

BAB IV

PENGUJIAN DAN ANALISIS

4.1 Pengujian

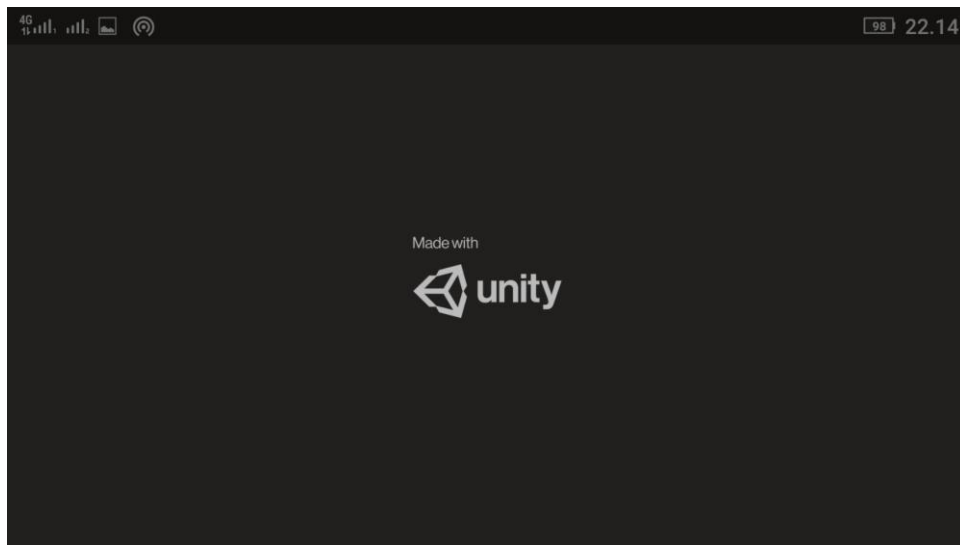
Aplikasi permainan edukasi untuk penerapan algoritma Djiksra ini diuji menggunakan pengujian *Alpha* atau *Alpha Testing* dengan menguji perangkat yang telah dibuat agar sistem dapat berjalan sesuai dengan seharusnya sebelum dirilis kepada pengguna. Pengujian ini juga menggunakan metode *Blackbox* dimana pengujian yang dilakukan setelah aplikasi selesai. Caranya adalah dengan menguji apakah fungsi-fungsi yang terdapat pada aplikasi telah berjalan sesuai dengan yang diinginkan.

Pengujian *Alpha* yang diharapkan dari pengujian ini difokuskan pada pengujian fungsi-fungsi sebagai berikut:

1. Menguji fungsi tombol, narasi, dan animasi pada aplikasi.
2. Menguji kesesuaian tampilan pada aplikasi dengan desain yang diinginkan.

4.1.1 Tampilan Antarmuka Aplikasi

Berikut merupakan tampilan hasil akhir dari antarmuka aplikasi permainan edukasi *maze* menggunakan penerapan algoritma Dijkstra yang telah dibuat. Gambar IV.1 merupakan tampilan dari aplikasi ketika aplikasi pertama kali dibuka. Tampilan ini merupakan tampilan *default* dari aplikasi pembuat gim, yaitu Unity.



Gambar IV.1 Tampilan awal aplikasi

Setelah gambar IV.1 berhasil tayang maka akan muncul gambar IV.2. Gambar IV.2 adalah tampilan dari menu utama dari gim edukasi ini. Pada menu utama ini terdapat tiga buah tombol, yaitu tombol *play* untuk memulai permainan, kemudian tombol *exit* untuk keluar dari permainan, dan yang terakhir tombol *credit* untuk menampilkan *author* dari permainan. Selain itu terdapat *backsound* yang diputar oleh sistem ketika tampilan ini muncul.



