirim dalam menggunakan fitur *cloud messaging*. Cuplikan dari kode inisialisasi dari proyek yang dibangun pada layanan *backend* Firebase dapat dilihat pada gambar 4.3.

```
apiKey: "AIzaSyAhchcq7S8ramBROmhaOPT_aJczbA3rl0k",
authDomain: "umy-snake-ladder.firebaseio.com",
databaseURL: "https://umy-snake-ladder.firebaseio.com",
projectId: "umy-snake-ladder",
storageBucket: "umy-snake-ladder.appspot.com",
messagingSenderId: "201754786022"
```

Gambar 4. 3 Kode Inisialisasi Proyek pada Firebase

Permainan *online multiplayer* ular tangga bergenre strategi ini menggunakan fitur dari layanan *backend* Firebase berupa Firebase *authentication* dan Firebase *real-time database*. Berdasarkan hal tersebut, penulis hanya memasang kunci API, *authentication domain*, dan *database URL* pada *plugin* Firebase yang sebelumnya telah dipasang pada proyek permainan *online multiplayer* ular tangga bergenre strategi di Construct 2. Gambar hasil pemasangan kunci API dan kode inisialisasi dari fitur *backend* yang digunakan dapat dilihat pada gambar 4.4.

Properties	
Api key	AlzaSyAhchcq7S8ramBR0mhaOPT_aJczbA3rl0k
Auth domain	ummy-snake-ladder.firebaseio.com
Database URL	https://ummy-snake-ladder.firebaseio.com

Gambar 4. 4 Properti pada *Plugin* Firebase

4.1.2. Implementasi *Coding* dan Antarmuka

Implementasi *coding* dan antarmuka merupakan penerapan dari rancangan sistem yang telah dibahas sebelumnya. Implementasi ini membahas penerapan *coding* dan tampilan antarmuka pada permainan yang dibangun menggunakan *game engine* Construct 2. Berikut adalah hasil dari implementasi *coding* dan antarmuka pada permainan.

1. Halaman *Log In*

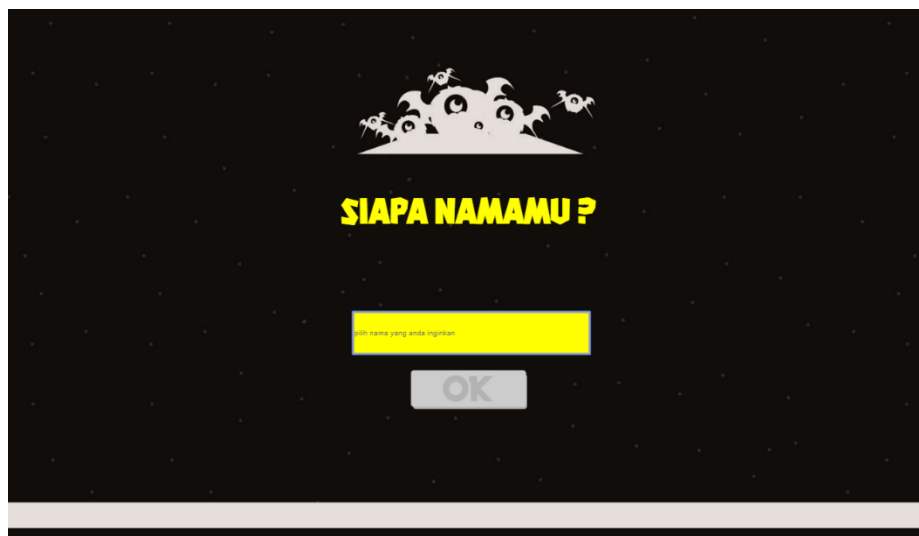
Pembangunan halaman *log in* pada aplikasi permainan *online multiplayer* ular tangga bergenre strategi ini dibuat berdasarkan BAB III sub bab 3.2.4 poin 10 bagian a. Pada halaman tersebut pemain diminta untuk melakukan *log in* sebelum memainkan permainan. *Log in* ini bertujuan untuk mendapatkan nama pengguna yang nantinya akan digunakan dalam proses *matchmaking* ketika bermain. Pemain yang telah melakukan *log in* sebelumnya, tidak perlu melakukan proses *log in* kembali pada halaman ini dan dapat langsung menuju ke halaman menu utama. Hal tersebut dapat terjadi apabila data *log in* yang dilakukan pemain telah terdaftar pada fitur Firebase *authentication*. Implementasi dari hal tersebut terdapat pada *source code login events* yang ditunjukkan oleh gambar 4.5.

Auth	Pick instance with UID Auth.UID	System	Go to MainMenu
Auth	Is login	Add action	

Gambar 4. 5 *Source Code Event* Memeriksa Data *Log In* sebelumnya

Pada gambar 4.5, digunakan objek Firebase *authentication* dengan *condition command* berupa *pick instance* dan *is login* untuk menyatakan kondisi dimana pengguna pernah melakukan *log in* sebelumnya. Pada *condition command* tersebut diberikan *action command* berupa *go to MainMenu* untuk berpindah halaman menu utama.

Apabila pemain belum pernah melakukan *log in* sebelumnya, maka pemain diminta untuk memasukkan namanya di kotak *textbox* yang telah disediakan. Apabila pemain telah selesai memasukkan nama, maka sistem akan melakukan validasi apakah kotak *textbox* tersebut telah terisi atau belum. Apabila telah terisi, maka tombol OK yang berada dibawah kotak *textbox* dapat ditekan oleh pemain. Namun apabila masih kosong, maka tombol OK akan dinonaktifkan. Proses validasi tersebut dapat dilihat pada gambar 4.6 dan 4.7.







Gambar 4. 6 Tampilan ketika *Textbox* Belum Terisi



Gambar 4. 7 Tampilan ketika *Textbox* Telah Terisi

Proses validasi tersebut dibangun dengan *source code* yang terdapat pada *login events*. Pada *source code* tersebut dibutuhkan objek *textbox* bernama

NicknameBox serta objek *system*. Selain itu terdapat *condition command* berupa *compare text* untuk memvalidasi teks yang terdapat pada kotak *textbox* serta *else* untuk menunjukkan kondisi lainnya. Pada *condition command* ketika NicknameBox tidak berisi teks, maka akan ada *action command* berupa *set animation frame* pada objek *button* bernama OKBtn. Pada *action command set animation frame*, diberikan *expression 0* untuk menyatakan tombol sedang dinonaktifkan dan *expression 1* untuk menyatakan tombol sedang diaktifkan. Source code tersebut dapat dilihat pada gambar 4.8.

 NicknameBox	Text is "" (case insensitive)	 OKBtn	Set animation frame to 0
		Add action	
 System	Else	 OKBtn	Set animation frame to 1
		Add action	

Gambar 4. 8 Source Code Memeriksa Nama pada *Textbox*

Apabila tombol OK yang telah aktif tersebut ditekan oleh pemain, maka sistem akan menyimpan nama pemain dan mengalihkan pemain menuju halaman menu utama. Hal tersebut dapat dilihat pada gambar *source code* 4.9. Pada gambar tersebut terdapat objek berupa *mouse*, *button* bernama OKBtn, *textbox* bernama NicknameBox, *firebase authentication* dan *system*. *Condition command* pada *source code* tersebut berupa *on object clicked*, *compare frame*, *compare text*, dan *on login succes*. Apabila pemain berada di *condition command on object clicked* pada objek *mouse*, maka NicknameBox akan melakukan *compare text*. Setelah itu akan dilakukan *action command login anonymous* pada objek *firebase authentication*. Pada *condition command on login success*, *firebase authentication* akan melakukan *action command update display name* sesuai dengan teks pada NicknameBox dan *system* akan melakukan *action command go to MainMenu* untuk berpindah ke halaman menu utama. Namun apabila pemain mendapati *condition command on login error*, maka *system* akan menampilkan layar Alert berisi *debugtext* yang memberitahukan *log in error*.

Mouse	On Left button Clicked on OKBtn	Add action
OKBtn	Animation frame = 1	
NicknameBox	Text is "" (case insensitive)	Auth
		Login with anonymous, persisting type to sessionOnly
		Add action
Auth	On login success	Auth
		Update display name to NicknameBox.Text, photo URL to ""
		System
		Go to MainMenu
		Add action
Auth	On login error	System
		Set layer "Alert" Visible
		NicknameBox
		Set Invisible
		DebugTxt
		Set text to "Log In Error" &newline& "Periksa Konfigurasi Jaringan Anda"

Gambar 4. 9 Source Code Menekan Tombol OK

2. Halaman Menu Utama

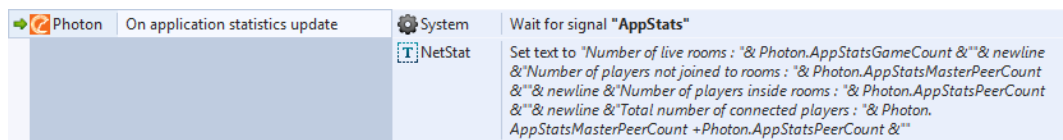
Pembangunan halaman menu utama pada aplikasi permainan *online multiplayer* ular tangga bergenre strategi ini dibuat berdasarkan BAB III sub bab 3.2.4 poin 10 bagian b. Pada halaman menu utama terdapat 3 tombol yaitu tombol bermain untuk menuju ke halaman *preparation game*, tombol panduan bermain untuk menuju ke halaman panduan bermain, dan tombol keluar untuk menutup aplikasi. Selain itu juga terdapat tombol *help* yang digunakan untuk mengetahui data statistik pada sistem *matchmaking* permainan. Tampilan halaman menu utama dapat dilihat pada gambar 4.10.



Gambar 4. 10 Tampilan Halaman Menu Utama

Pada halaman tersebut terdapat *source code* yang digunakan untuk menjalankan fungsi tombol pada halaman menu utama. *Source code* tersebut dapat ditemukan pada *main menu events*. *Source code* tersebut terdiri dari objek *mouse*, *system*, *teks* bernama NetStat dan *photon*. *Condition command* pada

source code tersebut yaitu *on object clicked*, *compare value*, dan *on application statistic update*. Pada *condition command compare value* apabila *global variable State* sama dengan *JoinedLobby*, maka pemain dapat menekan *MenuBtn* tanpa memiliki kondisi tertentu dimana *system* akan melakukan *action command go to* ke halaman yang sesuai dengan variabel *GoToLayout* pada *MenuBtn*. Selain itu pemain dapat melakukan *condition command on object clicked* pada teks *HelpBtn* dimana *system* akan melakukan *action command signal* untuk memberikan sinyal kepada *condition command on application statistic update* untuk dapat melakukan *action command* menampilkan data statistik fitur *multiplayer* pada *NetStat* seperti jumlah *room* yang dapat digunakan untuk berkumpul, jumlah pemain yang berada di dalam *lobby*, jumlah pemain yang berada di dalam *room*, dan jumlah total pemain yang terhubung ke dalam *game server*. *Source code* tersebut ditunjukkan pada gambar 4.11.



Gambar 4. 11 *Source Code* Menampilkan Data Statistik Aplikasi

Apabila pada *condition command compare value global variable State* sama dengan *Error* atau *Disconnected*, maka pemain akan meninggalkan *room lobby*. Pemain yang menekan *MenuBtn Bermain* kemudian *system* akan melakukan *action command set layer visible* pada layar *Alert* dan memberitahukan bahwa terdapat kondisi *error* pada aplikasi. Apabila Pemain yang menekan *MenuBtn selain Bermain* maka *system* akan melakukan *action command go to* ke halaman yang sesuai dengan variabel *GoToLayout* pada *MenuBtn*. Apabila pemain mendapati *condition command* berupa *on object clicked* pada teks *MenuBtn*, apabila pemain menekan *MenuBtn* yang memiliki *text* sama dengan *Keluar* maka *photon* akan melakukan *action command disconnect from all servers* dan

system akan menutup aplikasi. *Source code* tersebut ditunjukkan pada gambar 4.12.

