BAB 4
IMPLEMENTASI DAN PENGUJIAN SISTEM

4.1 Implementasi

Proses implementasi merupakan tahapan dimana aplikasi yang sudah dibuat dan dirancang, diuji kelayakannya.

4.1.1 Implementasi Perangkat Keras

Perangkat keras yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu sebagai berikut :

1. *Smartphone* Android *2.2 (froyo)* ke atas
2. Layar *touchscreen*
3. *Processor* 800 Mhz, 32 MB RAM

4.1.2 Implementasi Perangkat Lunak

Perangkat lunak yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu Sistem Operasi Android Android *2.2 (froyo)* keatas.

4.1.3 Implementasi Antarmuka

Pada bagian ini akan ditampilkan implementasi antarmuka menu utama, menu *play*, menu highscores, menu option, menu help dan menu *About.*

4.1.4 Implementasi Antarmuka Menu Utama

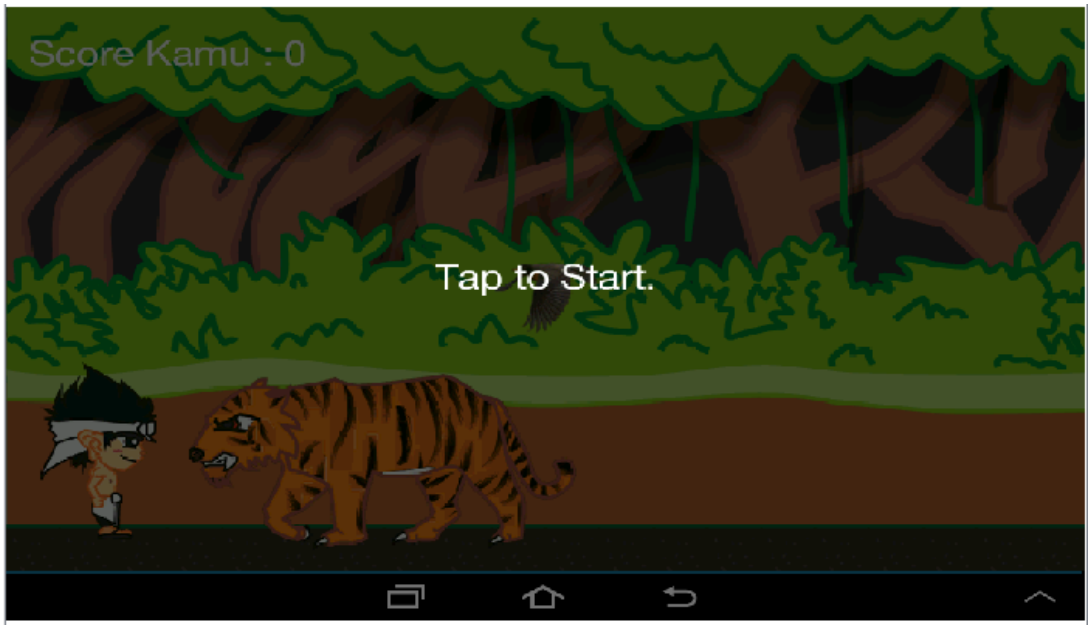
Berikut adalah Implementasi Antarmuka Menu Utama yang ditunjukkan oleh gambar di bawah ini.



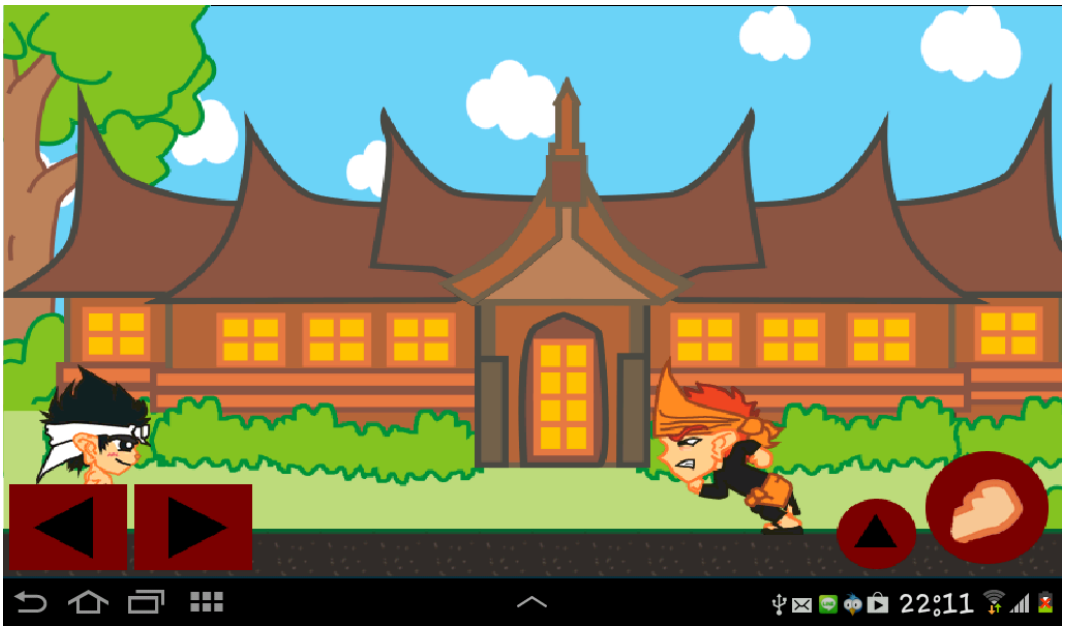
Gambar 4.1. Implementasi Antarmuka Menu Utama

4.1.5 Implementasi Antarmuka Menu *Play*

Berikut adalah Implementasi Antarmuka Menu *Play* yang ditunjukkan oleh gambar di bawah ini.