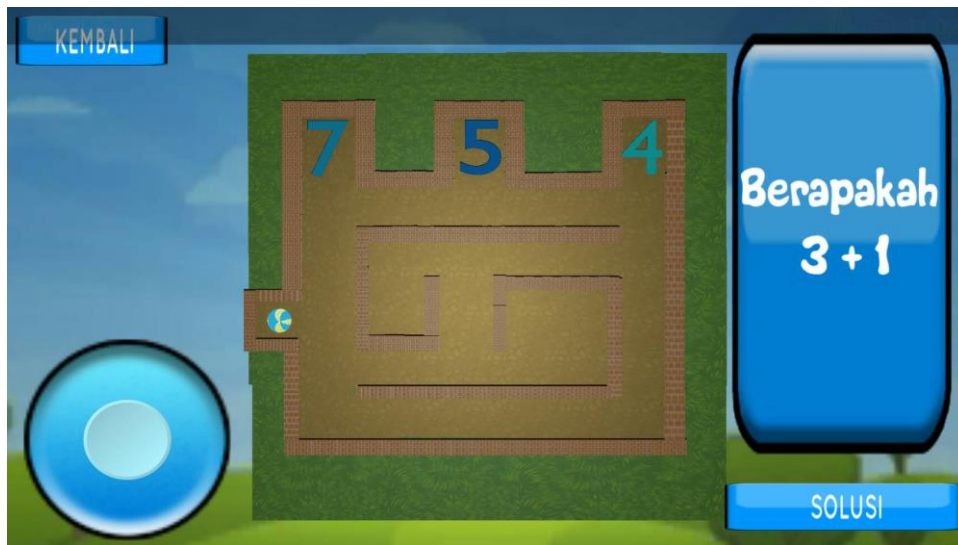
Gambar IV.2 Tampilan Menu Utama

Gambar IV.3 merupakan tampilan ketika menu *credit* ditekan dimana dalam menu ini terdapat tombol kembali di sebelah kiri atas. Selain itu terdapat juga logo Unikom beserta tulisan *created by, produced by, programed by* yang diikuti dengan nama pembuat aplikasi dan diiringi dengan *backsound*.

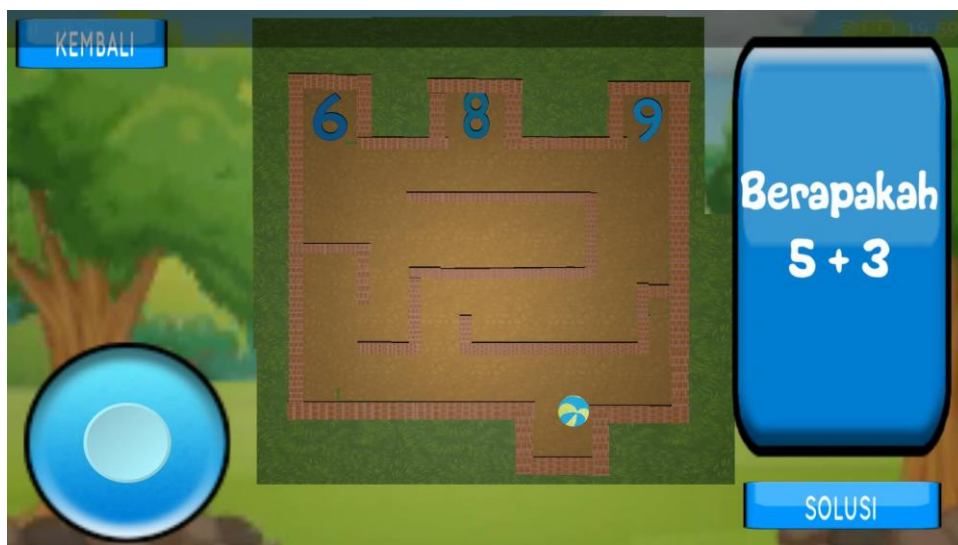


Gambar IV.3 Tampilan menu *credit*

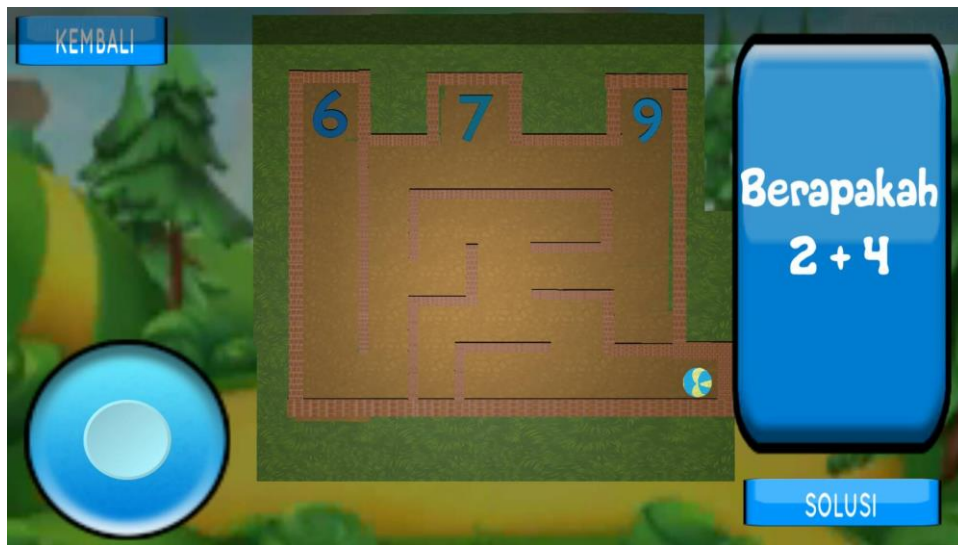
Berikut tampilan dari menu *play* yang langsung masuk ke permainan utama. Permainan ini memiliki tiga level yang harus diselesaikan oleh pemain. Gambar IV.4 adalah tampilan dari level 1, gambar IV.5 tampilan dari level 2, dan gambar IV.6 adalah tampilan dari level 3. Pada setiap level terdapat tombol “kembali” di bagian kanan atas untuk kembali ke menu utama. Di bawah tombol kembali terdapat tombol *joystick* yang digunakan sebagai alat mengontrol gerak aktor yang dikendalikan oleh pemain. Di bagian tengah terdapat labirin sebagai panel permainan utama, kemudian di sebelah kiri terdapat soal yang harus dijawab. Di bawah soal terdapat tombol “solusi”. Fungsinya adalah untuk menampilkan jalur terpendek menuju jawaban yang benar. Penampilan jalur ini diperoleh dari penerapan algoritma Dijkstra.