System	State = "JoinedLobby"	Add action	
Mouse	On Left button Clicked on MenuBtn	System	Go to layout ""&MenuBtn.GoToLayout
Mouse	On Left button Clicked on HelpBtn	System	Signal "AppStats"
		System	Wait 7.0 seconds
		NetStat	Set text to ""
		HelpBtn	Set Invisible
		Add action	
System	State = "Error"	Rooms	Leave current room
	- or -	Add action	
System	State = "Disconnected"	System	Set layer "Layer 1" Invisible
Mouse	On Left button Clicked on MenuBtn	System	Set layer "Alert" Visible
System	MenuBtn.Text = "Bermain"	DebugTxt	Set text to "Terjadi Kesalahan Jaringan." &newline& "Coba Lagi ?" &newline& "Err Code : " &Photon.ErrorCode
		Add action	
Mouse	On Left button Clicked on MenuBtn	System	Go to layout ""&MenuBtn.GoToLayout
System	MenuBtn.Text ≠ "Bermain"	Add action	
Mouse	On Left button Clicked on MenuBtn	Photon	Disconnect from all servers
System	MenuBtn.Text = "Keluar"	Electron	Close electron windows
		Add action	

Gambar 4. 12 *Source Code* Tombol pada Halaman Menu Utama

Ketika pertama kali berada di halaman menu utama, pengguna yang telah melakukan *log in* akan melakukan koneksi ke server. Hal tersebut dapat dilihat pada *source code* pada gambar 4.13. *Source code* tersebut terdiri dari objek berupa *system*, *photon*, dan *rooms*. Selain itu terdapat *condition command* berupa *on start layout* dan *compare value*. Pada *condition command on start layout*, objek *photon* akan melakukan *action command connect to name server* untuk menghubungkan ke server Photon. Pada *condition command compare value* variabel *display name* pada Firebase *authentication* yang tidak kosong, *photon* akan melakukan *action command set name local actor* untuk menyimpan nama pemain sesuai dengan variabel *display name* pada Firebase *authentication*. Selain itu *photon* juga akan melakukan *action command set user id* untuk memberikan pemain *user id* dari Firebase *authentication*. Sedangkan objek *rooms* akan melakukan *action command set my info, update*, serta *create* untuk memasukkan pemain ke dalam *lobby* pada fitur Firebase *real-time database*. Apabila *condition command compare value* variabel *display name* pada Firebase *authentication* tersebut kosong, maka *system* akan melakukan

action command go to Login untuk membawa pemain kembali ke halaman *login*.

System	On start of layout	System	Reset global variables to default
		Photon	Connect to the name server
		MenuBtn	Set web font "squirkregular" from "stylesheet.css"
		Add action	
Auth.DisplayName ≠ ""		Photon	Set name of local actor to Auth.DisplayName
		Photon	Set optional user id to Auth.UserID
		Rooms	Set user name to Auth.Username, user ID to Auth.UserID
		Rooms	Start updating "game-block" rooms list
		Rooms	Leave current room, then create or join Persisted "lobby" room: "Lobby", ID: "Lobby", with max peers to 0, door state to Open
		Add action	
Auth.DisplayName = ""		System	Go to Login
		Add action	

Gambar 4. 13 Source Code On start layout pada Halaman Menu Utama

3. Halaman Panduan Bermain

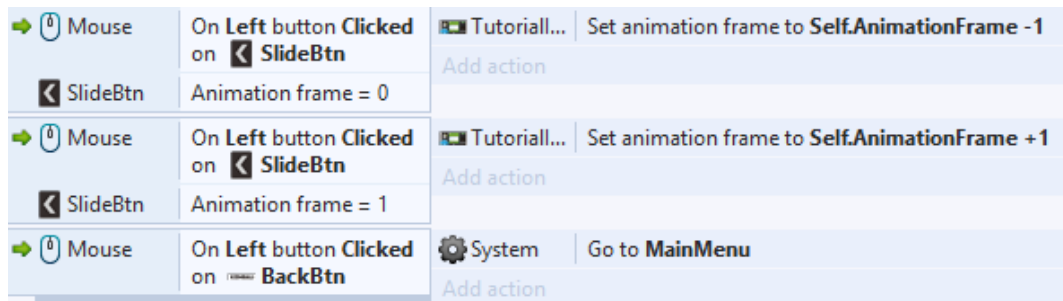


Gambar 4. 14 Tampilan Halaman Panduan Bermain

Pembangunan halaman panduan bermain pada aplikasi permainan *online multiplayer* ular tangga bergenre strategi ini dibuat berdasarkan BAB III sub bab 3.2.4 poin 10 bagian c. Tampilan halaman panduan bermain ditunjukkan pada gambar 4.14. Pada halaman tersebut pemain dapat mengetahui tentang permainan ular tangga bergenre strategi sekaligus cara bermainnya lewat beberapa gambar yang disediakan. Pemain dapat menekan tombol *next* atau *previous* yang berada di samping kanan dan kiri gambar. Tombol tersebut berfungsi untuk memindahkan gambar. Apabila pemain telah selesai

memahami permainan maka pemain dapat kembali ke halaman menu utama dengan menekan tombol kembali.

Halaman panduan bermain memiliki *source code* untuk menjalankan fungsi-fungsi yang terdapat di dalamnya. *Source code* tersebut dapat ditemukan pada *tutorial events*. *Source code* tersebut terdiri dari 3 *condition command*. *Condition command* yang pertama yaitu apabila pemain menekan *mouse* pada objek *SlideBtn* yang memiliki frame 0, maka objek *sprite* bernama *TutorialImage* akan melakukan *action command* berganti gambar ke *frame* sebelumnya. *Condition command* yang kedua yaitu apabila pemain menekan *mouse* pada objek *SlideBtn* yang memiliki frame 1, maka objek *sprite* bernama *TutorialImage* akan melakukan *action command* berganti gambar ke *frame* selanjutnya. *Condition command* yang terakhir yaitu apabila pemain menekan *mouse* pada objek *BackBtn*, maka *system* akan melakukan *action command go to MainMenu* untuk berpindah ke halaman menu utama. *Source code* tersebut dapat dilihat pada gambar 4.15.



Gambar 4. 15 *Source Code* Tombol pada Halaman Panduan Bermain

4. Halaman *Preparation Game*

Pembangunan halaman *preparation game* pada aplikasi permainan *online multiplayer* ular tangga bergenre strategi ini dibuat berdasarkan BAB III sub bab 3.2.4 poin 10 bagian d. Halaman *preparation game* merupakan halaman untuk menentukan urutan bermain. Pada halaman tersebut juga terjadi proses *matchmaking* yang dilakukan oleh Photon *real-time multiplayer*. Untuk melakukan *matchmaking*, pemain yang memasuki halaman *preparation game* dan masih berada di dalam *lobby* akan memasuki sebuah *random room*. Apabila pemain tidak menemukan *room* yang masih terdapat ruang, maka pemain akan

membuat sebuah *room* baru. Hal tersebut dapat dilihat pada gambar *source code* 4.16 yang dapat ditemukan pada *event roll phase*.

Photon	Is connected to a lobby	Photon	Join random room by FillRoom matching my room: Yes in lobby ""/Default with filter ""
		Add action	
Photon	On joinRandomRoom no match found	Photon	Create room "Block "& int(random(10)) in lobby ""/Default
		Add action	

Gambar 4.16 *Source Code* Membuat atau Memasuki *Room*

Pada *source code* tersebut ditunjukkan dua *event* yang terdapat pada halaman *preparation game*. *Event* yang pertama berisi *condition command is connected to a lobby*. Apabila pemain mendapati *condition command* tersebut, maka pemain akan melakukan *action command join random room* dimana Photon Cloud akan memasukkan pemain ke dalam *random room* yang masih memiliki ruang untuk diisi. *Event* yang kedua berisi *condition command on joinRandomRoom no match found*. Apabila pemain mendapati *condition command* tersebut, maka pemain akan melakukan *action command create room* untuk membuat *room* baru dan masuk ke dalamnya.

Setelah membuat atau memasuki sebuah *room*, maka pemain akan memasuki dan berada di dalam *room*. Pada *source code* yang ditunjukkan pada gambar 4.17, dijelaskan *event* mengenai pemain yang memasuki *room*. Pada *event* tersebut terdiri dari objek *photon*, *system*, *rooms*, *text* bernama *PlayerNameTxt*, *progress bar*, dan *button* bernama *CancelBtn*. Pada *source code* tersebut juga terdapat *condition command* berupa *on join room*, *on actor join*, *pick instance*, *compare value*, *compare text*, *room is full*, dan *else*. Pada *condition command on join room*, sistem akan melakukan *action command set value* untuk merubah *global variable ROOM_NAME* menjadi nama *room* yang dicatat pada Photon Cloud. Selain itu *rooms* akan melakukan *action command create* untuk membuat atau masuk ke *room* yang sesuai dengan *global variable ROOM_NAME*. *Action command* lainnya yaitu *set suspended player live time* dan *set room live time* yang dilakukan oleh *plugin Photon*. Pada *source code* ini juga terdapat *event* dengan *condition command room is full*, dimana *CancelBtn* akan melakukan *action command set animation frame* untuk merubah *frame* menjadi 0. *Frame* tersebut menandakan bahwa tombol tersebut dinonaktifkan.

Action command selanjutnya yaitu *set status room* dan *close room* yang berstatus *no* dan *close* yang menyatakan bahwa pemain lain tidak dapat memasuki ke room tersebut serta *progress bar* yang akan berubah menjadi 65. Apabila pemain menemui *condition command else*, maka *CancelBtn* akan melakukan *action command set animation frame* menjadi 1 untuk merubah *frame* menjadi 1 dan menunggu selama 30 detik. Pemain yang menemui *condition command On actor Join* akan mendapatkan *action command set text* pada *PlayerNameTxt* dan *action command set animation frame* pada objek *Pawn*. Hasil dan *source code* pada gambar 4.18 dan 4.19.

