**BAB VI**

**UJI COBA**

Bab ini akan menjelaskan tentang uji coba yang dilakukan selama pengerjaan Tugas Akhir ini. Uji coba yang dilakukan meliputi 3 metode uji coba yaitu White Box Testing, Black Box Testing, dan Kuesioner yang dijabarkan pertanyaan berserta jawabannya.

1. **White Box Testing**

White box testing, juga dikenal sebagai testing struktural, adalah metode pengujian perangkat lunak yang melibatkan pemeriksaan dan evaluasi struktur internal dari kode sumber. Pendekatan ini memerlukan pemahaman yang mendalam terhadap logika dan implementasi program yang diuji. White box testing yang dilakukan selama pengerjaan Tugas Akhir ini akan diuraikan dalam subbab di bawah ini.

1. **White Box Testing Karakter**

Bagian ini menjelaskan testing white box yang dilakukan pada karakter pemain. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.1.

**Tabel 6.1**

**Tabel White Box Testing Karakter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Gerak Kiri Kanan Karakter | Menekan tombol ‘A’ atau ‘D’ | Value x position dikurang saat menekan tombol ‘A’.  Value x position ditambah saat menekan tombol ‘D’. Play anims jalan. Flip jika tombol ‘A’ ditekan. | Berjalan sesuai target |

**Tabel 6.1**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 2 | Karakter Merangkak | Menekan tombol ‘S’ | Masuk kedalam kondisi modeJongkok(). Collider diubah menjadi jongkok, Play anims jongkok. | Berjalan sesuai target |
| 3 | Karakter Melompat | Menekan tombol ‘Space’ | Masuk kedalam kondisi modeLompat(). Value y pos dikurangi kemudian ditambah lagi sesuai dengan gravity. Play anims lompat. | Berjalan sesuai target |
| 4 | Karakter Lari | Menekan tombol ‘Shift’ | Value x position ditambah sejumlah dxLari saat menekan tombol ‘Shift’. | Berjalan sesuai target |
| 5 | Karakter Memukul | Menekan tombol ‘O’ | Masuk kedalam kondisi modePukul(), status isAttack == true. Play anims memukul. | Berjalan sesuai target |
| 6 | Karakter Kena Serang | Tidak sembunyi saat terkejar hantu | isHurting == true, healthPoins -= 1, beri efek sprite merah. Play anims karakter terkena serang. | Berjalan sesuai target |

1. **White Box Testing Rintangan Hantu**

Bagian ini menjelaskan testing white box yang dilakukan pada rintangan hantu. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.2.

**Tabel 6.2**

**Tabel White Box Testing Rintangan Hantu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Blok Hantu | Memberi pembeda warna blok. | buatObject(jalan\_hantu\_single, x, tinggi, layer\_jalan) tereksekusi. | Berjalan sesuai target |
| 2 | Cek Penginjakan | Melewati setiap blok yang tergenerate. | Collision.gameObject.  CompareTag(“Player”) == true, sudahDipijak == true. | Berjalan sesuai target |
| 3 | Memunculkan hantu | Melewati setiap blok yang tergenerate | <Hantu\_single>().startMuncul = true, karakter\_hantu.Set  Active(true); | Berjalan sesuai target |

1. **White Box Testing Rintangan Duri**

Bagian ini menjelaskan testing white box yang dilakukan pada rintangan duri. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.3.

**Tabel 6.3**

**Tabel White Box Testing Rintangan Duri**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Rintangan Duri | Melewati setiap blok yang tergenerate. | Function PCG\_Jebakan() dipanggil saat generate level. Sp.sprite=image\_jebakan  [index] berjalan. | Berjalan sesuai target |

**Tabel 6.3**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 2 | Pemain Terkena Duri | Pemain menginjak duri secara sengaja | gameObject.setActive=false, collision.gameObject. CompareTag(“Player”) == true, karakter.kenaSerangan berjalan | Berjalan sesuai target |

1. **White Box Testing Tempat Bersembunyi**

Bagian ini menjelaskan testing white box yang dilakukan pada spawn tempat bersembunyi. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.4.