Gambar IV.4 Tampilan level 1

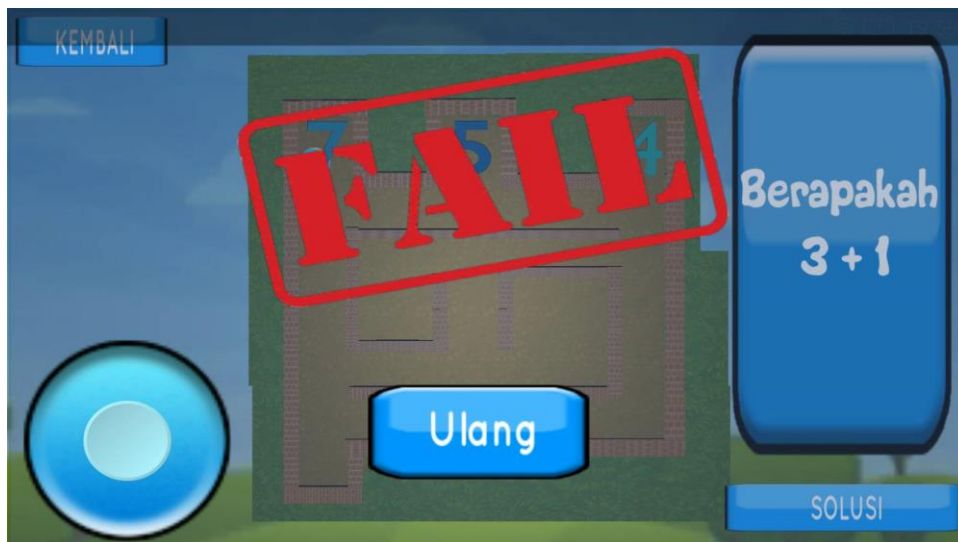


Gambar IV.5 Tampilan level 2



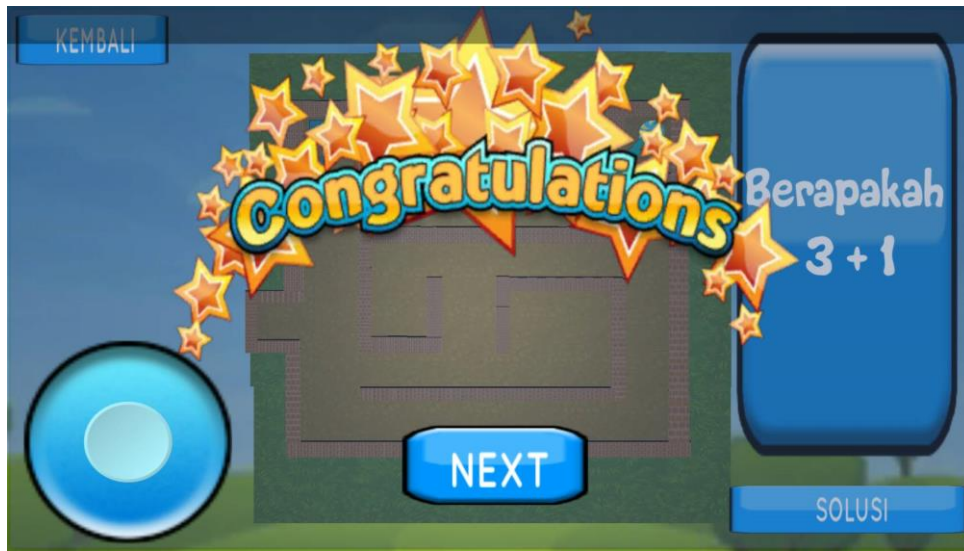
Gambar IV.6 Tampilan level 3

Ketika pemain gagal atau salah dalam menjawab pertanyaan yang diajukan maka akan muncul pesan kegagalan *FAIL* sebagaimana gambar IV.7. Setiap pemain salah menjawab pertanyaan maka pemain harus mengulang kembali dari level paling awal. Di bawah tulisan *FAIL* terdapat tombol “ulang” yang berfungsi untuk mengembalikan pemain ke level yang paling awal.



Gambar IV.7 Tampilan fail

Tampilan dari transisi level yang muncul ketika pemain berhasil menyelesaikan level tersebut dan lanjut ke level berikutnya ditunjukkan pada gambar IV.8. Apabila pemain berhasil menjawab pertanyaan dengan benar maka akan muncul tulisan *CONGRATULATION* sebagai pujian untuk pemain. Di bawah tulisan tersebut terdapat tombol *next* untuk lanjut ke level selanjutnya.