Photon	On join room	System	Set ROOM_NAME to Photon.MyRoomName
		Rooms	Leave current room, then create or join Temporary "game-block" room: ROOM_NAME, ID: ROOM_NAME, with max peers to 2, door state to Open
		Photon	Set suspended player live time to 30000
		Photon	Set room live time to 500
		Add action	
Rooms	Room is full	CancelBtn	Set animation frame to 0
		Photon	Set room open status to No
		Rooms	Close current room
		ProgressBar	Set progress to 65
		Add action	
System	Else	System	Wait 30.0 seconds
		CancelBtn	Set animation frame to 1

Gambar 4. 17 Source Code On Join Room

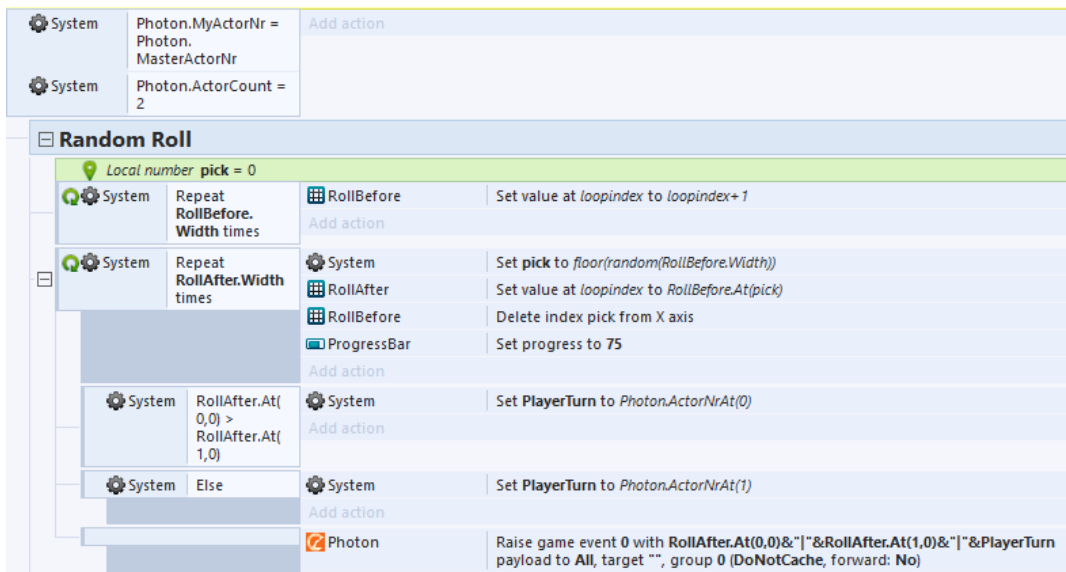
Photon	On actor join	Add action	
System	Photon.MyActorNr = Photon.ActorNr	Add action	
PlayerNam...	Pick instance with UID 53	PlayerNameTxt	Set text to Photon.ActorNameByNr(Photon.MasterActorNr)
		Add action	
Pawn	Pick instance with UID 57	Pawn	Set animation frame to Photon.MasterActorNr+2
		Pawn	Set Visible
		ProgressBar	Set progress to 25
		Add action	
System	Photon.MyActorNr ≠ Photon.ActorNr	Add action	
	- or -		
System	Photon.ActorNr ≠ Photon.MasterActorNr		
PlayerNam...	Pick instance with UID 54	PlayerNameTxt	Set text to Photon.ActorNameByNr(Photon.ActorNr)
		Add action	
Pawn	Pick instance with UID 58	Pawn	Set animation frame to Photon.ActorNr+2
		Pawn	Set Visible
		ProgressBar	Set progress to 50

Gambar 4. 18 Source Code On Actor Join

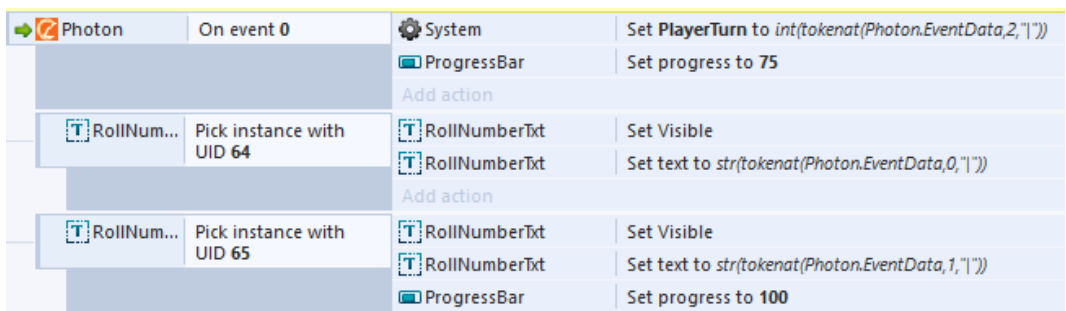


Gambar 4. 19 Tampilan Ketika Pemain Memasuki *Preparation Game*

Ketika pemain memasuki sebuah *room*, pemain yang masuk pertama kali akan menjadi *master actor* atau dapat disebut dengan *host*. Sedangkan pemain yang masuk kedalam sebuah *room* yang terdapat *host* di dalamnya, maka pemain tersebut dapat disebut *actor* atau dapat disebut dengan *peer*. Ketika sebuah *room* telah berisi 1 *host* dan 1 *peer*, maka akan dilakukan tahap *preparation game* atau pengundian urutan bermain. Penjelasan pengundian urutan bermain dapat dilihat pada *source code* di gambar 4.20. Pada *source code* tersebut terdapat objek *system*, *photon*, *text* yang bernama *RollNumberTxt*, *progress bar*, dan *array* bernama *RollBefore* dan *RollAfter*. Selain itu *source code* tersebut juga berisi *condition command repeat*, *compare text*, *else*, *on event*, dan *pick instance*. Pada *condition command repeat* ukuran dari *array* *RollBefore*, *RollBefore* akan melakukan *action command set value*. Pada *condition command repeat* ukuran dari *array* *RollAfter*, *system* dan *RollAfter* akan melakukan *action command set value*. Kemudian *RollBefore* akan melakukan *action command delete* untuk menghapus nilai *index* pada *array* dan *progress bar* akan merubah nilai progresnya menjadi 65. *System* kemudian akan melakukan *action command set value* pada *global variable* *PlayerTurn* sesuai dengan nilai *x* pada *array* *RollAfter*. *Action command* selanjutnya yaitu melakukan *action command raise event* untuk mengirimkan data nilai *x* pada *RollAfter* dan nilai *PlayerTurn* ke semua tipe aktor.



Gambar 4. 20 Source Code Random Roll pada Tahap Prepration Game



Gambar 4. 21 Source Code Raise Event Roll Dadu



Gambar 4. 22 Tampilan Pengundian Giliran Bermain pada Preparation Game

Pada gambar 4.21 dan 4.22 ditunjukkan apabila aktor mendapat *condition command raise event* maka *system* akan melakukan *action command set value* pada *global variable PlayerTurn* sesuai dengan data pada *raise event* dan nilai

progres pada *progress bar* menjadi 75. Selain itu teks pada *RollNumberTxt* akan berubah menjadi nilai undian yang terdapat pada data *raise event*.

Apabila pemain mendapati *condition command* dimana value pada *progress bar* mencapai 100, maka pemain akan menunggu selama 1,5 menit untuk kemudian berpindah ke halaman *in game*. *Source code* tersebut ditunjukkan pada gambar 4.23.

ProgressBar	Progress = 100	System	Wait 1.5 seconds
		System	Go to Game

Gambar 4. 23 *Source Code Progress Bar* Mencapai 100

Apabila pemain tidak menemukan lawan bermain, pemain dapat menekan tombol batal untuk membatalkan permainan. Proses tersebut dapat dilihat pada *source code* dan hasil tampilan yang ditunjukkan gambar 4.24 dan 4.25. *Source code* tersebut menunjukkan *condition command on object clicked* di *CancelBtn* dengan nilai *animation frame* berupa 1. Pada *condition command* tersebut, *photon* dan *rooms* akan melakukan *action command leave room* dan *system* akan melakukan *action command go to MainMenu* yang menunjukkan bahwa pemain akan berpindah ke halaman menu utama.

Mouse	On Left button Clicked on CancelBtn	System	Wait 1 seconds
CancelBtn	Animation frame = 1	Photon	Leave room
		Rooms	Leave current room
		System	Go to MainMenu

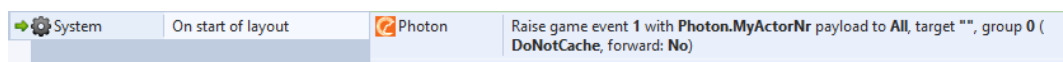
Gambar 4. 24 *Source Code Tombol Batal*



Gambar 4. 25 Tampilan Ketika Tombol Batal Aktif

5. Halaman *In Game*

Pembangunan halaman *in game* pada aplikasi permainan *online multiplayer* ular tangga bergenre strategi ini dibuat berdasarkan BAB III sub bab 3.2.4 poin 10 bagian e. Pada halaman *in game*, pemain dapat memainkan permainan ular tangga bergenre strategi. Ketika kedua pemain memasuki halaman *in game*, maka sistem akan terlebih dahulu melakukan sinkronisasi terhadap dua pemain yang berada dalam permainan. Hal tersebut dapat dilihat pada gambar 4.26 dan 4.27.




Gambar 4. 26 *Source Code On Start Layout* Halaman *In Game*

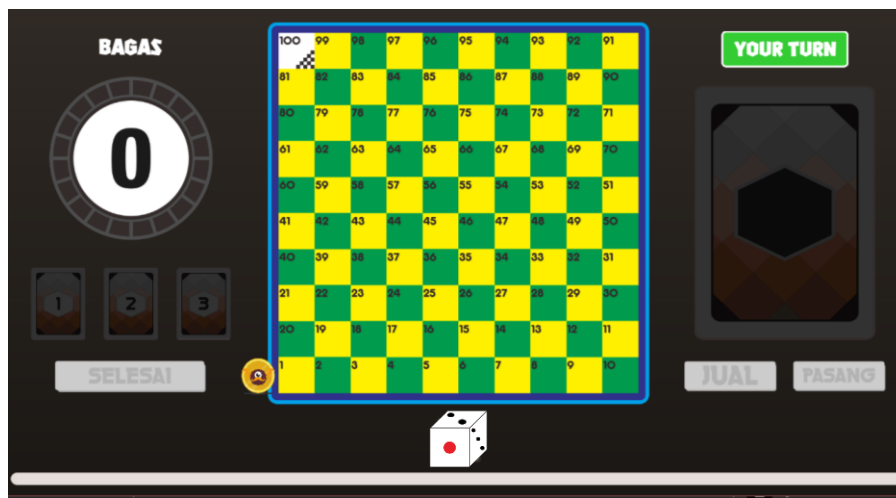


Gambar 4. 27 *Source Code On Actor Join* Halaman *In Game*

Gambar 4.26 dan 4.27 menunjukkan *source code* dimana masing-masing pemain akan mengirimkan data berupa nomor aktor dalam permainan. Data tersebut akan digunakan untuk membuat pion pada papan permainan. Pada *source code* tersebut objek *photon* akan melakukan *action command raise event* untuk mengirimkan data berupa nomor aktor ke pemain lainnya. Apabila pemain menerima data tersebut, maka sistem akan membuat pion pada papan permainan. Hal tersebut dapat dilihat pada *source code* dan tampilan antarmuka yang ditunjukkan gambar 4.28 dan 4.29. *Source code* tersebut terdiri dari objek *photon*, pion bernama *Pawn*, dan *point bar*. Pada *source code* tersebut terdapat *condition command on event*. Pada *condition command* tersebut sistem akan membuat *Pawn* pada kotak *start* yang berada di papan permainan. Objek *Pawn* yang terbentuk kemudian akan melakukan *action command* seperti *set animation frame*, *set size*, *visible*, dan *set value* pada variabel *number_player*, *x_position*, *y_position*, dan *player_id*. Pada *condition command* tersebut, sistem juga akan mengatur objek *pointbar* dengan melakukan *action command set value* pada variabel *owner_id*.


Photon	On event 1	Board	Create chess  Pawn to [0, 9, int(Photon.EventData)] on layer "Player"
		Pawn	Set Visible
		Pawn	Set size to (60, 64)
		Pawn	Set number_player to int(Photon.EventData)
		Pawn	Set x_position to Self.Chess.LX
		Pawn	Set y_position to Self.Chess.LY
		Pawn	Set animation frame to Self.number_player + 2
		PointBar	Set owner_id to str(Pawn.number_player)
		Pawn	Set player_id to "&Photon.ActorNameByNr(Self.number_player)

Gambar 4. 28 Source Code Sinkronisasi Pembuatan Pion dan Point Bar



Gambar 4. 29 Tampilan Hasil Sinkronisasi Pembuatan Pion dan Point Bar

Ketika pemain melakukan *turn phase*, pion yang dimiliki pemain akan bergerak sesuai kotak yang terdapat pada papan permainan. Untuk mengatur sinkronisasi pion antar setiap pemain, dibutuhkan sebuah fungsi. Fungsi tersebut dapat dilihat pada gambar *source code* 4.30. Pada *source code* tersebut terdapat *action command raise event* yang akan mengirim data berupa variabel `number_player`, `x_position`, `y_position`, dan `move` pada objek Pawn ke semua pemain.