**Tabel 6.4**

**Tabel White Box Testing Tempat Bersembunyi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Semak Bambu | Melewati setiap blok yang tergenerate. | Function PCG\_Bambu() dipanggil saat generate level. Sp.sprite=image\_bambu  [index] berjalan. | Berjalan sesuai target |
| 2 | Pemain Bersembunyi | Pemain bersembunyi dalam semak saat dikejar hantu. | Saat karakter didalam collider, karakter.setStatusHide(true), jika tidak maka karakter.setStatusHide(false) | Berjalan sesuai target |

1. **White Box Testing Jumpscare Penampakan**

Bagian ini menjelaskan testing white box yang dilakukan pada jumpscare penampakan. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.5.

**Tabel 6.5**

**Tabel White Box Testing Jumpscare Penampakan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Rintangan Jumpscare Penampakan | Melewati setiap blok yang tergenerate. | Function PCG\_kuburan\_sesajen dipanggil saat generate level. Sp.sprite=image\_kuburan\_  sesajen[randValue] berjalan. | Berjalan sesuai target |
| 2 | Pemain Menginjak Jumpscare Penampakan | Melewati setiap blok yang tergenerate. | doJumpscare() berhasil dipanggil saat collision.gameObject. CompareTag(“Player”) == true, dan  canvasJumpscare muncul. | Berjalan sesuai target |

1. **White Box Testing Pencatat Log**

Bagian ini menjelaskan testing white box yang dilakukan pada sistem pencatat log. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.6.

**Tabel 6.6**

**Tabel White Box Testing Pencatat Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Pencatatan Emosi | Membuat raut wajah | Value\_emosi mendapat suplay data dari EmotionsManager.Emotions | Berjalan sesuai target |

**Tabel 6.6**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
|  |  | takut, netral, dan marah. | library moodme. Dan variabel emotion dapat diisi label “emosi” dan return isi. |  |
| 2 | Pencatatan Jenis Rintangan | Melewati masing-masing jenis rintangan dan didebug | Ketika jenis rintangan tertentu ditrigger, akan mengembalikan data jenis\_rintangannya, dan berhasil mengisi jenis obstacle berserta status rintangan gagal atau sukses. | Berjalan sesuai target |

1. **White Box Testing Pembaca Log**

Bagian ini menjelaskan testing white box yang dilakukan pada sistem pembaca log. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.7.

**Tabel 6.7**

**Tabel White Box Testing Pembaca Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Pengecekan File Log | Membuat file log kosong dan menghapus file log dari direktori. | Function LogFilePath berhasil memberi return logFilePath Ketika ada file output.json, jika tidak ada maka akan muncul error pada console debug. | Berjalan sesuai target |

**Tabel 6.7**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 2 | Membaca Isi File Json | File json ditempatkan didirektori seharusnya, namun isinya kosong, salah format, dan berisi benar. | LogWrapper.FromJson<Log  Wrapper>(jsonContent) mendapat supply data, kemudian jika tidak terpenuhi kondisinya, maka akan muncul error pada console debug sesuai dengan pesan penyebab error. | Berjalan sesuai target |
| 3 | Membaca Json sesuai dengan parameter yang dibutuhkan DDA | Menyortir data json menjadi supply data untuk DDA | foreach(PembacaLog.Log  Entry entry in logEntries) berhasil melakukan looping data dan menyortir entry.obstacle json dan melakukan penyortiran entry.status perkategori rintangan. | Berjalan sesuai target |

1. **White Box Testing Penentuan Kategori Pemain**

Bagian ini menjelaskan testing white box yang dilakukan pada sistem penentuan kategori pemain melalui skoring. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.8.

**Tabel 6.8**

**Tabel White Box Testing Penentuan Kategori Pemain**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Mendapatkan jumlahJalan pada level sebelumnya | Membuat debug log berisi variabel global ctr\_pjrintangan | JumlahRintangan pada debug log level sebelumnya dimunculkan dengan benar, sehingga dapat ditampung pada variabel jumlahJalanRintangan. | Berjalan sesuai target |
| 2 | Mendapatkan value ctr\_rintangan\_  sukses, gagal, dan tidak\_injak | Melakukan debug log pada ke 3 variabel tersebut | Mendapatkan nilai yang benar dengan mencocokkan counter sukses dan gagal melewati rintangan serta jumlah rintangan yang tidak diinjak. | Berjalan sesuai target |
| 3 | Melakukan perhitungan performa pemain dan label pemain | Simulasi variabel pemain dengan angka statik, kemudian akan didebug log satu per satu kondisi pemain dilabeli “pemula”, “normal” dan “mahir”. | Semua kondisi untuk pelabelan berjalan sesuai dengan simulasi variabel statik yang menyerupai pemain menyelesaikan 1 level. Kemudian hasil debug kategori\_pemain sesuai dengan kondisi percabangan. | Berjalan sesuai target |