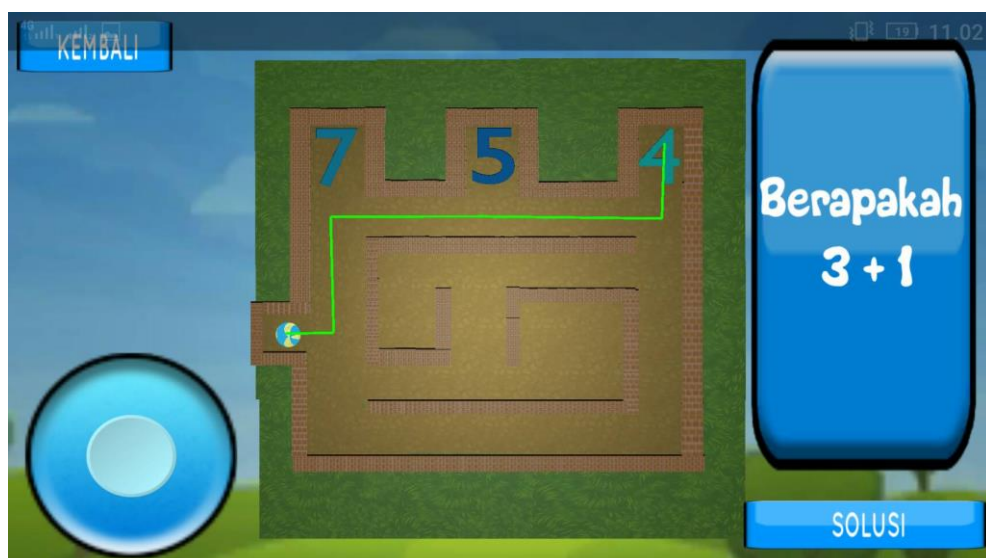
Gambar IV.8 Tampilan transisi level

Apabila seluruh level telah terselesaikan oleh pemain maka pemain akan kembali ke menu utama untuk memulai permainan baru sebagaimana ditunjukkan gambar IV.9. Tampilan dari gambar IV.9 ini hampir sama seperti gambar IV.8 hanya saja fungsi tombol dibawah tulisan *CONGRATULATION* digunakan untuk kembali ke menu utama dengan tulisan tombol main lagi.