Gambar 4.2. Antarmuka Menu *Play* di Level Pulau Sumatera



Gambar 4.3. Antarmuka Menu *Play* di Level Pulau Sumatera (2)



Gambar 4.4. Antarmuka Menu *Play* di Level Pulau Kalimantan



Gambar 4.5. Antarmuka Menu *Play* di Level Pulau Sulawesi



Gambar 4.6. Antarmuka di Level Pulau Sulawesi (2)



Gambar 4.7. Antarmuka Menu *Play* di Level Pulau Papua



Gambar 4.8. Antarmuka di Level Pulau Papua (2)



Gambar 4.9. Antarmuka Menu *Play* di Level Pulau Jawa

4.1.6 Implementasi Antarmuka Prolog *Game*

Berikut adalah Implementasi Antarmuka Prolog *Game* Adventure "Sangkuriang and Indonesian Five Largest Islands" yang ditunjukkan oleh gambar di bawah ini.



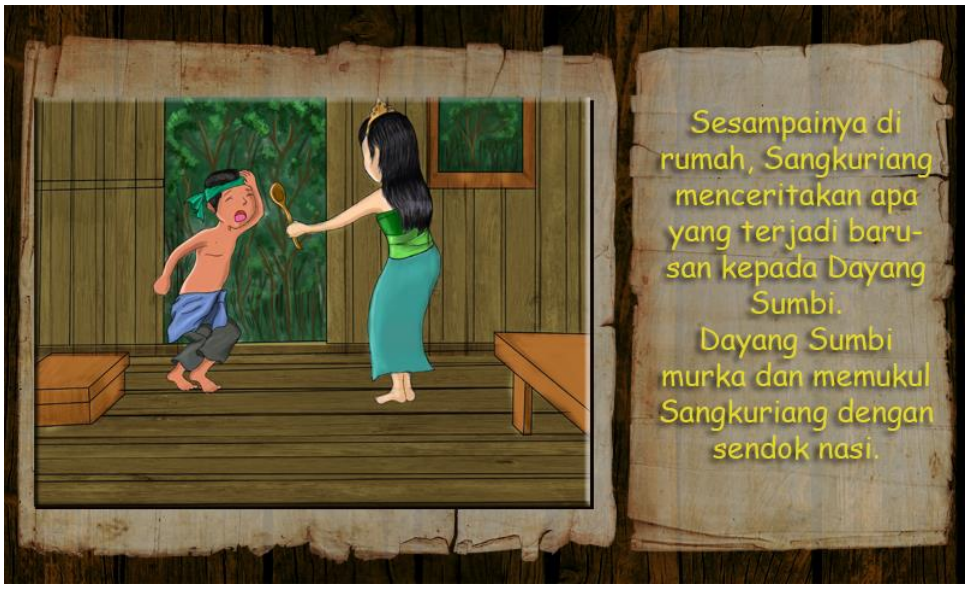
Gambar 4.10. Antarmuka Prolog *Game* 1



Gambar 4.11. Antarmuka Prolog *Game* 2



Gambar 4.12. Antarmuka Prolog *Game* 3



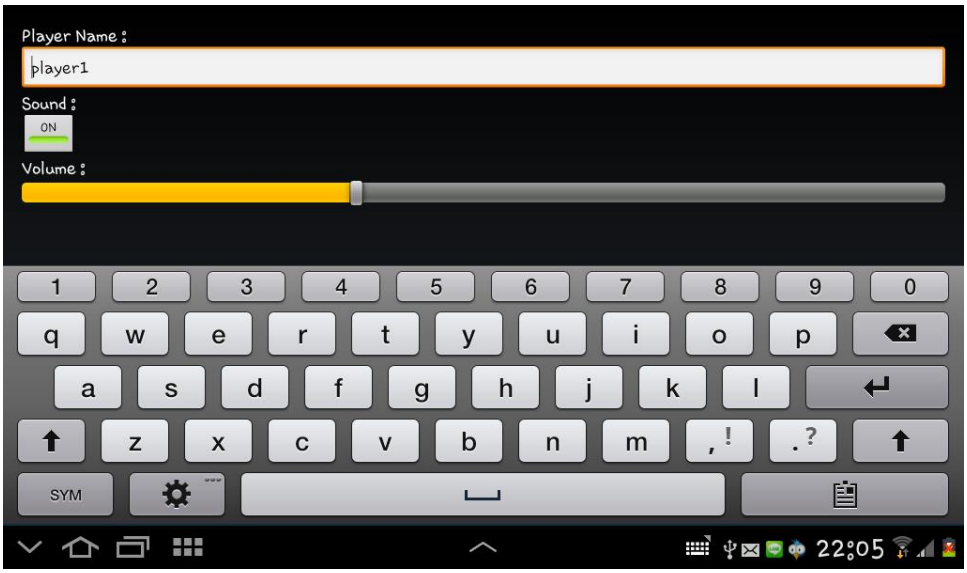
Gambar 4.13. Antarmuka Prolog *Game* 4



Gambar 4.14. Antarmuka Prolog *Game* 5

4.1.7 Implementasi Antarmuka Menu Option

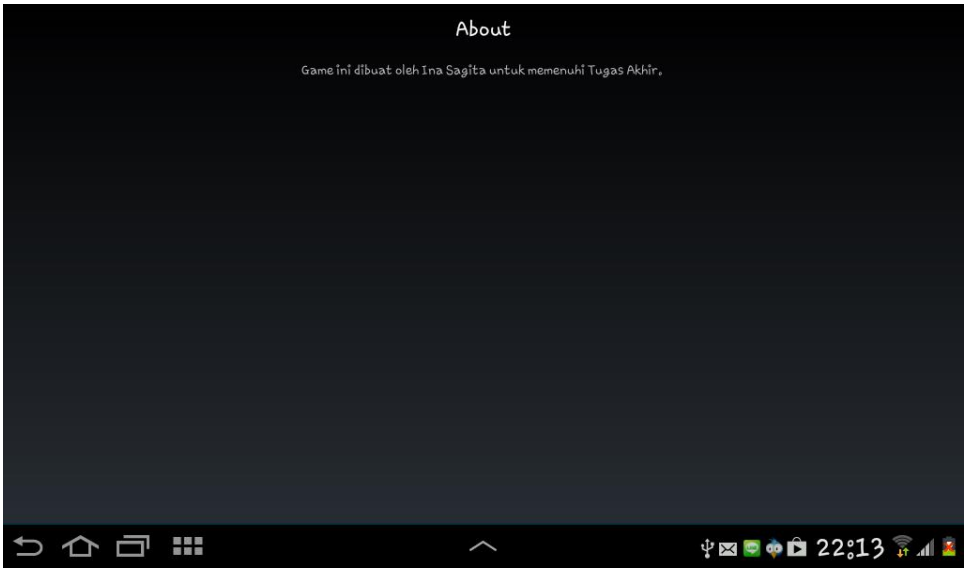
Berikut adalah Implementasi Antarmuka Menu *Option* yang ditunjukkan oleh gambar di bawah ini.



Gambar 4.15. Implementasi Antarmuka Menu Option

4.1.8 Implementasi Menu *About*

Berikut adalah Implementasi Antarmuka Quiz yang ditunjukkan oleh gambar gambar di bawah ini.



Gambar 4.16. Implementasi Menu *About*

4.1.9 Implementasi Pesan

Pada bagian ini akan ditampilkan implementasi pesan *pause menu*. Berikut adalah Implementasi Pesan *Pause Menu* yang ditunjukkan oleh gambar gambar di bawah ini.



Gambar 4.17. Implementasi Pesan Pause Menu

4.2 Pengujian Sistem