**Tabel 6.8**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 4 | Melakukan perhitungan emosi pemain | Melakukan debug log pada value emosi yang terdeteksi oleh moodme, kemudian membandingkan hasil emosi yang lebih dominan. Kemudian digunakan untuk validasi kategori pemain yang sudah dihitung. | Konsol log mengeluarkan nilai emosi pemain, kemudian diseleksi nilai emosi yang lebih dominan. Kemudian label emosi yang telah diproses divalidasi apakah emosinya dominan ‘takut’, ‘normal’ atau ‘marah’, jika memenuhi kondisi maka nilai label pemain akan diubah. | Berjalan sesuai target |
| 5 | Melakukan perbandingan label dengan pengatur jumlah rintangan | Variabel weight ceiling wc\_rintangan memiliki value default dan init\_rintangan diubah jumlahnya berdasarkan value wc\_rintangan yang sesuai dengan label pemain. | Value init\_rintangan berubah sesuai dengan pengaturan weight ceiling DDA. | Berjalan sesuai target |

1. **Black Box Testing**

Black box testing adalah metode pengujian perangkat lunak tanpa memperhatikan struktur internal atau logika kode. Pendekatan ini lebih fokus pada input dan output yang dihasilkan oleh sistem tanpa mempertimbangkan bagaimana perangkat lunak mencapai hasil tersebut. Black box testing yang dilakukan selama pengerjaan Tugas Akhir ini akan diuraikan dalam subbab di bawah ini.

1. **Black Box Testing Karakter**

Bagian ini menjelaskan testing black box yang dilakukan pada karakter pemain. Terdapat 5 macam testing yang dilakukan. Testing yang dilakukan meliputi testing gerak karakter, karakter merangkak, karakter melompat, karakter memukul, dan karakter kena serang. Penjelasannya dijabarkan dalam Tabel 6.9.

**Tabel 6.9**

**Tabel Black Box Testing Karakter**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Gerak Kiri Kanan Karakter | Menekan tombol ‘A’ atau ‘D’ | Karakter akan berjalan kekiri dan kekanan saat pemai menekan tombol ‘A’ atau ‘D’ | Berjalan sesuai target |
| 2 | Karakter Merangkak | Menekan tombol ‘S’ | Karakter akan bergerak merangkak | Berjalan sesuai target |
| 3 | Karakter Melompat | Menekan tombol ‘Space’ | Karakter akan melompat | Berjalan sesuai target |
| 4 | Karakter Lari | Menekan tombol ‘Shift’ | Karakter akan berlari | Berjalan sesuai target |

**Tabel 6.9**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 5 | Karakter Memukul | Menekan tombol ‘O’ | Karakter akan melakukan aksi memukul. | Berjalan sesuai target |
| 6 | Karakter Kena Serang | Tidak sembunyi saat terkejar hantu | Nyawa karakter akan berkurang 1 | Berjalan sesuai target |

1. **Black Box Rintangan Hantu**

Bagian ini menjelaskan testing black box yang dilakukan pada rintangan hantu. Terdapat 3 macam testing yang dilakukan. Testing yang dilakukan meliputi testing generate pijakan hantu, cek penginjakan, dan memunculkan hantu. Penjelasannya dijabarkan dalam Tabel 6.10.

**Tabel 6.10**

**Tabel Black Box Testing Hantu**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Blok Hantu | Melewati setiap blok. | Terdapat blok spawn hantu sejumlah X pada level yang muncul. | Berjalan sesuai target |
| 2 | Cek Penginjakan | Melewati setiap blok. | Berhasil mendeteksi pemain telah melewati blok dengan memunculkan hantu. | Berjalan sesuai target |
| 3 | Memunculkan hantu | Melewati setiap blok. | Animasi hantu berhasil digenerate dan dimunculkan saat pemain telah menginjak blok hantu. | Berjalan sesuai target |

1. **Black Box Rintangan Duri**

Bagian ini menjelaskan testing black box yang dilakukan pada rintangan hantu. Testing yang dilakukan meliputi testing generate rintangan duri, dan pemain terkena duri. Penjelasannya dijabarkan dalam Tabel 6.11.