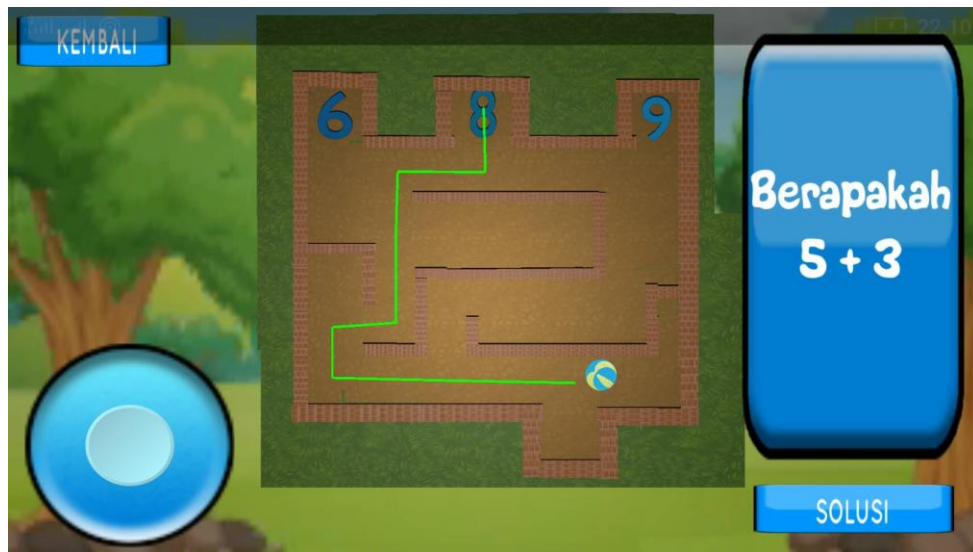


Gambar IV.9 Tampilan seluruh level telah berhasil diselesaikan

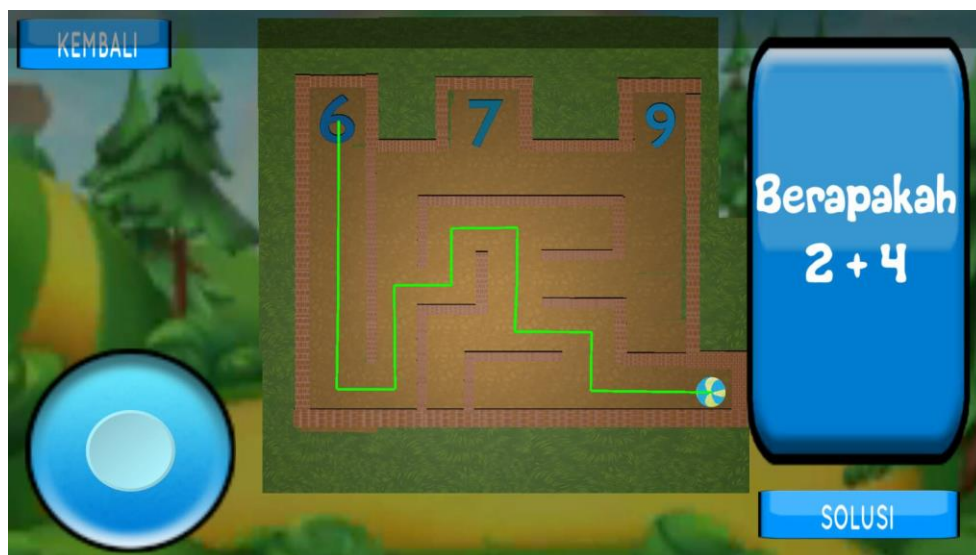
Ketika pemain mengalami kesulitan dalam mencari jalur menuju jawaban yang benar maka pemain dapat memilih tombol “solusi”. Tombol “solusi” ini menerapkan algoritma Dijkstra, dimana algoritma ini digunakan untuk mencari rute terpendek menuju jawaban benar pada setiap level. Adapun antarmuka dari algoritma Dijkstra setiap levelnya yang tertera pada gambar IV.10 sampai gambar IV.12



Gambar IV.10 Algoritma Dijkstra pada level 1



Gambar IV.11 Algoritma Dijkstra pada level 2



Gambar IV.12 Algoritma Dijkstra pada level 3

4.1.2 Komponen Pengujian dengan Metode *Blackbox*

Komponen-komponen pengujian dari aplikasi gim edukasi yang akan diuji dengan metode *Blackbox* terdapat pada tabel IV-1.

Tabel IV-1 Komponen pengujian *Blackbox*

No	Komponen	Aksi yang dilakukan	Hasil yang diharapkan
1	Tampilan awal aplikasi	Membuka aplikasi	Dapat menampilkan aplikasi dan langsung menuju ke menu utama
2	Tampilan menu utama	Menekan tombol play	Dapat menampilkan permainan dimulai dari level 1
		Menekan tombol kembali pada permainan level 1	Dapat kembali ke menu utama
		Menekan tombol solusi pada permainan level 1	Dapat menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.
		Menggerakkan joystick pada permainan level 1	Dapat menggerakkan aktor keatas, kebawah, kekanan, kekiri.
		Menyelesaikan level 1 dengan jawaban benar	Dapat menampilkan transisi level berhasil dan melanjutkan ke level selanjutnya
		Menyelesaikan level 1 dengan jawaban salah	Dapat menampilkan tampilan bahwa jawaban salah dan kembali ke level awal
		Menekan tombol kembali pada permainan level 2	Dapat kembali ke menu utama

		Menekan tombol solusi pada permainan level 2	Dapat menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.
		Menggerakkan joystick pada permainan level 2	Dapat menggerakkan aktor keatas, kebawah, kekanan, kekiri.
		Menyelesaikan level 2 dengan jawaban benar	Dapat menampilkan transisi level berhasil dan melanjutkan ke level selanjutnya
		Menyelesaikan level 2 dengan jawaban salah	Dapat menampilkan tampilan bahwa jawaban salah dan kembali ke level awal
		Menekan tombol kembali pada permainan level 3	Dapat kembali ke menu utama
		Menekan tombol solusi pada permainan level 3	Dapat menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.
		Menggerakkan joystick pada permainan level 3	Dapat menggerakkan aktor keatas, kebawah, kekanan, kekiri.

		Menyelesaikan level 3 dengan jawaban benar	Dapat menampilkan transisi level berhasil dan kembali ke menu utama
		Menyelesaikan level 3 dengan jawaban salah	Dapat menampilkan tampilan bahwa jawaban salah dan kembali ke level awal
		Menekan tombol <i>exit</i>	Dapat keluar dari aplikasi
		Menekan tombol <i>credit</i>	Dapat membuka <i>credit</i> aplikasi
		Menekan tombol kembali pada menu <i>credit</i>	Dapat kembali pada menu utama

4.1.3 Hasil Pengujian *Blackbox*

Berdasarkan dari komponen pengujian menggunakan metode *Blackbox*, maka hasil pengujian yang didapat yaitu sebagai berikut:

1. Pengujian Tampilan Awal Aplikasi

Pengujian dilakukan dengan membuka aplikasi yang telah di install di smartphone. Keterangan dari pengujian ini terdapat pada tabel IV-2

Tabel IV-2 Hasil pengujian Tampilan Awal Aplikasi

Pengujian Tampilan Awal Aplikasi				
No	Butir Uji	Hasil yang diharapkan	Hasil Pengujian	Status Uji
1	Membuka aplikasi	Dapat menampilkan aplikasi dan langsung menuju ke menu utama	Menampilkan aplikasi dan langsung menuju ke menu utama	Berhasil