Dalam melakukan pengujian terhadap aplikasi permainan game adventure ini dilakukan dengan dua macam teknik, yaitu pengujian black-box dan pengujian white-box. Adapun pengujian black box berfokus pada menemukan kesalahan program, sedangkan pengujian white box berfokus kepada struktur internal (source code) program untuk mengetahui apakah masih terjadi *error*.

4.2.1.1 Skenario Pengujian

Skenario pengujian menjelaskan pengujian terhadap sistem yang ada pada aplikasi game adventure "Sangkuriang and Indonesian Five Largest Islands". Skenario pengujian dari aplikasi permainan ini dapat dilihat pada tabel di bawah ini.

Tabel 4.1. Skenario Pengujian

Pengujian	Butir Uji	Jenis Pengujian
Menampilkan level permainan	Memilih menu *Play*	*Black Box*
	Menampilkan Prolog	*Black Box*
	Menampilkan arena permainan	*Black Box*

	Karakter bergerak ke kanan	*Black Box*
	Karakter bergerak ke kiri	*Black Box*
	Karakter meloncat	*Black Box*
	Karakter menabrak musuh	*Black Box*
	Musuh mati jika dipukul pemain	*Black Box*
	Score bertambah jika musuh mati	*Black Box*
	Musuh mengikuti pemain	*Black Box*
	Perhitungan skor	*Black Box*
	Algorima A*	*White Box*
Menampilkan menu option	Menampilkan menu pengaturan *game*	*Black Box*
Mengatur nama pemain	Memasukkan nama pemain yang dinginkan	*Black Box*
Mengatur volume	Geser tombol volume arah ke kanan	*Black Box*
	Geser tombol volume arah ke kiri	*Black Box*
Mengatur sound	Memilih menu off	*Black Box*
	Memilih menu on	*Black Box*
Menampilkan menu *About*	Menampilkan halaman tentang pembuat *game*	*Black Box*
Menutup aplikasi	Memilih menu exit	*Black Box*