**Tabel 6.11**

**Tabel Black Box Testing Rintangan Duri**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Rintangan Duri | Melewati setiap blok yang tergenerate. | Terdapat blok dengan rintangan duri diatasnya sejumlah X pada level yang muncul. | Berjalan sesuai target |
| 2 | Pemain Terkena Duri | Pemain menginjak duri secara sengaja | Nyawa pemain berhasil berkurang sebanyak 1 dan rintangan duri menghilang setelah diinjak. | Berjalan sesuai target |

1. **Black Box Rintangan Kabut**

Bagian ini menjelaskan testing black box yang dilakukan pada rintangan kabut. Terdapat 2 macam testing yang dilakukan. Testing yang dilakukan meliputi testing generate rintangan kabut, dan pemain menginjak blok rintangan kabut. Penjelasannya dijabarkan dalam Tabel 6.12.

**Tabel 6.12**

**Tabel Black Box Testing Kabut**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Rintangan Kabut | Melewati setiap blok yang tergenerate. | Terdapat blok kabut suara sejumlah X pada level yang muncul. | Berjalan sesuai target |

**Tabel 6.12**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 2 | Pemain Menginjak Blok Kabut | Melewati setiap blok yang tergenerate. | Intensitas kabut dilayar bertambah tebal. | Berjalan sesuai target |

1. **Black Box Testing Item**

Bagian ini menjelaskan testing black box yang dilakukan pada spawn item. Terdapat 3 macam testing yang dilakukan. Testing yang dilakukan meliputi testing generate item, mengambil item dan menggunakan item. Penjelasannya dijabarkan dalam Tabel 6.13.

**Tabel 6.13**

**Tabel Black Box Testing Spawn Item**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Item Pada Map | Melewati setiap blok yang tergenerate. | Terdapat sejumlah item pada level yang digenerate. | Berjalan sesuai target |
| 2 | Pemain Mengambil Item | Saat berdiri diatas item, karakter menekan tombol ‘E’. | Item yang terdapat pada level hilang 1 dan inventory item pemain bertambah 1. | Berjalan sesuai target |
| 3 | Pemain Menggunakan Item | Pada Rintangan X Pemain Menggunakan Item X. | Rintangan tertentu menjadi tidak efektif terhadap pemain dan jumlah item di inventory pemain berkurang 1. | Berjalan sesuai target |

1. **Black Box Testing Jumpscare Penampakan**

Bagian ini menjelaskan testing black box yang dilakukan pada rintangan jumpscare penampakan. Terdapat 2 macam testing yang dilakukan. Testing yang dilakukan meliputi testing generate blok jumpscare penampakan, dan pemain menginjak jumpscare penampakan. Penjelasannya dijabarkan dalam Tabel 6.14.

**Tabel 6.14**

**Tabel Black Box Testing Jumpscare Penampakan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Rintangan Jumpscare Penampakan | Melewati setiap blok yang tergenerate. | Terdapat blok jumpscare penampakan sejumlah X pada level yang muncul. | Berjalan sesuai target |
| 2 | Pemain Menginjak Jumpscare Penampakan | Melewati setiap blok yang tergenerate. | Muncul canvas hantu mengagetkan serta sound effect mengkagetkan saat pemain menginjak blok. | Berjalan sesuai target |

1. **Black Box Testing Tempat Bersembunyi**

Bagian ini menjelaskan testing black box yang dilakukan pada tempat bersembunyi. Terdapat 2 macam testing yang dilakukan. Testing yang dilakukan meliputi testing generate semak bambu, dan pemain bersembunyi. Penjelasannya dijabarkan dalam Tabel 6.15.

**Tabel 6.15**

**Tabel Black Box Testing Tempat Bersembunyi**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Generate Semak Bambu | Melewati setiap blok pada level. | Terdapat semak bambu sejumlah X pada level yang muncul. | Berjalan sesuai target |

**Tabel 6.15**

**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 2 | Pemain Bersembunyi | Pemain bersembunyi dalam semak saat dikejar hantu. | Saat karakter dikejar hantu dan bersembunyi didalam semak, jumlah nyawanya tetap tidak berkurang. | Berjalan sesuai target |

1. **Black Box Testing Pencatat Log**

Bagian ini menjelaskan testing black box yang dilakukan pada sistem pencatat log. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.16.