	Raise game event 2 with Pawn.number_player &" "& Pawn.x_position &" "& Pawn.y_position &" "& Pawn.move payload to All, target "", group 0 (DoNotCache, forward: No)
---	---

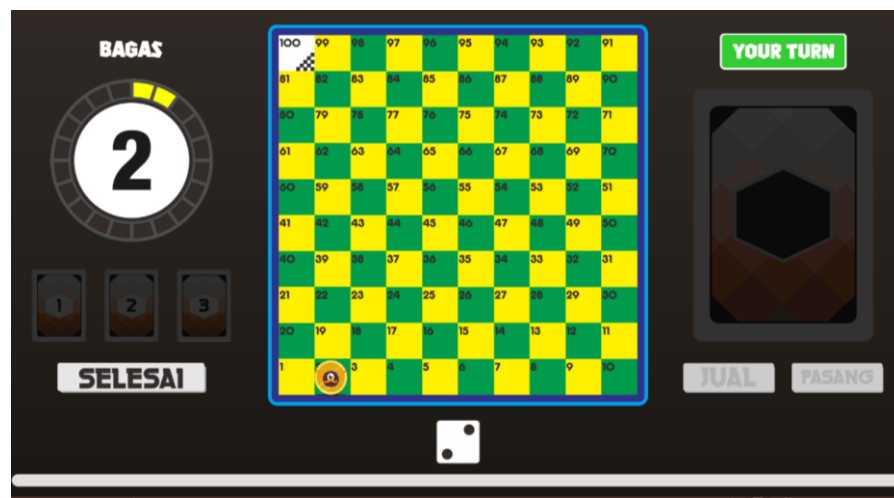
Gambar 4. 30 Raise Event Sinkronisasi Pion

Apabila pemain mendapati *condition command on event* dari *source code* pada gambar 4.30, maka akan dilakukan *action command* kepada objek pawn berupa *move to* pada *behavior grid* serta *set value* pada variabel `x_position`,

$y_position$, dan $move$. *Source code* dan tampilan dari *condition command on event 2* dapat dilihat pada gambar 4.31 dan 4.32.

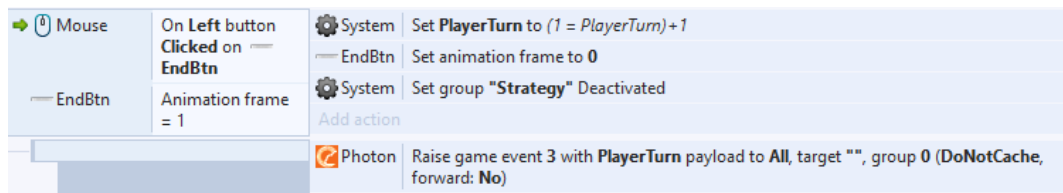


Gambar 4. 31 *Source Code* Sinkronisasi Pion



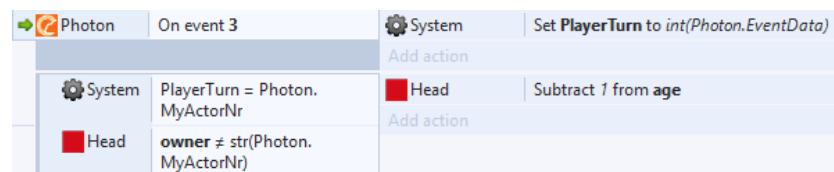
Gambar 4. 32 Tampilan Pion Berjalan

Ketika pemain menekan tombol selesai, maka giliran bermain akan berganti ke pemain lainnya. Untuk memberitahu bahwa giliran bermain telah berganti, maka sistem akan melakukan sinkronisasi antar pemain. Hal tersebut dapat dilihat pada gambar *source code* 4.33. Pada *source code* tersebut terdapat *condition command on object clicked* pada objek `EndBtn` yang memiliki *animation frame* sama dengan 1. *Condition command* tersebut akan melakukan *action command set value* pada *global variable* `PlayerTurn`, *set group active* pada grup *strategy* dan *set animation frame* pada objek `EndBtn`. Setelah melakukan hal tersebut, maka *photon* akan melakukan *action command raise event* dengan membawa data *global variable* `PlayerTurn` untuk dikirimkan ke semua pemain.



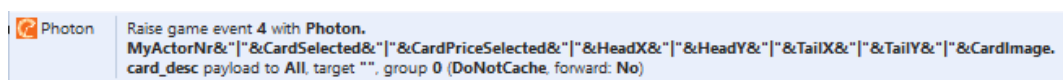
Gambar 4. 33 Raise Event End Phase

Apabila pemain mendapat *condition command On event* dari *source code* 4.33 maka akan dilakukan *action command set value* pada *global variable* PlayerTurn. Pada *condition command On event*, terdapat juga *condition command* lainnya berupa *compare text* dan *compare value* pada variabel owner objek Head. Pada *condition command* tersebut, akan dilakukan *action command subtract* pada variabel age objek Head. *Source code* tersebut dapat dilihat pada gambar 4.34.



Gambar 4. 34 Source Code Sinkronisasi Pergantian Giliran Bermain

Ketika pemain melakukan pemasangan *item* pada papan permainan, maka pemain lain juga akan mengetahui juga letak *item* pemain lain pada layar perangkatnya. Untuk melakukan hal tersebut, maka sistem akan melakukan sinkronisasi *item* ke antar pemain. Hal tersebut dapat dilihat pada gambar *source code* 4.35. Pada *source code* tersebut terdapat *condition command compare value* pada *local variable* PairItemCheck. Pada *condition command* tersebut, akan dilakukan *action command raise event* dengan mengirimkan data ke semua pemain berupa variabel nomor aktor pada Photon *real-time multiplayer*, *global variable* berupa CardSelected, CardPriceSelected, HeadX, HeadY, TailX, dan TailY, serta variabel card_desc pada objek CardImage.



Gambar 4. 35 Raise Event Sinkronisasi Item

Apabila pemain mendapat *condition command On event* dari *source code* 4.35 maka akan dilakukan *action command create chess* objek Head dan Tail

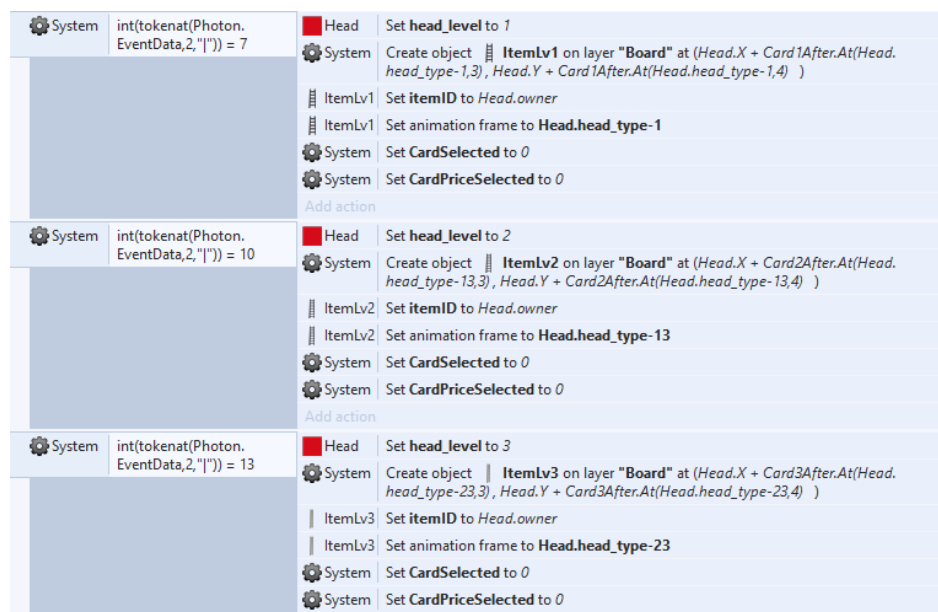
sesuai data yang telah didapatkan. Setelah membuat sebuah objek Head dan Tail pada papan permainan, akan dieksekusi beberapa *action command* selanjutnya berupa *set value* pada variabel owner, age, head_desc, dan head_type di objek Head. Setelah itu setiap pemain memiliki Head dan Tail yang berbeda-beda dengan menjalankan *action command set animation frame* pada objek Head dan Tail. *Source code* tersebut ditunjukkan pada gambar 4.36.

Photon	On event 4	Board	Code
		Board	Create chess ■ Head to [int(tokenat(Photon.EventData,3,"")), int(tokenat(Photon.EventData,4,"")), 5] on layer "Board"
		Board	Create chess ■ Tail to [int(tokenat(Photon.EventData,5,"")), int(tokenat(Photon.EventData,6,"")), 5] on layer "Board"
		Head	Set owner to str(tokenat(Photon.EventData,0,""))
		Head	Set age to 4
		Head	Set head_desc to str(tokenat(Photon.EventData,7,""))
		Head	Set head_type to int(tokenat(Photon.EventData,1,""))
		Head	Set animation frame to (int(Head.owner))-1
		Tail	Set animation frame to (int(Head.owner))-1

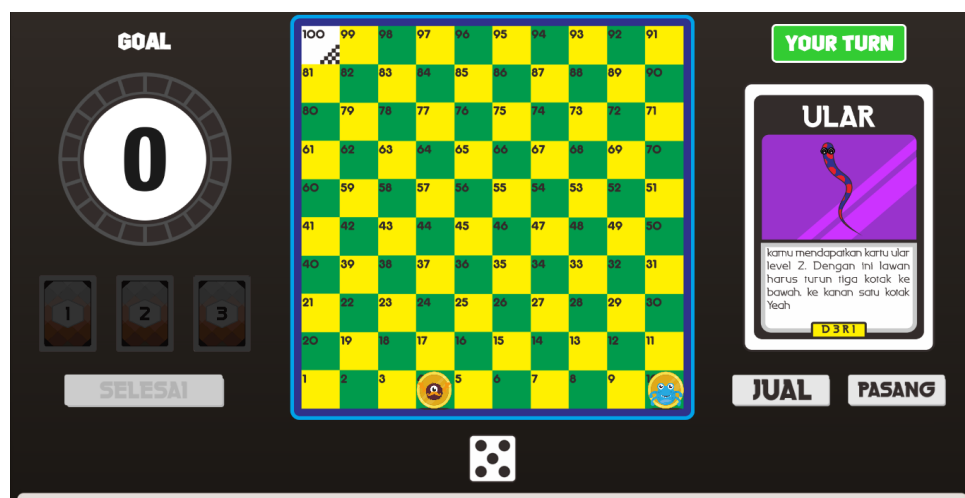
Gambar 4. 36 Source Code Sinkronisasi Item

Setelah meletakkan objek Head dan Tail pada papan permainan, maka sistem akan membuat *item* sesuai dengan gambar pada kartu yang didapat. Hal tersebut dapat dilihat pada *source code* pada gambar 4.37. Pada gambar tersebut terdapat 3 *conditional command compare text* dari *system* yang berfungsi untuk menyatakan bahwa nilai pada *event data* sama dengan harga kartu yang didapat. Apabila nilai pada *event data* sama dengan 7 maka akan dilakukan *action command set value* 1 pada variabel head_level objek Head. Kemudian *system* akan membuat objek ItemLv1 pada papan permainan sesuai dengan *event data*. *Item* yang telah terbentuk akan melakukan *action command set value* pada variabel itemid dan *set animation frame* objek ItemLv1. Yang terakhir, *system* akan melakukan *action command* pada *global variable* CardSelected dan CardPriceSelected. Hal yang sama juga terjadi apabila nilai pada *event data* sama dengan 10, maka akan dilakukan *action command set value* 2 pada variabel head_level objek Head. Kemudian *system* akan membuat objek ItemLv2 pada papan permainan sesuai dengan *event data*. *Item* yang telah terbentuk akan melakukan *action command set value* pada variabel itemid dan *set animation frame* objek ItemLv2. Yang terakhir, *system* akan melakukan *action command* pada *global variable* CardSelected dan CardPriceSelected.