4.2.1.2 Pengujian Black Box

Tabel 4.2 berikut adalah hasil pengujian *black box* yang dilakukan sesuai skenario pengujian yang telah dirancang sebelumnya :

Tabel 4.2. Hasil Pengujian Black Box

Menampilkan Level Permainan		
Masukan	Hasil yang diharapkan	Hasil pengujian
Sentuh menu *Play*	Menampilkan prolog	[√] diterima [ ] ditolak
	Menampilkan arena permainan pada level 1	[√] diterima [ ] ditolak
	Menampilkan *adventure game*	[√] diterima [ ] ditolak
Sentuh tombol arah kanan di *touchscreen*	Karakter bergerak ke kanan	[√] diterima [ ] ditolak
Sentuh tombol arah kiri di *touchscreen*	Karakter bergerak ke kiri	[√] diterima [ ] ditolak
Sentuh tombol arah ke atas di *touchscreen*	Karakter meloncat	[√] diterima [ ] ditolak
Karakter menabrak monster	Karakter mati	[√] diterima [ ] ditolak
Karakter mengenai ranjau	Karakter mati	[√] diterima [ ] ditolak
Karakter berada dalam radius kejar monster	Musuh mengikuti pemain	[√] diterima [ ] ditolak
Perhitungan Skor		
Menyelesaikan masing-masing level	Dapat menampilkan informasi skor yang di dapat.	[√] diterima [ ] ditolak

Menampilkan menu option		
Masukan	Hasil yang diharapkan	Hasil pengujian
Sentuh menu option	Menampilkan menu pengaturan *game*	[√] diterima [ ] ditolak
Mengatur Nama Pemain		
Masukan	Hasil yang diharapkan	Hasil pengujian
Sentuh menu *Play*er Name (pada menu *option*)	Memasukkan nama pemain yang di inginkan	[√] diterima [ ] ditolak
Mengatur Volume		
Masukan	Hasil yang diharapkan	Hasil pengujian
Geser tombol volume arah ke kanan	Membesarkan volume suara *game*	[√] diterima [ ] ditolak
Geser tombol volume arah ke kiri	Mengecilkan volume suara *game*	[√] diterima [ ] ditolak
Mengatur Sound		
Masukan	Hasil yang diharapkan	Hasil pengujian
Sentuh pilihan On (pada menu *Sound)*	Menghidupkan sound *game*	[√] diterima [ ] ditolak
Sentuh pilihan Off (pada menu *Sound)*	Mematikan Sound *Game*	[√] diterima [ ] ditolak
Menampilkan Menu *About*		

Masukan	Hasil yang diharapkan	Hasil pengujian
Sentuh menu *About*	Menampilkan halaman tentang pembuat *game*	[√] diterima [ ] ditolak
Menutup Aplikasi		
Masukan	Hasil yang diharapkan	Hasil pengujian
Sentuh menu exit	Sistem menutup aplikasi	[√] diterima [ ] ditolak

4.2.1.3 Pengujian White Box

Di bawah ini merupakan kasus algoritma A* (A star) untuk menguji perangkat lunak yang sudah dibangun dengan menggunakan metode white box, berdasarkan skenario pengujian aplikasi yang sudah dijabarkan.

a. Algorima A*

```
1. endNode = new PathNode(null, null, endTile, 0);
   startNode = new PathNode(null, endNode, startTile, 0);
   addNodeToOpenList(startNode);
   while (openList.Count > 0)
      {
       PathNode currentNode = openList[openList.Count - 1];
       if (currentNode.IsEqualToNode(endNode))
      {
        List<Vector2> bestPath = new List<Vector2>();
        while (currentNode != null)
      {
        bestPath.Insert(0, currentNode.GridLocation);
        currentNode = currentNode.ParentNode;
      }
        return bestPath;
      }
2. public Node(int x, int y, int cost, int sog)
   {
    if(sog==1)
```