**Tabel 6.16**

**Tabel Black Box Testing Pencatat Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Pencatatan Emosi | Membuat raut wajah takut, netral, dan marah. | Slider emosi pada game bergerak menunjukan persentase emosi yang dideteksi. | Berjalan sesuai target |
| 2 | Pencatatan Jenis Rintangan | Melewati masing-masing jenis rintangan | Ketika level selesai, nilai performa pemain ditampilkan. Dapat dibandingan dengan isi file log yang digenerate. | Berjalan sesuai target |

1. **Black Box Testing Pembaca Log**

Bagian ini menjelaskan testing black box yang dilakukan pada sistem pembaca log. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.17.

**Tabel 6.17**

**Tabel Black Box Testing Pembaca Log**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Pengecekan File Log | Membuat file log kosong dan menghapus file log dari direktori. | Jika file log tidak ada didalam direktori, maka game membuat file log baru. | Berjalan sesuai target |
| 2 | Membaca Isi File Json | File json ditempatkan didirektori seharusnya, namun isinya kosong, salah format, dan berisi benar. | File log tetap dapat terbaca oleh game dan tidak berdampak apapun terhadap jalannya permainan. | Berjalan sesuai target |
| 3 | Membaca Json sesuai dengan parameter yang dibutuhkan DDA | Menyortir data json menjadi supply data untuk DDA | Panel report pada ending menampilkan performa pemain dengan benar. | Berjalan sesuai target |

1. **Black Box Testing Penentuan Kategori Pemain**

Bagian ini menjelaskan testing black box yang dilakukan pada sistem penentuan kategori pemain melalui skoring. Terdapat beberapa testing yang dilakukan. Penjelasannya dijabarkan dalam Tabel 6.18.

**Tabel 6.18**

**Tabel Black Box Testing Penentuan Kategori Pemain**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 1 | Mendapatkan nilai jumlahJalan pada level sebelumnya | Menyelesaikan level yang telah diberikan. | Jumlah jalan dan rintangan pada panel report sesuai dengan level yang diselesaikan. | Berjalan sesuai target |
| 2 | Mendapatkan skor performa ‘Pemula’ | Menyelesaikan level yang telah diberikan dengan ceroboh dan membuat ekspresi “Takut”. | Mendapatkan nilai ‘Pemula’ pada panel report dan jenis rintangan yang digenerate pada level berikutnya dipermudah. | Berjalan sesuai target |
| 3 | Mendapatkan skor performa ‘Mahir’ | Menyelesaikan level yang telah diberikan dengan sengaja menghindari sebagian besar rintangan dan membuat ekspresi “Netral”. | Mendapatkan nilai ‘Mahir’ pada panel report dan jumlah jenis rintangan yang digenerate pada level berikutnya dipertahankan, dan akan diturunkan saat mendapat skor Normal. | Berjalan sesuai target |

**Tabel 6.18**

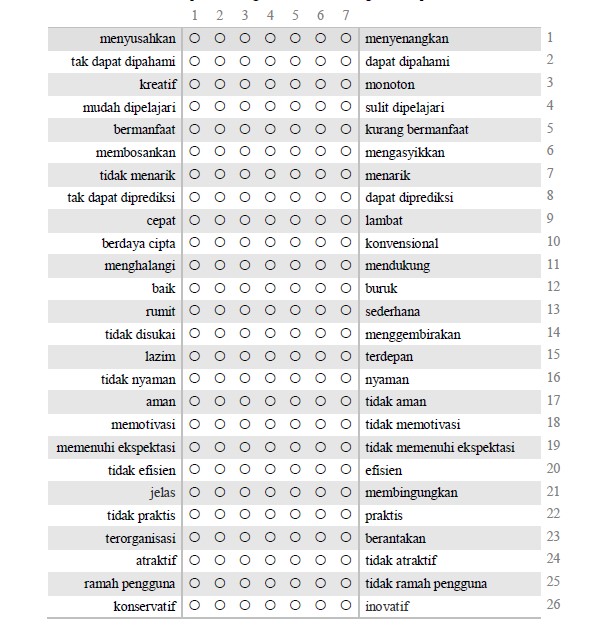
**(Lanjutan)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Keterangan** | **Skenario Testing** | **Target** | **Status** |
| 4 | Mendapatkan skor performa ‘Normal’ | Menyelesaikan level dengan sengaja mengenai sebagian rintangan. | Mendapatkan nilai ‘Normal’ pada panel report dan jumlah jenis rintangan berubah lebih sulit. | Berjalan sesuai target |
| 5 | Mendapatkan report rata-rata emosi pada jenis rintangan tertentu | Menyelesaikan level dengan rintangan yang ada. | Rata-rata emosi yang terdeteksi pada jenis rintangan tertentu berhasil ditampilkan pada panel report. | Berjalan sesuai target |