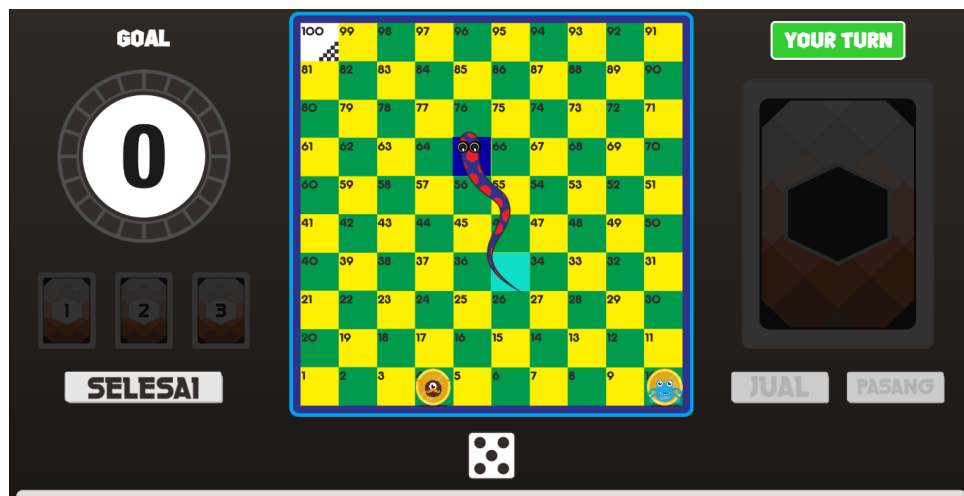
Begitu juga bila nilai pada *event data* sama dengan 13 maka akan dilakukan *action command set value* 3 pada variabel *head_level* objek *Head*. Kemudian *system* akan membuat objek *ItemLv3* pada papan permainan sesuai dengan *event data*. *Item* yang telah terbentuk akan melakukan *action command set value* pada variabel *itemid* dan *set animation frame* objek *ItemLv3*. Yang terakhir, *system* akan melakukan *action command* pada *global variable* *CardSelected* dan *CardPriceSelected*. Hasil tampilan dari *source code* gambar 4.37 dapat dilihat pada gambar 4.38 dan 4.39.



Gambar 4. 37 Source Code Pembuatan Item Sesuai Level



Gambar 4. 38 Tampilan Ketika Pemain Membeli Kartu



Gambar 4. 39 Tampilan Ketika Pemain Memasang Item Kartu

Ketika pemain lain mendapatkan giliran bermain, maka pemain akan menunggu pemain lain hingga melakukan *end*. Disaat menunggu tersebut pemain juga dapat melihat poin lawan dari notifikasi yang ditampilkan. Hal tersebut dapat dilihat pada gambar *source code* 4.40. Pada *source code* tersebut terdapat *condition command On object clicked* pada objek kartu. Ketika pemain menekan objek tersebut, maka *photon* akan melakukan *raise event* data variabel *point* pada objek Pawn untuk dikirimkan ke pemain lain.

Mouse	On Left button Clicked on CardLv1	System	Set CardPriceSelected to CardLv1.price
CardLv1	Animation frame = 0	Pawn	Subtract CardLv1.price from point
CardLv1	Animation frame = 0	System	Set CardSelected to (int(random(12))) + 1
		CardLv1	Set card_desc to Card1After.At(CardSelected-1,0)
		CardLv1	Set effect_onX to Card1After.At(CardSelected-1,1)
		CardLv1	Set effect_onY to Card1After.At(CardSelected-1,2)
		Transa...	Set animation frame to 1
		PairBtn	Set animation frame to 1
		Photon	Raise game event 6 with Pawn.point payload to Others, target "", group 0 (DoNotCache, forward: No)

Gambar 4. 40 Source Code Update Poin Lawan

Apabila pemain lain mendapatkan *condition command On event 6*, maka objek *function* akan melakukan *action command call function* Message untuk memberitahu poin lawan. Hal tersebut dapat dilihat pada gambar *source code* 4.41.

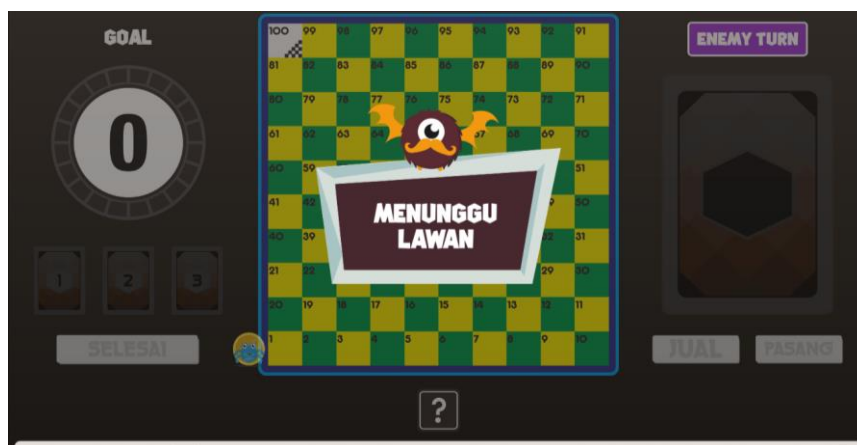
Photon	On event 6	Add action
int(tokenat[Photon.EventData,1,""]) = 7	Function	Call "Message" ("Lawan Membeli Kartu Level 1" & newline& "Poin Lawan Menjadi " & int(tokenat[Photon.EventData,0,""])& " poin")
int(tokenat[Photon.EventData,1,""]) = 10	Function	Call "Message" ("Lawan Membeli Kartu Level 2" & newline& "Poin Lawan Menjadi " & int(tokenat[Photon.EventData,0,""])& " poin")
int(tokenat[Photon.EventData,1,""]) = 13	Function	Call "Message" ("Lawan Membeli Kartu Level 3" & newline& "Poin Lawan Menjadi " & int(tokenat[Photon.EventData,0,""])& " poin")

Gambar 4. 41 Source Code Memberitahu Poin Lawan

Untuk *action command* yang dilakukan oleh objek *function* dapat dilihat pada gambar *source code* 4.42. Pada gambar tersebut terdapat 2 *function* berupa Message dan Info. Pada *condition command function* On Message, maka *system* akan melakukan *action command set visible layer* pada layar Notification kemudian melakukan *action command set text* pada objek DebugTxt. Sedangkan pada *condition command function* On Info, maka *system* akan melakukan *action command set visible layer* pada layar Info kemudian melakukan *action command set animation frame* pada objek SmallPopUp. Salah satu hasil *function* tersebut dapat dilihat pada gambar 4.43.

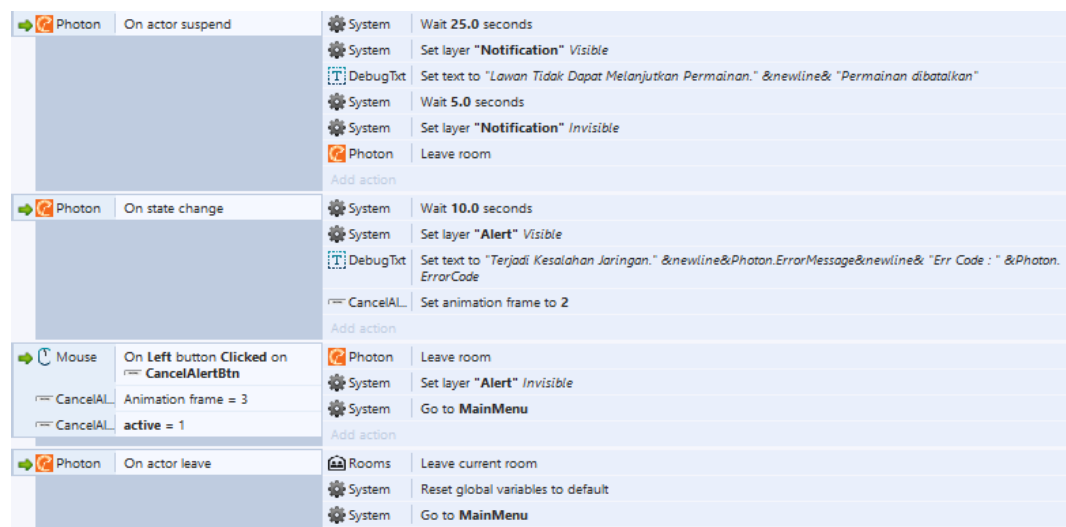
Function	On "Message"	System	Set layer "Notification" Visible
		Debug...	Set text to Function.Param(0)
		System	Wait 1.5 seconds
		System	Set layer "Notification" Invisible
		Add action	
Function	On "Info"	System	Set layer "Info" Visible
		Small...	Set animation frame to Function.Param(0)
		System	Wait 1.0 seconds
		System	Set layer "Info" Invisible

Gambar 4. 42 Source Code Objek Function



Gambar 4. 43 Tampilan Function Info

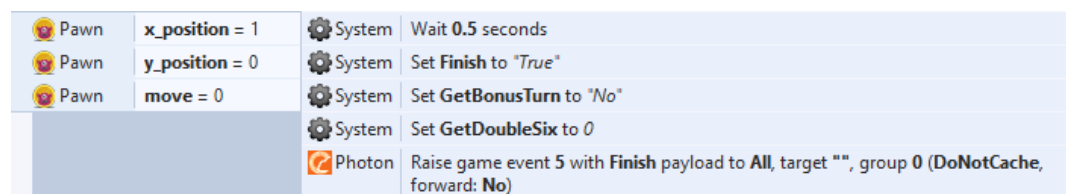
Ketika melakukan permainan, pemain dapat saja keluar dari *room* apabila mengalami sebuah gangguan yang disengaja maupun tidak disengaja. Untuk meninggalkan *room*, terdapat *condition command* *On actor suspend*, *On actor leave*, dan *On button clicked*. Apabila pemain mendapati *condition command* tersebut maka akan dilakukan *action command* *leave room* pada objek *rooms* dan *reset global variables* dan *go to MainMenu*. Source code tersebut dapat dilihat pada gambar 4.44.



Gambar 4. 44 Source Code Meninggalkan Permainan

6. Halaman Hasil Akhir

Pembangunan halaman hasil akhir pada aplikasi permainan *online multiplayer* ular tangga bergenre strategi ini dibuat berdasarkan bab 3 sub bab 3.2.4 poin 10 bagian f. Halaman ini menampilkan hasil akhir permainan apabila salah satu pemain telah mencapai kotak *finish*. Hal tersebut dapat ditunjukkan oleh gambar *source code* 4.45.

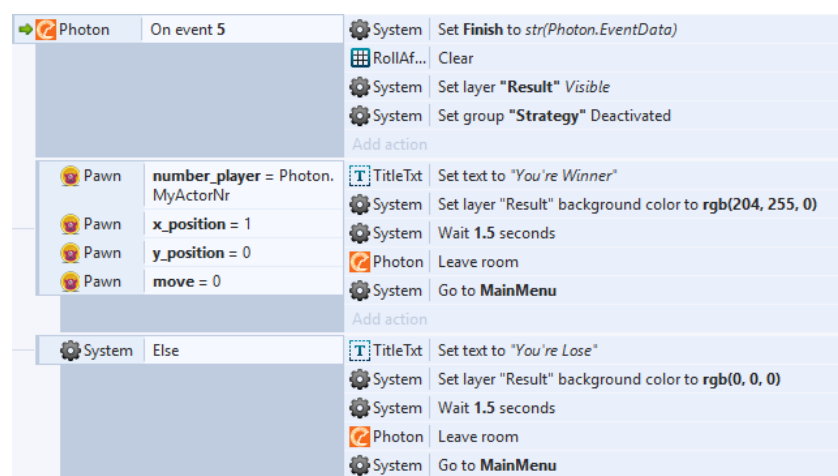


Gambar 4. 45 Source Code Berada di Kotak Finish

Pada *source code* tersebut ditunjukkan *condition command* apabila objek Pawn memiliki variabel *x_position* sama dengan 1, *y_position* sama dengan 0

dan move sama dengan 0. Pada *condition command* tersebut, *system* akan menjalankan *action command set value* pada *global variable* Finish, GetBonusTurn dan GetDoubleSix. Bagian terakhir akan dilakukan oleh *photon* dengan mengirimkan *action command raise event* data *global variable* Finish ke semua pemain.