2. Pengujian Tampilan Menu utama

Pengujian dilakukan dengan membuka setiap tombol yang terdapat pada menu utama tombol tersebut diantaranya adalah tombol “*Play*”, tombol “*Exit*”, dan tombol “*Credit*”. Keterangan dari pengujian ini terdapat pada tabel IV-3

Tabel IV-3 Pengujian Tampilan Menu Utama

Pengujian Tampilan Menu Utama				
No	Butir Uji	Hasil yang diharapkan	Hasil Pengujian	Status Uji
1	Menekan tombol <i>play</i>	Dapat menampilkan permainan dimulai dari level 1	Menampilkan permainan dimulai dari level 1	Berhasil
2	Menekan tombol <i>exit</i>	Dapat keluar dari aplikasi	Keluar dari aplikasi	Berhasil
3	Menekan tombol <i>credit</i>	Dapat membuka <i>credit</i> aplikasi	Membuka <i>credit</i> aplikasi	Berhasil

3. Pengujian Permainan Level 1

Pengujian dilakukan dengan mencoba setiap tombol dan memilih hasil jawaban yang terdapat pada level pertama. Keterangan dari pengujian ini terdapat pada tabel IV-4.

Tabel IV-4 Pengujian Permainan Level 1

Pengujian Permainan Level 1				
No	Butir Uji	Hasil yang diharapkan	Hasil Pengujian	Status Uji
1	Menekan tombol kembali pada permainan level 1	Dapat kembali ke menu utama	Kembali ke menu utama	Berhasil

2	Menekan tombol solusi pada permainan level 1	Dapat menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.	Menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.	Berhasil
3	Menggerakkan joystick pada permainan level 1	Dapat menggerakkan aktor keatas, kebawah, kekanan, kekiri.	Menggerakkan aktor keatas, kebawah, kekanan, kekiri.	Berhasil
4	Menyelesaikan level 1 dengan jawaban benar	Dapat menampilkan transisi level berhasil dan melanjutkan ke level selanjutnya	Menampilkan transisi level berhasil dan melanjutkan ke level selanjutnya	Berhasil
5	Menyelesaikan level 1 dengan jawaban salah	Dapat menampilkan tampilan bahwa jawaban salah dan kembali ke level awal	Menampilkan tampilan bahwa jawaban salah dan kembali ke level awal	Berhasil

4. Pengujian Permainan Level 2

Pengujian dilakukan dengan mencoba setiap tombol dan memilih hasil jawaban yang terdapat pada level pertama. Keterangan dari pengujian ini terdapat pada tabel IV-5

Tabel IV-5 Pengujian Permainan Level 2

Pengujian Permainan Level 2				
No	Butir Uji	Hasil yang diharapkan	Hasil Pengujian	Status Uji
1	Menekan tombol kembali pada permainan level 2	Dapat kembali ke menu utama	Kembali ke menu utama	Berhasil
2	Menekan tombol solusi pada permainan level 2	Dapat menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.	Menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.	Berhasil
3	Menggerakkan joystick pada permainan level 2	Dapat menggerakkan aktor keatas, kebawah, kekanan, kekiri.	Menggerakkan aktor keatas, kebawah, kekanan, kekiri.	Berhasil
4	Menyelesaikan level 2 dengan jawaban benar	Dapat menampilkan transisi level berhasil dan melanjutkan ke level selanjutnya	Menampilkan transisi level berhasil dan melanjutkan ke level selanjutnya	Berhasil

5	Menyelesaikan level 2 dengan jawaban salah	Dapat menampilkan tampilan bahwa jawaban salah dan kembali ke level awal	Menampilkan tampilan bahwa jawaban salah dan kembali ke level awal	Berhasil
---	--	--	--	----------

5. Pengujian Permainan Level 3

Pengujian dilakukan dengan mencoba setiap tombol dan memilih hasil jawaban yang terdapat pada level pertama. Keterangan dari pengujian ini terdapat pada tabel IV-6

Tabel IV-6 Pengujian Permainan Level 3

Pengujian Permainan Level 3				
No	Butir Uji	Hasil yang diharapkan	Hasil Pengujian	Status Uji
1	Menekan tombol kembali pada permainan level 3	Dapat kembali ke menu utama	Kembali ke menu utama	Berhasil
2	Menekan tombol solusi pada permainan level 3	Dapat menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.	Menampilkan algoritma Dijkstra untuk mencari solusi rute terpendek dengan menandai tanda garis hijau pada rute terpendek.	Berhasil

3	Menggerakkan joystick pada permainan level 3	Dapat menggerakkan aktor keatas, kebawah, kekanan, kekiri.	Menggerakkan aktor keatas, kebawah, kekanan, kekiri.	Berhasil
4	Menyelesaikan level 3 dengan jawaban benar	Dapat menampilkan transisi level berhasil dan kembali ke menu utama	Menampilkan transisi level berhasil dan kembali ke menu utama	Berhasil
5	Menyelesaikan level 3 dengan jawaban salah	Dapat menampilkan tampilan bahwa jawaban salah dan kembali ke level awal	Menampilkan tampilan bahwa jawaban salah dan kembali ke level awal	Berhasil

4.1.4 Pengujian *Portability*

Pengujian portability adalah pengujian dimana dilakukan percobaan terhadap beberapa perangkat *smartphone* dengan berbagai versi *android*. Adapun hasil pengujian dapat dilihat pada tabel IV-7.