1. **Kuesioner**

Pada tahap ini metode yang digunakan yaitu kuesioner yang dirancang menggunakan UEQ. Game dimainkan oleh 30 responden yang memenuhi kriteria yaitu pernah memainkan game horror dan percaya akan hantu atau cerita mistis. Kemudian setelah bermain, responden diminta untuk mengisi form yang telah dirancang metode evaluasi UEQ dipilih karena kekuatan dan kelemahan dari game yang dibuat dapat diidentifikasi. Dan yang paling penting yaitu dengan UEQ pengembang akan mendapatk masukan yang berharga untuk meningkatkan kualitas pengalaman bermain.

UEQ sendiri merupakan alat evaluasi yang dapat diterapkan dalam desain pengalaman pengguna untuk menilai pandangan pengguna terhadap suatu produk atau layanan. Salah satu keunggulan utamanya adalah kemampuannya untuk memberikan gambaran menyeluruh tentang beragam aspek pengalaman pengguna. Proses kerjanya melibatkan pengumpulan data melalui kuesioner yang dirancang secara khusus, yang kemudian dianalisis untuk mengenali titik kuat dan kelemahan suatu produk atau layanan serta memberikan panduan untuk perbaikan lebih lanjut.



**Gambar 6.1**

**26 Skala Pertanyaan UEQ**

Kuesioner disusun dengan mengikuti pedoman 26 item pertanyaan yang mewakili 6 skala pada panduan UEQ yaitu daya tarik, kejelasan, efisiensi, ketepatan, stimulasi dan kebaruan seperti pada gambar 6.1 diatas[[1]](#footnote-1). Kemudian hasil yang diperoleh akan diolah melalui data analyst UEQ untuk mendapatkan hasil benchmark yang digunakan sebagai acuan perbaikan desain game kedepannya agar game yang dibuat mendapat user experience yang lebih baik. Pertanyaan yang dibuat melalui google form akan dijabarkan pada tabel dibawah ini.

**Tabel 6.19**

**Item Pertanyaan**

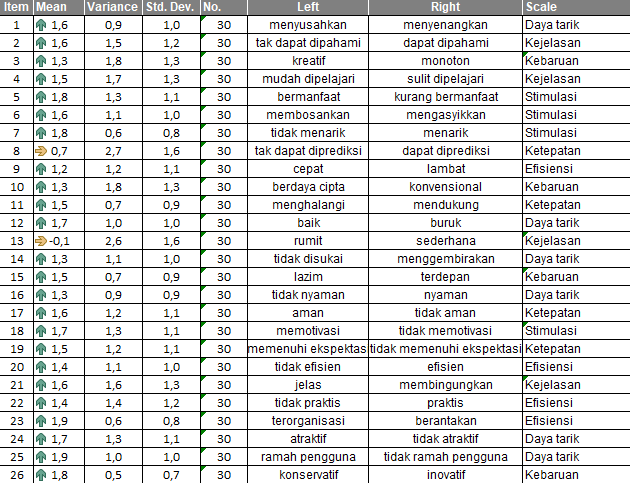
|  |  |  |
| --- | --- | --- |
| **No** | **Pertanyaan** | **Skala** |
| 1 | Apakah rancangan rintangan yang dibuat, membuat game menjadi menyenangkan untuk dimainkan? | Daya tarik |
| 2 | Seberapa mudah anda memahami konsep dan tujuan dari pengaturan rintangan yang dibuat? | Kejelasan |
| 3 | Seberapa kreatif menurut anda pengaturan tingkat kesulitan? | Kebaruan |
| 4 | Seberapa cepat anda dapat menguasai keterampilan yang diperlukan untuk mengatasi rintangan? | Kejelasan |
| 5 | Seberapa bermanfaatkah pengaturan rintangan dalam game? | Stimulasi |
| 6 | Seberapa mengasyikkan pengalaman bermain game ini? | Stimulasi |
| 7 | Seberapa menarik game dengan pengaturan rintangan ini menurut anda? | Stimulasi |
| 8 | Seberapa konsisten tingkat kesulitan disesuaikan dengan performa anda? | Ketepatan |
| 9 | Seberapa cepat tingkat kesulitan level berikutnya berubah? | Efisiensi |
| 10 | Menurut anda penggunaan konsep pengaturan tingkat kesulitan level menggunakan ekspresi wajah mengadopsi ide-ide baru? | Kebaruan |
| 11 | Seberapa efektif pengaturan rintangan dalam level berikutnya dalam meningkatkan pengalaman bermain pemain? | Ketepatan |
| 12 | Seberapa baik pengaturan rintangan dalam level berikutnya? | Daya tarik |
| 13 | Seberapa rumit pengaturan rintangan dalam level berikutnya? | Kejelasan |
| 14 | Seberapa menggembirakan pengalaman bermain yang anda dapatkan dengan adanya pengaturan rintangan level berikutnya? | Daya tarik |
| 15 | Seberapa terdepan ide pengaturan rintangan level berikutnya berdasarkan emosi pemain yang diterapkan kedalam game? | Kebaruan |
| 16 | Seberapa nyaman pengalaman bermain yang anda dapatkan dengan adanya pengaturan rintangan dalam level berikutnya? | Daya tarik |
| 17 | Seberapa aman konsep pengolahan data emosil pada game? | Ketepatan |
| 18 | Seberapa besar pengaturan rintangan memotivasi anda untuk menyelesaikan permainan? | Stimulasi |
| 19 | Sejauh mana pengalaman pemain dalam level berikutnya memenuhi ekspektasi anda? | Ketepatan |