Ketika pemain mendapatkan *condition command On event 5*, maka *system* akan melakukan *action command set value* pada *global variable* Finish, *set visible* pada layar Result, dan *set active group* pada untuk menonaktifkan grup *strategy*. Selain itu, pada *condition command On event 5* terdapat juga *condition command* dimana pemain akan mendapatkan tampilan hasil akhir permainan. Bagi pemain yang memenangkan permainan, maka objek Pawn akan mendapati *condition command compare value* pada variabel *number_player*, *x_position*, *y_position*, dan *move*. Ketika mendapati *condition command* tersebut, maka akan terdapat *action command set text* “You’re Winner” pada objek TitleTxt, *set background color* dan *go to* pada objek System, serta *leave room* pada objek *photon*. Sedangkan bagi pemain yang kalah, maka objek Pawn akan mendapati *condition command else*. Ketika mendapati *condition command* tersebut, maka akan terdapat *action command set text* “You’re Lose” pada objek TitleTxt, *set background color* dan *go to* pada objek System, serta *leave room* pada objek *photon*. *Source code* tersebut beserta tampilannya dapat dilihat pada gambar 4.46 dan 4.47.



Gambar 4. 46 Source Code Hasi Akhir Permainan



Gambar 4. 47 Tampilan Halaman Hasil Akhir Permainan

4.2. Testing

Untuk memastikan permainan digital *online multiplayer* ular tangga bergenre strategi dapat berjalan sesuai dengan yang diharapkan, maka penulis melakukan empat tahap pengujian pada aplikasi tersebut. Pengujian yang pertama yaitu pengujian konektifitas *backend*. Pengujian kedua yaitu pengujian *gameplay* permainan. Pengujian ketiga yaitu pengujian performa. Pengujian yang terakhir yaitu pengujian portabilitas.

4.2.1. Pengujian Konektifitas *Backend*

Pengujian konektifitas *backend* adalah sebuah pengujian untuk mengetahui kinerja layanan *backend* dalam mengelola fitur *multiplayer* pada permainan, baik ketika terhubung dengan jaringan maupun tidak.

a. Tujuan Pengujian

Pengujian konektifitas *backend* tersebut bertujuan untuk memastikan apakah layanan *backend* yang bertugas untuk mengelola fitur *multiplayer* pada permainan dapat melakukan sesuai dengan fungsinya atau tidak.

b. Lingkungan Pengujian

Dalam pengujiannya, penulis menggunakan dua perangkat PC atau Laptop. Berikut spesifikasi dari dua perangkat PC atau Laptop yang digunakan.

1) PC atau Laptop A

Perangkat yang digunakan adalah Lenovo Z40-45 dengan spesifikasi,

- *Processor* : AMD A8-6410 APU 2.00 GHz
- *Operating System (OS)* : Windows 10 Pro 64-bit
- *Installed Memory (RAM)* : 4,00 GB
- *Memory (Hardisk)* : 500 GB

2) PC atau Laptop B

Perangkat yang digunakan adalah Asus XAS20 dengan spesifikasi,

- *Processor* : Intel Core i3-3217U 1.80 GHz
- *Operating System (OS)* : Windows 10 Education 64-bit
- *Installed Memory (RAM)* : 4,00 GB
- *Memory (Hardisk)* : 500 GB

c. Prosedur Pengujian

Dalam melakukan pengujian konektivitas *backend* pada aplikasi, penulis menjalankan permainan *online multiplayer* ular tangga bergenre strategi dan mengamati *server* layanan *backend* yang diakses melalui *browser* secara bersamaan. Hal tersebut dilakukan untuk memastikan apakah fitur *backend* yang berjalan pada aplikasi telah sesuai dengan data yang ditampilkan pada *server* layanan *backend* atau tidak. Fitur *backend* yang diuji pada aplikasi ini yaitu fitur *real-time multiplayer* pada Photon Cloud serta fitur *authentication* dan *real-time database* pada Firebase. Pengujian konektivitas *backend* ini dilakukan dengan menggunakan alamat IP lokal 192.168.43.37 dan *port* 50000 untuk menghubungkan perangkat PC atau Laptop dengan permainan digital *online multiplayer* ular tangga bergenre strategi. Setiap skenario pengujian dari masing-masing fitur layanan *backend* akan dilakukan sebanyak 3 kali pengujian.

d. Hasil dan Pembahasan

Hasil dari pengujian konektivitas *backend* dapat dilihat pada tabel 4.1, 4.2, dan 4.3. Berikut hasil dan pembahasan dari pengujian konektivitas *backend* yang telah dilakukan.

Tabel 4. 1 Hasil Pengujian Fitur *Firestore Authentication*

Firestore Authentication				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
1	Pemain memasukkan nama dan menekan tombol OK yang aktif	Firestore menyimpan data nama dan memberikan ID Otentifikasi	Firestore <i>authentication</i> menyimpan nama dan memberikan ID Otentifikasi	Berhasil
2	Pemain membuka aplikasi yang telah terdaftar <i>login</i> sebelumnya	Firestore membaca <i>session</i> pada aplikasi dan langsung menuju halaman menu utama	Pemain langsung menuju halaman menu utama	Berhasil

Tabel 4. 2 Hasil Pengujian Fitur *Firestore Real-time Database*

Firestore Real-time Database				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
1	Pemain berada halaman main menu	Firestore memasukkan pemain ke dalam <i>lobby</i>	Firestore <i>real-time database</i> menampilkan pemain yang telah masuk ke dalam <i>lobby</i>	Berhasil
2	Pemain menekan tombol bermain	Pemain meninggalkan <i>lobby</i> dan membuat <i>room</i> baru atau memasuki <i>room</i> yang telah tersedia	Firestore <i>real-time database</i> menghapus data pemain yang terdapat pada <i>lobby</i> dan membuat <i>room</i> baru atau memasuki <i>room</i> yang telah tersedia	Berhasil

Firestore Real-time Database				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
3	Pemain menekan tombol batal	Pemain meninggalkan <i>room</i> dan kembali ke <i>lobby</i>	Firestore <i>real-time database</i> menghapus data pemain yang terdapat pada <i>room</i> dan memasukkan pemain ke dalam <i>lobby</i>	Berhasil
4	Pemain menyelesaikan permainan dan melihat hasil akhir permainan	Pemain meninggalkan <i>room</i> dan kembali ke <i>lobby</i>	Firestore <i>real-time database</i> menghapus data pemain yang terdapat pada <i>room</i> dan memasukkan pemain ke dalam <i>lobby</i>	Berhasil
5	Pemain kehilangan koneksi internet	Data pemain tidak muncul pada Firestore <i>real-time database</i>	Data pemain pada Firestore <i>real-time database</i> tidak muncul	Berhasil

Tabel 4. 3 Hasil Pengujian Fitur *Photon Real-time Multiplayer*

Photon Real-time Multiplayer				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
1	Pemain membuka aplikasi dan koneksi stabil	Data statistik CCU pada <i>photon server</i> menjadi 2, dan pemain masuk ke dalam <i>lobby</i>	Data statistik CCU pada <i>photon server</i> menjadi 2, dan pemain masuk ke dalam <i>lobby</i>	Berhasil
2	Pemain kehilangan koneksi internet	Pemain memutuskan koneksi dengan <i>photon server</i>	Pemain tidak terhubung ke <i>photon server</i>	Berhasil

Photon Real-time Multiplayer				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
3	Pemain menekan tombol bantuan	Sistem menampilkan jumlah <i>room</i> yang terdapat di dalam permainan, jumlah <i>client</i> yang tidak berada di dalam <i>room</i> , jumlah <i>client</i> yang berada di dalam <i>room</i> , dan jumlah total <i>client</i> yang terhubung dengan permainan	Halaman menu utama menampilkan jumlah <i>room</i> yang terdapat di dalam permainan, jumlah <i>client</i> yang tidak berada di dalam <i>room</i> , jumlah <i>client</i> yang berada di dalam <i>room</i> , dan jumlah total <i>client</i> yang terhubung dengan permainan	Berhasil
4	Pemain menekan tombol bermain	Pemain meninggalkan <i>lobby</i> dan membuat <i>room</i> baru atau memasuki <i>room</i> yang telah tersedia. Data statistik <i>Rooms</i> menjadi 1.	Data statistik <i>Rooms</i> pada photon server menjadi 1 yang menandakan pemain berada pada <i>room</i> dan meninggalkan <i>lobby</i>	Berhasil
5	Pemain menekan tombol batal	Pemain meninggalkan <i>room</i> dan kembali ke dalam <i>lobby</i> . Data statistik <i>Rooms</i> menjadi 0	Data statistik <i>Rooms</i> pada photon server menjadi 0 yang menandakan pemain berada pada <i>lobby</i> dan meninggalkan <i>room</i>	Berhasil
6	Pemain kehilangan koneksi internet saat berada di halaman <i>in game</i>	Sistem memberikan notifikasi menunggu untuk pemain yang masih berada di <i>room</i>	Terdapat notifikasi menunggu untuk pemain yang masih berada di <i>room</i>	Berhasil
7	Pemain tidak kembali ke dalam <i>room</i> yang sama selama 30 detik	Pemain yang masih berada di <i>room</i> meninggalkan permainan. Data statistik <i>Rooms</i> menjadi 0	Data statistik <i>Rooms</i> pada photon server menjadi 0 yang menandakan pemain meninggalkan <i>room</i>	Berhasil

Photon Real-time Multiplayer				
No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
8	Semua pemain meninggalkan permainan	Semua pemain meninggalkan <i>room</i> dan menghapus <i>room</i> . Data statistik <i>Rooms</i> menjadi 0	Data statistik <i>Rooms</i> pada photon server menjadi 0 yang menandakan pemain meninggalkan <i>room</i>	Berhasil

Hasil pengujian konektifitas *backend* yang ditunjukkan tabel 4.1, 4.2, 4.3 tersebut, menunjukkan bahwa dari keseluruhan poin yang diujikan, semuanya dapat memenuhi hasil yang diharapkan. Hal tersebut dapat dilihat dari 2 poin pada pengujian Firebase *authentication*, 5 poin pada pengujian Firebase *real-time database*, dan 8 poin pada pengujian Photon *real-time multiplayer* yang dapat berjalan sesuai dengan yang diharapkan.

e. Kesimpulan Pengujian

Berdasarkan hasil pengujian dari ketiga fitur yang digunakan pada permainan *online multiplayer* ular tangga bergenre strategi seperti Firebase *Authentication*, Firebase *Real-time Database*, dan Photon *Real-time Multiplayer*, diketahui bahwa layanan *backend* tersebut dapat berjalan dan berfungsi dengan baik ketika memainkan permainan. Hal tersebut membuat fitur *multiplayer* yang terdapat pada permainan dapat berjalan dengan baik.