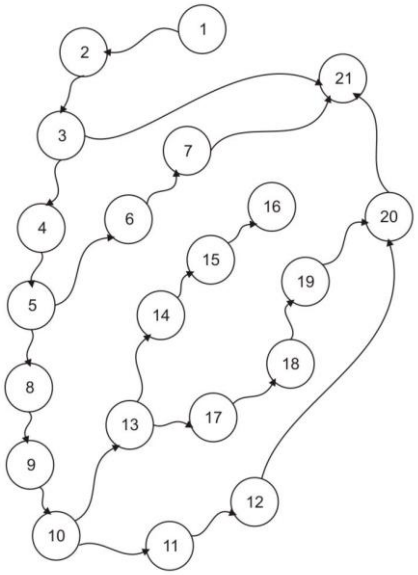
```
        {
        gesperrt = true;
        }
        xCoor = x;
        yCoor = y;
        NodeCost = cost;
        //perhitungan H
      directPath = 10*(Math.sqrt(Math.pow((xCoor-
      (world.getsangkuriangXcoor())), 2) + Math.pow((yCoor-
      (world.getsangkuriangYcoor())), 2)));
      previousPath = 10*(Math.sqrt(Math.pow((xCoor-
      (world.getStartXcoor())), 2) + Math.pow((yCoor-
      (world.getStartYcoor())), 2)));
       String h = String.valueOf(directPath);
       String g = String.valueOf(previousPath);
       F = directPath + previousPath + cost;
       String f = String.valueOf(F);
       if(cost>world.getNormalCost())
      }
      }
   3. public void makeAstar(MapParser parsedMap) {
// membuat daftar terbuka dari node, awalnya hanya berisisimpul
      mulai
   4. ArrayList<Node> openList = new ArrayList<Node>();
// membuat daftar tertutup dari node, awalnya kosong
   5. Map closedMap = new Map(rowSize, columnSize, "Closed map");
// Isi Peta tertutup dengan ini 0
   6. closedMap.buildMapWithValue(0);
// Membuat daftar tetangga
   7. ArrayList<Node> neighbourList = new ArrayList<Node>();
   Some vars
       int rowTemp = 0;
       int colTemp = 0;
int currentCost = 0;
       int tempCost = 0;
      // Inisialisasi dengan simpul pertama
   8. openList.add(new Node(0,0,9,this.getMapValue(0, 0)));
```

```
   9. while(!openList.isEmpty()) {
// Consider the best node, by sorting list according to F
   10.      Node.sortByF(openList);
Node currentNode = openList.get(0);
openList.remove(0);
// Berhenti jika mencapai tujuan
   11.      if(currentNode.getCol() == 4 && currentNode.getRow()
     == 4) {
System.out.println("Reached goal");
break;     }
// Pindahkan node saat ini ke daftar tertutup
   12.      closedMap.setMapValue(currentNode.getRow(),
      currentNode.getCol(), 1);
      // Dapatkan tetangga
   13.      rowTemp = currentNode.getRow()-1;
            colTemp = currentNode.getCol();
   14.      if(!currentNode.isTopRow()) {  // Add value ^
       neighbourList.add(new Node(rowTemp,
      colTemp,rowTemp+colTemp,this.getMapValue(rowTemp,
      colTemp)));}
   15.      rowTemp = currentNode.getRow();
            colTemp = currentNode.getCol()-1;
   16.       if(!currentNode.isRightColumn()) { // Add value <
       neighbourList.add(new Node(rowTemp,
      colTemp,rowTemp+colTemp,this.getMapValue(rowTemp,
      colTemp)));}
   17.      rowTemp = currentNode.getRow();
            colTemp = currentNode.getCol()+1;
   18.      if(currentNode.isLeftColumn()) { // Add value >
       neighbourList.add(new Node(rowTemp,
      colTemp,rowTemp+colTemp,this.getMapValue(rowTemp,
      colTemp)));      }
   19.      rowTemp = currentNode.getRow()+1;
            colTemp = currentNode.getCol();
   20.      if(currentNode.isBottomColumn()) { // Add value
      vneighbourList.add(new Node(rowTemp,
      colTemp,rowTemp+colTemp,this.getMapValue(rowTemp,
```

```
      colTemp)));             }
// Selama daftar tetangga tidak kosong
   21.      while(!neighbourList.isEmpty())
      {Node temp = neighbourList.get(0);
      neighbourList.remove(0);
       if(!isNotInClosedMap(temp.getRow(), temp.getCol())) {
continue;
                }
            }
         }
    }
```

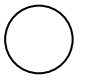
Gambar 4.18. Algoritma A* (A Star)

b. Flow Graph



Gambar 4.19. Flow Graph A *

Keterangan :

 = Menggambarkan kondisi

 = Menggambarkan aksi

c. Cyclomatic Complexity (V(G)

V(G) = E – N + 2

V(G) = 23 – 21 + 2

V(G) = 4

Di mana:

E = jumlah edge pada grafik alir

N = jumlah node pada grafik alir

d. Independent path

Path 1 = 1-2-3-4-5-6-7-21

Path 2 = 1-2-3-21

Path 3 = 1-2-3-4-5-8-9-10-11-12-20-21

Path 4 = 1-2-3-4-5-6-7-8-9-10-13-17-18-19-20-21

e. Graph Matrices

Berikut ini adalah graph matrices dari algoritma A*yang dapat dilihat pada tabel 4.8

Tabel 4.3. Graph Matrics

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	Hasil
1		1																				0
2			1																			0
3				1																	1	1
4					1																	0
5						1																0
6							1															0
7																					1	0
8									1													0
9										1												0
10											1											0
11												1										0
12																				1		0
13														1			1					1
14															1							0
15																1						0
16																				1		0

17																		1				0
18																			1			0
19																				1		0
20									1												1	1
21																						0
Hasil																						3

V(G) = Jumlah Graph Matric + 1

V(G) = 3 + 1

V(G) = 4

Berdasarkan hasil pengujian dihasilkan nilai *Cyclomatic Complexity* yang sama yaitu 4, maka dapat disimpulkan bahwa pada proses pencarian dengan algoritma A* (A *star*) berjalan dengan baik, karena setiap pengujian menghasilkan nilai yang sama.