**Tabel 6.19**

**(Lanjutan)**

|  |  |  |
| --- | --- | --- |
| 20 | Seberapa efisien menurut anda rancangan rintangan level berikutnya untuk membuat pemain merasa tertantang? | Efisiensi |
| 21 | Sejauh mana tujuan perancangan rintangan yang dibentuk dengan sedemikian rupa dapat dipahami dengan jelas oleh anda sebagai pemain? | Kejelasan |
| 22 | Seberapa praktis menurut anda persiapan dalam memainkan game ini? | Efisiensi |
| 23 | Seberapa terorganisasi menurut anda alur game yang dibuat dari menu hingga game berakhir? | Efisiensi |
| 24 | Seberapa atraktif menurut anda fitur-fitur dalam permainan ini? | Daya tarik |
| 25 | Seberapa ramah pengguna menurut anda keseluruhan game ini? | Daya tarik |
| 26 | Seberapa inovatif menurut anda penerapan konsep penyesuaian rintangan berdasarkan emosi pemain dalam game ini? | Kebaruan |

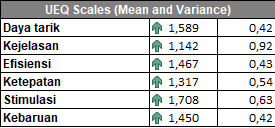
Hasil dari jawaban item pertanyaan yang diberikan berupa angka ber-range 1 hingga 7 yang kemudian dimasukkan kedalam sheet excel data analyst UEQ. Hasilnya berupa mean, variance, dan standar deviasi dari data yang diperoleh. Pehitungan ini dilakukan secara otomatis oleh rumus yang telah disusun tim UEQ.



**Gambar 6.2**

**Hasil Mean, Variance dan Standar Deviasi**

Nilai rata-rata >0.8 menunjukkan evaluasi positif (panah ke atas), sementara nilai rata-rata <-0.8 menunjukkan evaluasi negatif (panah ke bawah) dan jika nilai rata-rata berada di antara nilai positif dan negatif, akan ditandai dengan panah ke kanan. Kemudian angka evaluasi per item pertanyaan yang didapat, langkah selanjutnya yaitu mencari mean pada setiap skala yang dibuat sehingga akan ditemukan nilai mean dan variance yang didapat pada masing-masing 6 skala yang dievaluasi. Berikut merupakan hasil dari 6 skala yang dievaluasi berdasarkan data yang diperoleh.