4.2.2. Pengujian *Gameplay*

Pengujian *gameplay* adalah pengujian yang dilakukan untuk mengetahui apakah fungsi yang terdapat dalam permainan sudah berjalan sesuai dengan yang diharapkan atau tidak. Pengujian ini menggunakan metode *black box testing* yang akan menguji kesesuaian antara nilai *input* dengan hasil *output* dari beberapa fungsi pada permainan.

a. Tujuan Pengujian

Pengujian *gameplay* bertujuan untuk menjamin bahwa fungsi yang terdapat dalam permainan dapat berjalan sesuai dengan yang diharapkan dan bebas dari *bug*.

b. Prosedur Pengujian

Pengujian dimulai dengan memainkan permainan *online multiplayer* ular tangga bergenre strategi secara langsung oleh pemain untuk memastikan apakah fungsi yang terdapat pada aplikasi telah berjalan baik atau tidak. Setiap skenario pengujian akan dilakukan sebanyak 3 kali uji.

c. Hasil dan Pembahasan Pengujian

Hasil dari pengujian *gameplay* permainan dapat dilihat pada tabel 4.4. Berikut hasil dari pengujian *gameplay* yang dilakukan.

Tabel 4. 4 Hasil Pengujian Fungsi Aplikasi

No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
1	Pemain menekan tombol OK (halaman <i>log in</i>)	Menampilkan halaman menu utama	Sistem menampilkan halaman menu utama	Berhasil
2	Pemain menekan tombol bermain (halaman menu utama)	Menampilkan halaman <i>preparation game</i>	Sistem menampilkan halaman <i>preparation game</i>	Berhasil
		Menampilkan nama pemain dan pion yang digunakan pemain	Sistem menampilkan nama pemain dan pion yang digunakan pemain	Berhasil
3	Pemain menemukan lawan	Menampilkan nama lawan dan pion milik lawan	Sistem menampilkan nama lawan dan pion milik lawan	Berhasil
		Menampilkan hasil pengundian giliran bermain	Sistem menampilkan hasil pengundian giliran bermain	Berhasil
4	<i>Progress bar</i> mencapai 100 % (halaman <i>preparation game</i>)	Menampilkan pemberitahuan permainan dimulai dan pemberitahuan giliran bermain	Sistem menampilkan pemberitahuan permainan dimulai dan pemberitahuan giliran bermain	Berhasil

No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
5	Pemain menekan tombol batal (halaman <i>preparation game</i>)	Menampilkan halaman menu utama	Sistem menampilkan halaman menu utama	Berhasil
6	Pemain menekan tombol panduan bermain (halaman menu utama)	Menampilkan halaman tutorial permainan	Sistem menampilkan halaman tutorial permainan	Berhasil
7	Pemain menekan tombol keluar (halaman menu utama)	Keluar dari aplikasi	Sistem mengeluarkan pemain dari aplikasi	Berhasil
8	Pemain menekan tombol selanjutnya	Menampilkan slide gambar tutorial permainan selanjutnya	Sistem menampilkan slide gambar tutorial permainan selanjutnya	Berhasil
9	Pemain menekan tombol sebelumnya (halaman tutorial permainan)	Menampilkan slide gambar tutorial permainan sebelumnya	Sistem menampilkan slide gambar tutorial permainan sebelumnya	Berhasil
10	Pemain menekan tombol keluar (halaman tutorial permainan)	Menampilkan halaman menu utama	Sistem menampilkan halaman menu utama	Berhasil
11	Pemain menekan tombol dadu pada (halaman <i>in game</i>)	Dadu mengacak angka dan menampilkan angka yang di dapat	Sistem menampilkan dadu yang sedang mengacak angka dan menampilkan angka yang di dapat	Berhasil
12	Pion berjalan	<i>Point barmen</i> menampilkan poin yang didapat berdasarkan langkah pion	<i>Point barmen</i> menampilkan poin yang didapat berdasarkan langkah pion	Berhasil

No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
13	Pemain mendapatkan angka dadu 6	Menampilkan pemberitahuan bahwa pemain mendapat <i>bonus turn</i>	Sistem menampilkan pemberitahuan bahwa pemain mendapat <i>bonus turn</i>	Berhasil
14	Pemain menekan tombol selesai (halaman <i>in game</i>)	Menampilkan pemberitahuan bahwa giliran telah berganti dan pemberitahuan sedang menunggu lawan	Sistem menampilkan pemberitahuan bahwa giliran telah berganti dan pemberitahuan sedang menunggu lawan	Berhasil
15	Pemain menekan tombol kartu sesuai level (halaman <i>in game</i>)	<i>Point bar</i> berkurang sesuai dengan kartu yang dipilih	<i>Point bar</i> berkurang sesuai dengan kartu yang dipilih	Berhasil
		Menampilkan hasil kartu yang dipilih	Sistem menampilkan hasil kartu yang dipilih	Berhasil
		Tombol pasang dan jual aktif	Sistem mengaktifkan Tombol pasang dan jual	Berhasil
16	Pemain menekan tombol jual (halaman <i>in game</i>)	<i>Point bar</i> bertambah sesuai harga jual kartu	<i>Point bar</i> bertambah sesuai harga jual kartu	Berhasil
17	Pemain menekan tombol pasang (halaman <i>in game</i>)	Kotak pada papan permainan dapat dipilih atau ditekan	Kotak pada papan permainan dapat dipilih atau ditekan	Berhasil
18	Pemain menekan kotak <i>finish</i> pada papan permainan	Menampilkan notifikasi peringatan	Sistem menampilkan notifikasi peringatan	Berhasil
19	Pemain menekan kotak berisi objek (pion, <i>item</i>)	Menampilkan notifikasi peringatan	Sistem menampilkan notifikasi peringatan	Berhasil

No	Skenario Pengujian	Hasil yang diharapkan	Hasil yang diterima	Status
20	<i>Item</i> yang dipasang, mengenai objek lain (pion, <i>item</i> lain), berada di kotak <i>finish</i> , atau keluar area papan permainan	Menampilkan notifikasi peringatan	Sistem menampilkan notifikasi peringatan	Berhasil
21	Pemain menekan kotak yang tidak berisi objek	Memasang <i>item</i> sesuai hasil kartu yang didapat	Sistem memasang <i>item</i> sesuai hasil kartu yang didapat	Berhasil
		Menutup tampilan hasil kartu yang di dapat	Sistem menutup tampilan hasil kartu yang di dapat	Berhasil
22	Pion berhenti berjalan dan berada tepat di kepala <i>item</i> ular lawan	Pion berpindah posisi ke ekor <i>item</i> ular lawan	Sistem memindahkan pion ke ekor <i>item</i> ular lawan	Berhasil
		<i>Item</i> menghilang dari papan permainan	Sistem menghapus <i>item</i> dari papan permainan	Berhasil
23	Pion berhenti berjalan dan berada tepat di kepala <i>item</i> tangga pemain	Pion berpindah posisi ke ujung lainnya <i>item</i> tangga pemain	Sistem memindahkan pion ke ujung lainnya <i>item</i> tangga pemain	Berhasil
		<i>Item</i> menghilang dari papan permainan	Sistem menghapus <i>item</i> dari papan permainan	Berhasil
24	Umur <i>item</i> telah habis	<i>Item</i> menghilang dari papan permainan	Sistem menghapus <i>item</i> dari papan permainan	Berhasil
25	Pion berhenti berjalan dan berada tepat di kotak <i>finish</i>	Menampilkan hasil akhir permainan dan menuju halaman menu utama	Sistem menampilkan hasil akhir permainan dan menuju halaman menu utama	Berhasil

Dari hasil pengujian *gameplay* permainan tersebut, dapat diketahui bahwa dari seluruh skenario pengujian yang dilakukan semuanya dapat berjalan dan

memenuhi kriteria yang diharapkan. Terlihat pada tabel 4.4 semua tombol dan fungsi yang diujikan dapat berjalan dengan baik.

d. Kesimpulan Pengujian

Berdasarkan hasil pengujian *gameplay* yang telah dilakukan, seluruh fungsi pada permainan *online multiplayer* ular tangga bergenre strategi dapat digunakan dengan baik. Semua tombol dan fungsi yang terdapat pada aplikasi dapat digunakan oleh pemain sehingga permainan dapat berjalan dengan lancar.

4.2.3. Pengujian Performa

Pengujian performa merupakan pengujian untuk mengetahui performa PC atau Laptop dalam menjalankan permainan *online multiplayer* ular tangga bergenre strategi. Pengujian dilakukan dengan melihat aktifitas peningkatan pada *Random Access Memory* (RAM) dari perangkat PC atau Laptop karena menurut *Top Ten Reviews* (TopTenReviews, 2019), komponen tersebut merupakan bagian vital pada sebuah PC atau Laptop *gaming*.

a. Tujuan Pengujian

Pengujian performa bertujuan untuk mengetahui tingkat performa PC atau Laptop dalam menjalankan permainan *online multiplayer* ular tangga bergenre strategi di RAM yang berbeda-beda.

b. Lingkungan Pengujian

Berdasarkan apa yang dikutip oleh *Top Ten Reviews* (TopTenReviews, 2019), maka dari itu penulis melakukan pengujian pada RAM berkapasitas 2 GB sampai 8 GB dengan pembagian seperti berikut:

1. 2 PC atau Laptop dengan kapasitas RAM 2GB
2. 2 PC atau Laptop yang masing-masing berkapasitas RAM 2 GB dan 4 GB
3. 2 PC atau Laptop yang masing-masing berkapasitas RAM 2 GB dan 8 GB
4. 2 PC atau Laptop dengan kapasitas RAM 4 GB
5. 2 PC atau Laptop yang masing-masing berkapasitas RAM 4 GB dan 8 GB

6. 2 PC atau Laptop dengan kapasitas RAM 8 GB

c. Prosedur Pengujian

Setiap pengujian akan dilakukan 2 kali dengan salah satu perangkat PC atau Laptop bergantian menjadi *host* dalam permainan. Pengujian dimulai dengan melihat kondisi RAM sebelum memulai permainan dan ketika menjalankan permainan *online multiplayer* ular tangga bergenre strategi. Pengujian tersebut akan dilakukan selama 25 menit. Penulis kemudian akan mencatat kondisi RAM setiap 5 menit sekali.

d. Hasil dan Pembahasan

Hasil dan pembahasan dari pengujian tersebut dapat dilihat pada tabel 4.5 di bawah ini.