Tabel IV-7 Hasil Pengujian *Portability*

No	Jenis Perangkat	Jenis Android	Proses Installasi	<i>Running Game</i>
1	Lenovo Vibe K5 plus	Lollipop (5.1.1)	Berhasil	Berjalan tanpa <i>error</i>
2	Lenovo K6 Note	Nougat (7.0)	Berhasil	Berjalan tanpa <i>error</i>

3	Xiaomi Redmi 4 Prime	Marshmallow (6.0)	Berhasil	Berjalan tanpa <i>error</i>
---	-------------------------	----------------------	----------	--------------------------------

4.2 Analisis

Menurut data hasil pengujian yang terdapat pada tabel IV-4 hingga tabel IV-6 didapatkan bahwa aplikasi *Penerapan Algoritma Dijkstra Pada Game Edukasi The Maze Berbasis Android* ini dapat berjalan sesuai dengan semestinya. Sistem dapat menjalankan permainan dengan lancar dan dapat menerapkan algoritma Dijkstra untuk mencari jalur terpendek menuju jawaban yang benar dengan menekan tombol “solusi”.

Pada penelitian ini sistem hanya dapat menampilkan tiga level permainan. Setiap level hanya memiliki satu soal. Tingkat kesulitan soal berbeda pada setiap level. Level satu memiliki tingkat kesulitan yang lebih rendah daripada level dua, dan begitupun selanjutnya.

Gambar IV.10 sampai gambar IV.12 menunjukkan bahwa algoritma Dijkstra berhasil diterapkan pada aplikasi *game* edukasi The Maze berbasis Android dengan mencari rute terpendek menuju jawaban yang benar sesuai dengan soal pada setiap level

4.2.1 Analisis Algoritma

Algoritma dijkstra adalah sebuah algoritma untuk menentukan lintasan terpendek dalam graf tak berarah (atau graf berarah) berbobot tanpa mengenumerasi secara eksplisit semua lintasan yang mungkin. Algoritma ini didasarkan pada sebuah metode yang dikenal sebagai pemrograman dinamik.

Algoritma Dijkstra menentukan lintasan terpendek diantara pasangan-pasangan titik dalam suatu graph. Jika terdapat n titik dalam graph, kita perlu mengeksekusi algoritma nC_2 kali [10]. Jika menggunakan algoritma Dijkstra untuk menentukan jalur terpendek dari suatu graph, maka akan menemukan jalur yang terbaik. Pada saat menentukan jalur yang akan dipilih, algoritma akan menganalisis bobot dari jalur node yang belum terpilih, lalu memilih jalur dengan bobot yang

terkecil. Proses algoritma dijkstra ini karena pada waktu penentuan jalur yang akan dipilih, akan dianalisis bobot dari node yang belum terpilih, lalu dipilih node dengan bobot yang terkecil. Jika terdapat bobot yang lebih kecil melalui node tertentu, maka bobot akan dapat berubah. Algoritma Dijkstra akan berhenti ketika semua node sudah terpilih, dan dengan algoritma Dijkstra ini dapat menemukan jarak terpendek dari seluruh node, tidak hanya untuk node dari asal dan tujuan tertentu saja. Algoritma ini memberi set “belum terjamah” pada setiap node yang belum dilalui, biasanya dengan simbol “ ∞ ” setelah itu algoritma ini memeriksa node awal dan akhir. Node awal diterapkan pada posisi awal aktor sedangkan node akhir diterapkan pada jawaban yang benar dari soal yang diajukan dan memberi node pada setiap jalur yang dapat dilalui sehingga node awal dapat memeriksa node tetangga yang terdekat. Setelah memeriksa node tetangga maka akan menentukan bobot yang paling kecil untuk dipilih. Bobot yang terpilih akan melakukan hal yang sama seperti node awal dengan mempertimbangkan node tetangga yang lebih kecil dan menjumlahkan dengan bobot yang telah didapatkan sebelumnya hingga mencapai node terakhir.