4.2.2 Pengujian *Beta*

Pengujian beta merupakan pengujian yang dilakukan secara objektif. Pengujian dilakukan terhadap 40 responden yang merupakan siswa berumur 10 tahun ke atas. Responden disesuaikan dengan analisis pengguna dimana kriteria secara umumnya adalah bisa menggunakan smartphone android, laki-laki atau perempuan, bisa membaca dengan lancar, dan tidak ada masalah pendengaran. Adapun pengujian dilakukan hanya pada level 1 dikarenakan keterbatasan waktu. Metodologi penelitiannya adalah dengan metode deskriptif yaitu salah satu jenis metode penelitian yang berusaha menggambarkan dan menginterpretasi objek sesuai dengan apa adanya.

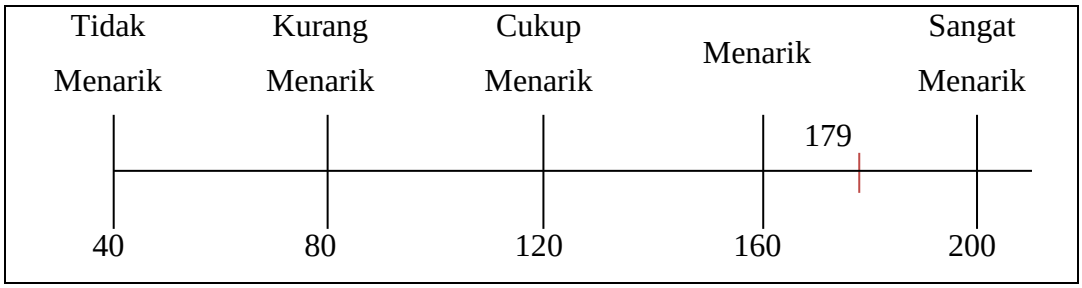
Pengujian yang dilakukan adalah dengan cara membagikan Kuisioner, dilakukan untuk mengetahui ketertarikan anak-anak terhadap *game* yang dibangun, dan untuk mengetahui apakah *game* ini bisa menjadi sarana pengenalan budaya Indonesia yang menantang dan menyenangkan. Pembuatan kuisioner menggunakan skala likert, dimana masing-masing pilihan diberi poin 5 sampai 1. Berikut ini adalah daftar pertanyaan kuisioner yang diberikan, serta hasil yang didapat setelah nilainya dihitung.

1. Menurut adik-adik, apakah tampilan *game* ini menarik?

Tabel 4.4. Hasil Pengolahan Data Kuisioner Pertanyaan (1)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Menarik	5	23	115
Menarik	4	13	52
Cukup Menarik	3	4	12
Kurang Menarik	2	0	0
Tidak Menarik	1	0	0
		TOTAL	179

Jika semua responden (40 orang responden) menjawab Sangat Menarik, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 179. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 1 adalah (179:200) x 100% = 89,5% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 179 terletak pada daerah menarik. Secara kontinum dapat digambarkan sebagai berikut :



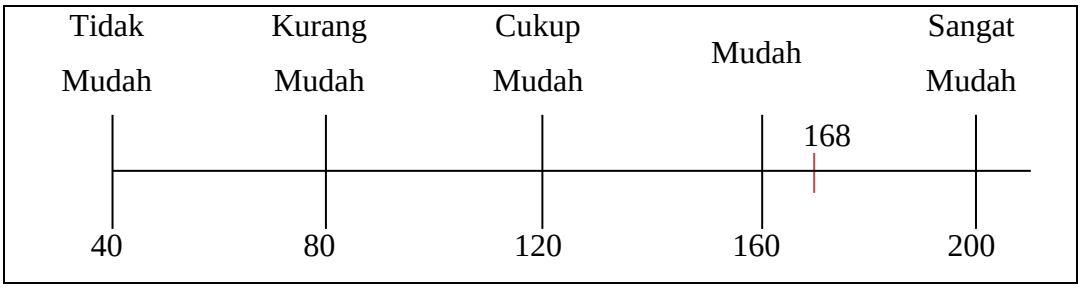
Gambar 4.20. Interpretasi Skor Pertanyaan Ke-1

2. Menurut adik-adik, apakah *game* ini mudah digunakan?

Tabel 4.5. Hasil Pengolahan Data Kuisioner Pertanyaan (2)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Mudah	5	14	70
Mudah	4	20	80
Cukup Mudah	3	6	18
Kurang Mudah	2	0	0
Tidak Mudah	1	0	0
		TOTAL	168

Jika semua responden (40 orang responden) menjawab Sangat Mudah, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 168. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 2 adalah (168:200) x 100% = 84% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 168 terletak pada daerah mudah. Secara kontinum dapat digambarkan sebagai berikut :



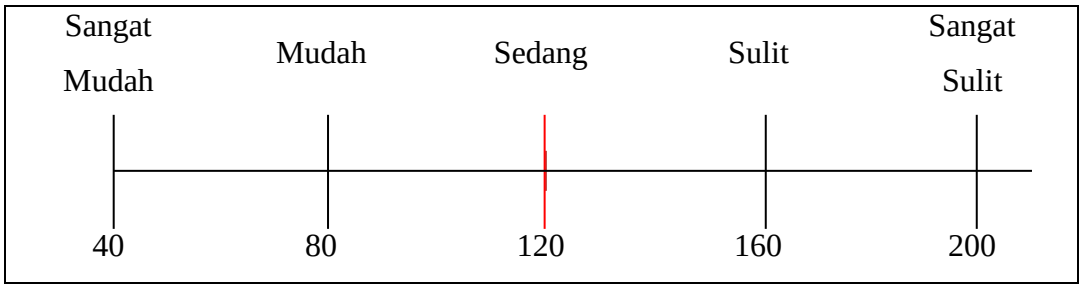
Gambar 4.21. Interpretasi Skor Pertanyaan Ke-2