Tabel 4. 5 Hasil Pengujian Performa

No	Perangkat	RAM (GB)	Sebelum Bermain (GB)	Peningkatan Saat Bermain di 5 menit ke- (GB)					Rata-Rata (GB)
				1	2	3	4	5	
1	PC atau Laptop A (<i>Host</i>)	2	1	0,3	0,4	0,4	0,3	0,3	0,34
	PC atau Laptop B	2	0,7	0,2	0,3	0,4	0,3	0,3	0,3
2	PC atau Laptop A	2	1,1	0,3	0,3	0,2	0,3	0,2	0,26
	PC atau Laptop B (<i>Host</i>)	2	0,8	0,3	0,3	0,3	0,2	0,3	0,28
3	PC atau Laptop A (<i>Host</i>)	2	1,1	0,4	0,3	0,4	0,5	0,4	0,4
	PC atau Laptop B	4	2,4	0,3	0,3	0,3	0,3	0,2	0,28
4	PC atau Laptop A	2	1,2	0,3	0,4	0,3	0,3	0,3	0,32
	PC atau Laptop B (<i>Host</i>)	4	2,4	0,3	0,4	0,4	0,3	0,3	0,34
5	PC atau Laptop A (<i>Host</i>)	2	0,9	0,5	0,5	0,5	0,4	0,5	0,48
	PC atau Laptop B	8	3,2	0,5	0,6	0,6	0,5	0,5	0,54

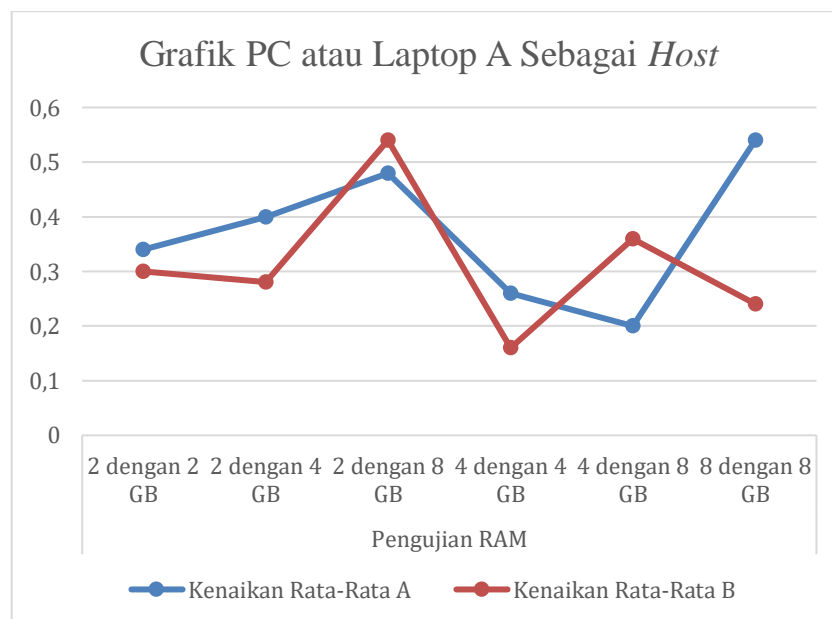
No	Perangkat	RAM (GB)	Sebelum Bermain (GB)	Peningkatan Saat Bermain di 5 menit ke- (GB)					Rata-Rata (GB)
				1	2	3	4	5	
6	PC atau Laptop A	2	0,9	0,4	0,4	0,5	0,5	0,5	0,46
	PC atau Laptop B (Host)	8	3,1	0,5	0,5	0,6	0,7	0,7	0,6
7	PC atau Laptop A (Host)	4	1,8	0,3	0,3	0,2	0,3	0,2	0,26
	PC atau Laptop B	4	2,5	0,1	0,2	0,2	0,1	0,2	0,16
8	PC atau Laptop A	4	1,7	0,3	0,3	0,2	0,3	0,3	0,28
	PC atau Laptop B (Host)	4	2,3	0,2	0,4	0,2	0,3	0,3	0,28
9	PC atau Laptop A (Host)	4	1,9	0,1	0,2	0,3	0,2	0,2	0,2
	PC atau Laptop B	8	3,2	0,4	0,4	0,3	0,3	0,4	0,36
10	PC atau Laptop A	4	2	0,2	0,2	0,1	0,1	0	0,12
	PC atau Laptop B (Host)	8	3,2	0,5	0,5	0,5	0,5	0,3	0,46
11	PC atau Laptop A (Host)	8	3,1	0,4	0,6	0,6	0,6	0,5	0,54
	PC atau Laptop B	8	2,8	0,2	0,2	0,2	0,3	0,3	0,24
12	PC atau Laptop A	8	3,2	0,6	0,5	0,5	0,5	0,4	0,5
	PC atau Laptop B (Host)	8	2,8	0,3	0,3	0,2	0,2	0,2	0,24

Dari hasil pengujian yang ditunjukkan pada tabel 4.5, dapat diketahui bahwa konsumsi RAM pada saat memainkan permainan *online multiplayer* ular tangga bergenre strategi berada dikisaran 0,1 GB sampai 0,7 GB pada semua jenis RAM. Dengan kondisi tersebut, permainan *online multiplayer* ular tangga bergenre strategi masih dapat berjalan dengan stabil dan lancar. Selain itu,

penulis juga menguji permainan dengan membandingkan antara perangkat PC atau Laptop yang menjadi *host* dengan perangkat PC atau Laptop yang menjadi *Client* pada permainan. Berikut hasil pengelompokan dari tabel di atas.

Tabel 4. 6 PC atau Laptop A Sebagai *Host*

RAM PC atau Laptop A (<i>Host</i>)	RAM PC atau Laptop B (<i>Client</i>)	Rata-rata Peningkatan	
		PC atau Laptop A (GB)	PC atau Laptop B (GB)
2	2	0,34	0,3
	4	0,4	0,28
	8	0,48	0,54
4	4	0,26	0,16
	8	0,2	0,36
8	8	0,54	0,24

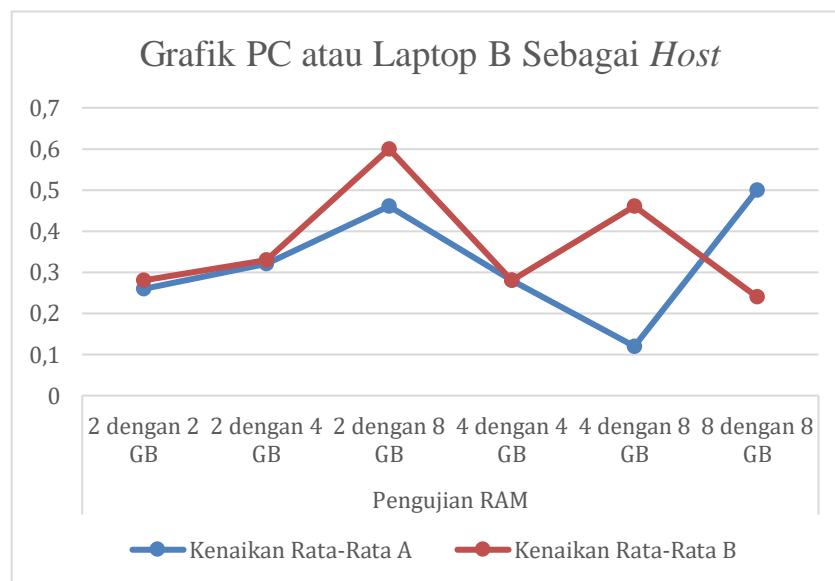


Gambar 4. 48 Grafik Ketika PC atau Laptop A Sebagai *Host*

Tabel 4. 7 PC atau Laptop B Sebagai *Host*

RAM PC atau Laptop B (<i>Host</i>)	RAM PC atau Laptop A (<i>Client</i>)	Rata-rata Peningkatan	
		PC atau Laptop B (GB)	PC atau Laptop A (GB)
2	2	0,28	0,26
4	2	0,33	0,32
	4	0,28	0,28

RAM PC atau Laptop B (<i>Host</i>)	RAM PC atau Laptop A (<i>Client</i>)	Rata-rata Peningkatan	
		PC atau Laptop B (GB)	PC atau Laptop A (GB)
8	2	0,6	0,46
	4	0,46	0,12
	8	0,24	0,5



Gambar 4. 49 Grafik Ketika PC atau Laptop B Sebagai *Host*

Berdasarkan pengelompokan yang ditunjukkan pada tabel 4.6 dan 4.7, rata-rata konsumsi RAM pada perangkat PC atau Laptop yang berperan sebagai *host* berada di sekitar 0,2 GB sampai 0,6 GB, sedangkan pada perangkat PC atau Laptop yang berperan sebagai *client* berada disekitar 0,12 GB sampai 0,54 GB. Selain itu, apabila dilihat pada grafik pada gambar 4.48 dan 4.49, perangkat PC atau Laptop yang menjadi *host* lebih mengonsumsi RAM lebih banyak daripada PC atau Laptop yang menjadi *client*. Hal ini menunjukkan bahwa perangkat PC atau Laptop yang berperan sebagai *host* sedikit lebih banyak mengonsumsi RAM pada perangkat. Namun hal tersebut masih dianggap wajar dan tidak mempengaruhi jalannya permainan antara dua perangkat.

e. Kesimpulan Pengujian

Berdasarkan hasil dari pengujian performa tersebut, dapat diketahui bahwa permainan *online multiplayer* ular tangga bergenre strategi dapat berjalan

dengan normal pada RAM yang berbeda-beda. Konsumsi RAM yang digunakan juga cukup stabil karena berada di kisaran 0,1 sampai 0,7 GB, baik ketika menjadi *host* ataupun *client* pada permainan. Hal tersebut juga dianggap wajar karena pada pengujian sebelumnya pada permainan *offline multiplayer*, konsumsi RAM berada di kisaran yang kurang lebih sama yaitu 0,2 sampai 0,7 GB.

4.2.4. Pengujian Portabilitas

Pengujian portabilitas merupakan pengujian untuk mengetahui apakah permainan *online multiplayer* ular tangga bergenre strategi dapat berjalan pada *Operating System* (OS) Windows yang berbeda-beda. Windows menjadi target pada pengujian ini karena OS tersebut menjadi pilihan terpopuler di dunia dalam mengembangkan sebuah *software* (Statista, 2018).

a. Tujuan Pengujian

Pengujian Portabilitas bertujuan untuk menguji kemampuan permainan *online multiplayer* ular tangga bergenre strategi yang berjalan pada OS Windows yang berbeda-beda.

b. Lingkungan Pengujian

Pengujian portabilitas akan diujikan pada *Operating System* (OS) Windows. Melihat hal tersebut, maka penulis menggunakan 6 PC atau Laptop yang menggunakan OS Windows 7, 8 atau 8.1, dan 10.

c. Prosedur Pengujian

Pengujian dimulai dengan melakukan instalasi permainan *online multiplayer* ular tangga bergenre strategi dan memainkan permainan selama 25 menit pada 6 PC atau Laptop dengan OS Windows yang berbeda.

d. Hasil dan Pembahasan

Hasil dari pengujian tersebut dapat dilihat pada tabel di bawah ini.

Tabel 4. 8 Hasil Pengujian Portabilitas Permainan pada OS Windows

No.	PC atau Laptop	OS Windows			Hasil Pengujian
		7	8 atau 8.1	10	
1	A	v	-	-	Berhasil
2	B	v	-	-	Berhasil

No.	PC atau Laptop	OS Windows			Hasil Pengujian
		7	8 atau 8.1	10	
3	C	-	v	-	Berhasil
4	D	-	v	-	Berhasil
5	E	-	-	v	Berhasil
6	F	-	-	v	Berhasil

Pada pengujian portabilitas yang telah dilakukan, permainan yang telah diinstal dan dimainkan selama 25 menit dapat berjalan dengan baik. Hal tersebut dapat dilihat pada tabel 4,8, dimana setiap perangkat PC atau Laptop yang memiliki OS berbeda-beda dapat menjalankan permainan digital *online multiplayer* ular tangga bergenre strategi.

e. Kesimpulan Pengujian

Dari hasil pengujian tersebut, permainan *online multiplayer* ular tangga bergenre strategi dapat berjalan pada OS Windows yang berbeda-beda. Namun perlu diketahui dari masing-masing perangkat PC atau Laptop yang digunakan untuk pengujian memiliki spesifikasi yang berbeda-beda.

4.3. Limitasi Penelitian

Dari penelitian yang telah dilakukan, terdapat beberapa kekurangan dan kelemahan di dalamnya. Kekurangan atau kelemahan pada tugas akhir ini adalah sebagai berikut.

1. Pengembangan permainan *online multiplayer* ular tangga bergenre strategi ini hanya dapat dimainkan di jaringan area lokal saja dengan kapasitas maksimal perangkat yang dapat terhubung dengan permainan ini sebanyak 20 perangkat PC atau Laptop.
2. *Game engine* yang digunakan dalam membangun permainan ini hanya mendukung fitur-fitur pengembangan permainan yang sederhana.
3. Pengujian RAM pada permainan ini hanya diuji pada dua perangkat saja, sehingga dapat terjadi perubahan standar kenaikan RAM pada permainan apabila dilakukan penambahan perangkat yang diuji.