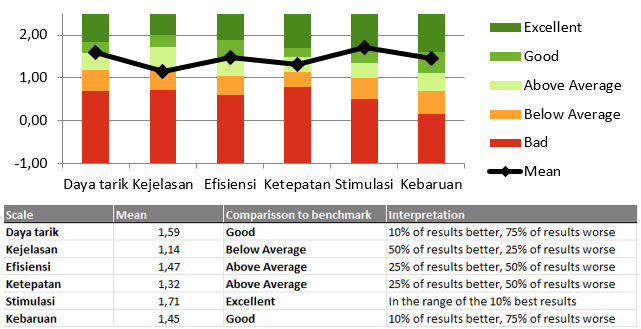


**Gambar 6.3**

**Hasil Mean dan Variance 6 Skala UEQ**

Hasil yang didapat pada seluruh skala mendapatkan nilai positif. Skala daya tarik mendapat nilai positif dengan mean atau rata-rata 1,589. Skala kejelasan mendapat nilai positif dengan mean atau rata-rata 1,142. Skala efisiensi mendapat nilai positif dengan mean atau rata-rata 1,467. Skala ketepatan mendapat nilai positif dengan mean atau rata-rata 1,317. Skala stimulasi mendapat nilai positif dengan mean atau rata-rata 1,708. Skala kebaruan mendapat nilai positif dengan mean atau rata-rata 1,450.

Setelah pengolahan data selesai dilakukan, untuk memperoleh feedback yang diperlukan untuk pengembangan rancangan rintangan atau pengembangan game kedepannya pada tahapan selanjutnya akan dilakukan benchmark UEQ yang berdasarkan dari data 21175 orang dari 467 studi produk yang telah menggunakan UEQ. Data yang diperoleh akan dijabarkan dibawah.



**Gambar 6.4**

**Hasil Benchmark UEQ**

Berdasarkan hasil benchmark yang dilakukan, nilai yang didapat pada skala daya tarik dan skala kebaruan mendapatan nilai benchmark ‘Baik’. Nilai yang didapat pada skala efisiensi dan ketepatan mendapatkan benchmark ‘Diatas rata-rata’. Nilai “Luar biasa” didapat pada skala stimulasi. Dan skala kejelasan mendapatkan nilai “Dibawah rata-rata”. Berdasarkan data diatas, dapat disimpulkan bahwa pada skala kejelasan perlu perbaikan lebih lanjut untuk meningkatkan experience bermain game yang dikembangkan.

Dari aspek daya tarik yang mendapat nilai mean 1,59 dari keseluruhan impresi pemain game ini, mengindikasikan penggunaan konsep pengaturan rintangan yang dapat menyesuaikan tingkat kesulitan berdasarkan ekspresi wajah pemain didalam game ini membuat game menarik untuk dimainkan bahkan untuk kesekian kalinya. Dari aspek kejelasan yang mendapatkan nilai mean 1,14 dari keseluruhan impresi pemain game ini, dapat ditarik bahwa diperlukan penjelasan lebih lanjut tentang tujuan dan fungsi pengaturan rintangan berdasarkan emosi pemain yang digunakan didalam game sehingga pemain menjadi paham bahwa tujuan utamanya yaitu meningkatkan kualitas bermain pemain serta membuat pemain tidak terlalu takut sehingga akan meninggalkan permainan. Dari aspek efisiensi yang mendapat nilai mean 1,47 dari keseluruhan impresi pemain game ini, tingkat efisiensi pengaturan rintangan yang dilakukan sudah cukup baik untuk menyesuaikan tingkat kesulitan terhadap performa pemain yang sedang memainkan game. Dari aspek ketepatan yang mendapat nilai mean 1,32 dari keseluruhan impresi pemain game ini, tingkat ketepatan pengaturan setiap jenis rintangan sudah cukup akurat menyesuaikan dengan performa masing-masing rintangan yang dilalui pemain didalam game. Dari aspek stimulasi yang mendapat nilai mean 1,71 dari keseluruhan impresi pemain game ini, dapat disimpulkan bahwa penggunaan rancangan rintangan yang dapat menyesuaikan pemainnya cukup berhasil dalam menstimulasi untuk pemain melanjutkan game hingga finish dan tidak berhenti ditengah jalan. Dan yang terakhir yaitu aspek kebaruan yang mendapat nilai mean 1,45 dari keseluruhan impresi pemain game ini, game yang dilengkapi dengan penyesuaian rintangan berdasarkan skor dan emosi pemain termasuk terbaharukan atau inovatif menurut sebagian besar pemain.

1. Hinderks, Andreas; Schrepp, Martin; Domínguez Mayo; Francisco José; Escalona, M.J.; Thomaschewski, Jörg. (2019). Developing a UX KPI based on the User Experience Questionnaire. Computer Standards & Interfaces, 65. [↑](#footnote-ref-1)