3. Menurut adik-adik, bagaimana tingkat kesulitan *game* ini?

Tabel 4.6. Hasil Pengolahan Data Kuisioner Pertanyaan (3)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Sulit	5	0	0
Sulit	4	6	24
Sedang	3	29	87
Mudah	2	4	8
Sangat Mudah	1	1	1
		TOTAL	120

Jika semua responden (40 orang responden) menjawab Sangat Sulit, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 120. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 3 adalah (120:200) x 100% = 60% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 120 terletak pada daerah sedang. Secara kontinum dapat digambarkan sebagai berikut :



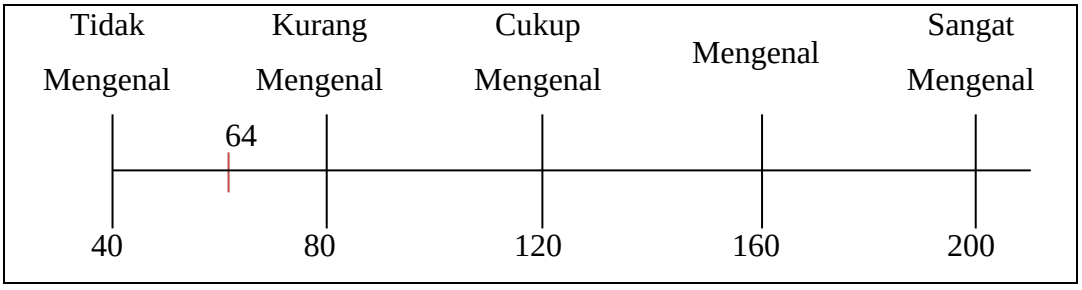
Gambar 4.22. Interpretasi Skor Pertanyaan Ke-3

4. Apakah sebelum memainkan *game* ini, adik-adik mengenal keanekaragaman budaya Indonesia yang terdapat dalam *game*?

Tabel 4.7. Hasil Pengolahan Data Kuisioner Pertanyaan (4)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Mengenal	5	0	0
Mengenal	4	0	0
Cukup Mengenal	3	2	6
Kurang Mengenal	2	20	40
Tidak Mengenal	1	18	18
		TOTAL	64

Jika semua responden (40 orang responden) menjawab Sangat Mengenal, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 64. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 4 adalah (64:200) x 100% = 32% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 64 terletak pada daerah tidak mengenal. Secara kontinum dapat digambarkan sebagai berikut :



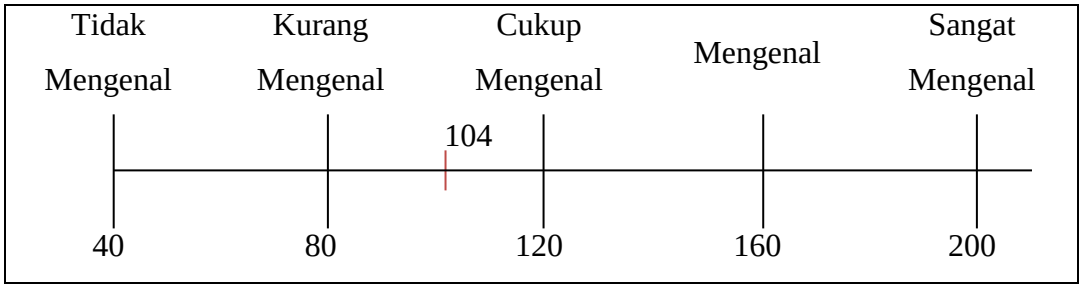
Gambar 4.23. Interpretasi Skor Pertanyaan Ke-4

5. Apakah sebelum memainkan *game ini*, adik-adik sudah mengenal legenda Sangkuriang ?

Tabel 4.8. Hasil Pengolahan Data Kuisioner Pertanyaan (5)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Mengenal	5	0	0
Mengenal	4	5	20
Cukup Mengenal	3	17	51
Kurang Mengenal	2	15	30
Tidak Mengenal	1	3	3
		TOTAL	104

Jika semua responden (40 orang responden) menjawab Sangat Mengenal, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 104. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 5 adalah (104:200) x 100% = 52% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 104 terletak pada daerah kurang mengenal. Secara kontinum dapat digambarkan sebagai berikut :



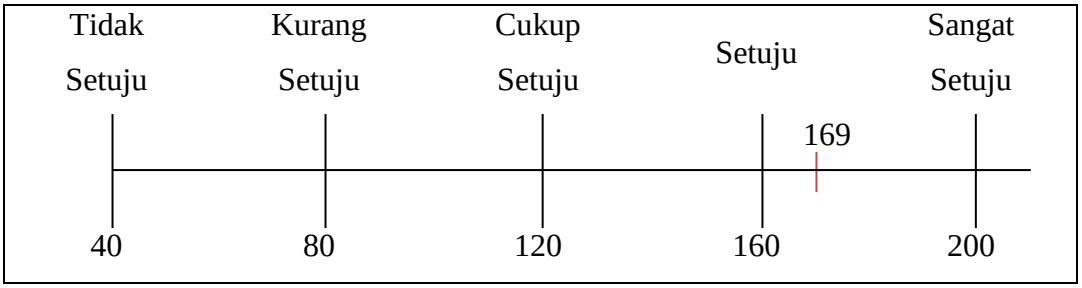
Gambar 4.24. Interpretasi Skor Pertanyaan Ke-5

6. Apakah setuju dengan adanya *game* ini, adik-adik menjadi tahu keanegaraman budaya Indonesia dan Legenda Sangkuriang?

Tabel 4.9. Hasil Pengolahan Data Kuisioner Pertanyaan (6)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Setuju	5	10	50
Setuju	4	23	92
Cukup Setuju	3	7	27
Kurang Setuju	2	0	0
Tidak Setuju	1	0	0
		TOTAL	169

Jika semua responden (40 orang responden) menjawab Sangat Setuju, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 169. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 6 adalah (169:200) x 100% = 78,5% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 169 terletak pada daerah setuju. Secara kontinum dapat digambarkan sebagai berikut :



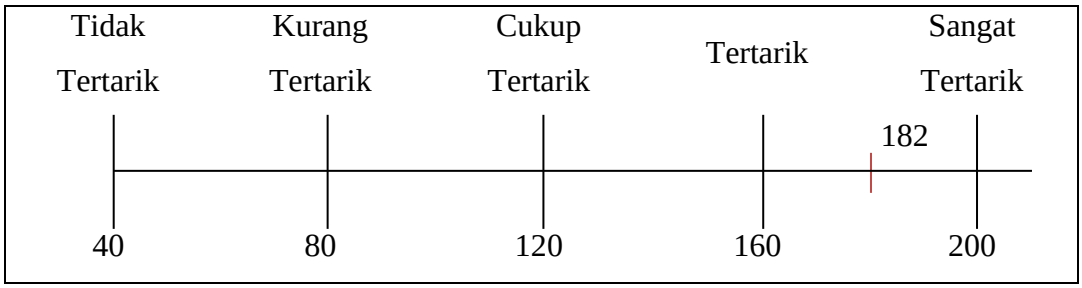
Gambar 4.25. Interpretasi Skor Pertanyaan Ke-6

7. Apakah adik-adik tertarik untuk memainkan *game* ini lagi?

Tabel 4.10. Hasil Pengolahan Data Kuisioner Pertanyaan (7)

JAWABAN	SKOR	RESPONDEN	JUMLAH SKOR (SKOR x JUMLAH RESPONDEN)
Sangat Tertarik	5	27	135
Tertarik	4	8	32
Cukup Tertarik	3	5	15
Kurang Tertarik	2	0	0
Tidak Tertarik	1	0	0
		TOTAL	182

Jika semua responden (40 orang responden) menjawab Sangat Tertarik, maka jumlah skor ideal (kriterium) untuk seluruh item adalah 200 (diperoleh dari 5 x 40). Berdasarkan hasil penelitian, jumlah skor yang diperoleh adalah 182. Jadi berdasarkan data tersebut, maka tingkat persetujuan responden terhadap pertanyaan nomor 7 adalah (182:200) x 100% = 91% dari yang diharapkan (100%). Jadi berdasarkan pengolahan data yang diperoleh dari 40 responden, maka rata-rata 182 terletak pada daerah tertarik. Secara kontinum dapat digambarkan sebagai berikut :



Gambar 4.26. Interpretasi Skor Pertanyaan Ke-1

Berdasarkan hasil pengolahan data diatas, maka dapat ditarik kesimpulan sebagai berikut:

1. Tampilan *game* menarik
2. *Game* mudah digunakan
3. Tingkat kesulitan *game* cukup
4. Responden tidak mengenal keanekaragaman budaya di pulau 5 besar Indonesia yang ada pada *game* sebelum memainkannya
5. Responden kurang mengenal Legenda Sangkuriang sebelum memainkan *game* ini
6. Responden mengetahui keanekaragaman budaya di pulau 5 besar Indonesia dan Legenda Sangkuriang setelah memainkan *game* ini.
7. Responden tertarik untuk memainkan *game* ini kembali.

4.2.2.1 Kesimpulan Pengujian Beta

Berdasarkan hasil persentase jawaban setiap user atau responden terhadap pertanyaan yang diajukan pada pengujian beta dapat ditarik kesimpulan bahwa *game* yang dibangun memiliki tampilan yang menarik, mudah digunakan, tingkat kesulitannya cukup dan anak-anak tertarik untuk memainkan *game* ini kembali. Oleh karena itu bisa disimpulkan *game* ini sebagai sarana pengenalan budaya Indonesia yang lebih menantang dan menyenangkan.