Algoritma ini serupa dengan *greedy best-first search* dimana setiap perpindahan node hanya menjumlahkan dengan node yang telah dilalui sebelumnya. Oleh karena itu, algoritma ini dikatakan lebih mudah dibandingkan dengan algoritma A* karena algoritma A* memiliki rumus matematika

$$f(n) = g(n) + h(n).$$

Dimana:

$f(n)$ = fungsi evaluasi

$g(n)$ = biaya yang sudah dikeluarkan dari keadaan awal sampai keadaan n

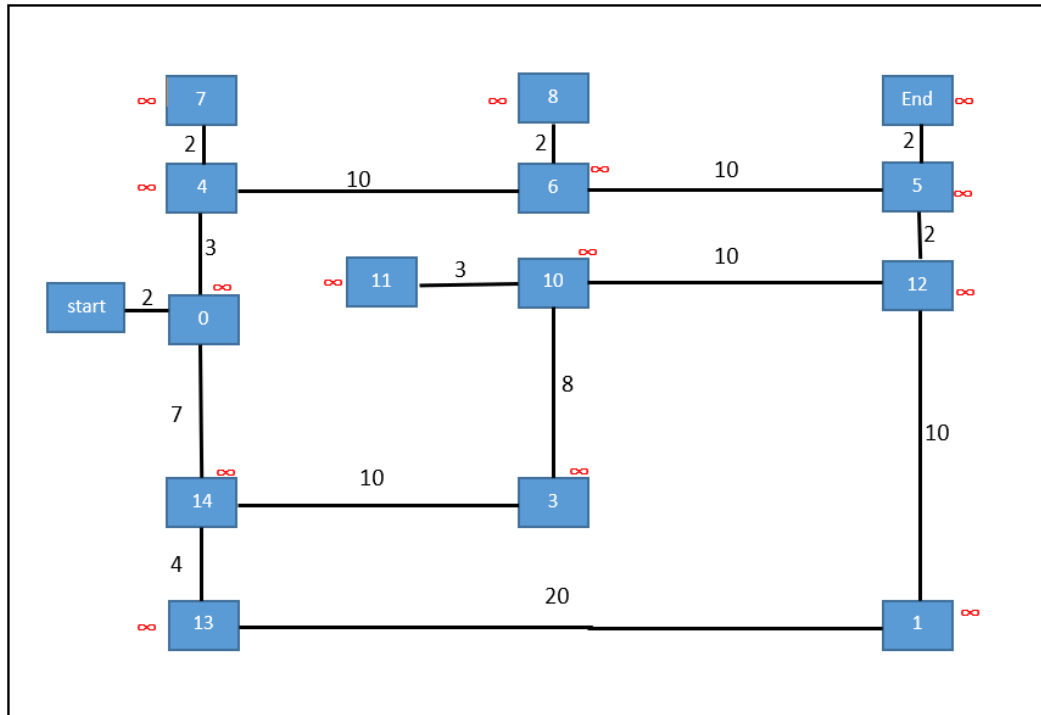
$h(n)$ = estimasi biaya untuk sampai pada suatu tujuan mulai dari n. [8]

Algoritma A* menambahkan jumlah node yang telah dilaluinya dengan menambahkan lagi estimasi jarak dari node terpilih ke node terakhir atau bisa dikatakan garis lurus ke node terakhir.

Dari analisa diatas penulis memilih Algoritma Dijkstra untuk pencarian rute terpendek dengan cara yang lebih efektif.

4.2.1.1 Analisis Dijkstra Level 1

Ketika level 1 dimainkan maka akan muncul node yang tergabung menjadi sebuah graph dari algoritma dijkstra. Graph tersebut tertera pada gambar IV.13



Gambar IV.13 Graph Dijkstra Level 1

Perhitungan Dijkstra yang didapatkan dari graph Dijkstra level 1 terdapat pada tabel IV-8 sebagai berikut:

Tabel IV-8 Perhitungan Dijkstra Level 1

V	S	0	1	3	4	5	6	7	8	10	11	12	13	14	E
S	0 _S	2 _S	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
0		2 _S	∞	∞	5 ₀	∞	∞	∞	∞	∞	∞	∞	∞	9 ₀	∞
4			∞	∞	5 ₀	∞	15 ₄	7 ₄	∞	∞	∞	∞	∞	9 ₀	∞
7			∞	∞	∞	∞	15 ₄	7 ₄	∞	∞	∞	∞	∞	9 ₀	∞
14			∞	19 ₁₄	∞	∞	15 ₄	∞	∞	∞	∞	∞	13 ₁₄	9 ₀	∞
13			33 ₁₃	19 ₁₄	∞	∞	15 ₄	∞	∞	∞	∞	∞	13 ₁₄	∞	∞
6			33 ₁₃	19 ₁₄	∞	25 ₆	15 ₄	∞	17 ₆	∞	∞	∞	∞	∞	∞
8			33 ₁₃	19 ₁₄	∞	25 ₆	∞	∞	17 ₆	∞	∞	∞	∞	∞	∞
3			33 ₁₃	19 ₁₄	∞	25 ₆	∞	∞	∞	27 ₃	∞	∞	∞	∞	∞
5			33 ₁₃	∞	∞	25 ₆	∞	∞	∞	27 ₃	∞	∞	∞	∞	27 ₅
10			33 ₁₃	∞	∞	∞	∞	∞	∞	27 ₃	30 ₁₀	37 ₁₀	∞	∞	27 ₅
E			33 ₁₃	∞	∞	∞	∞	∞	∞	∞	30 ₁₀	37 ₁₀	∞	∞	27 ₅

Keterangan :

S = Start (Node 0)

E = End (Node 15)

V = Titik node saat ini

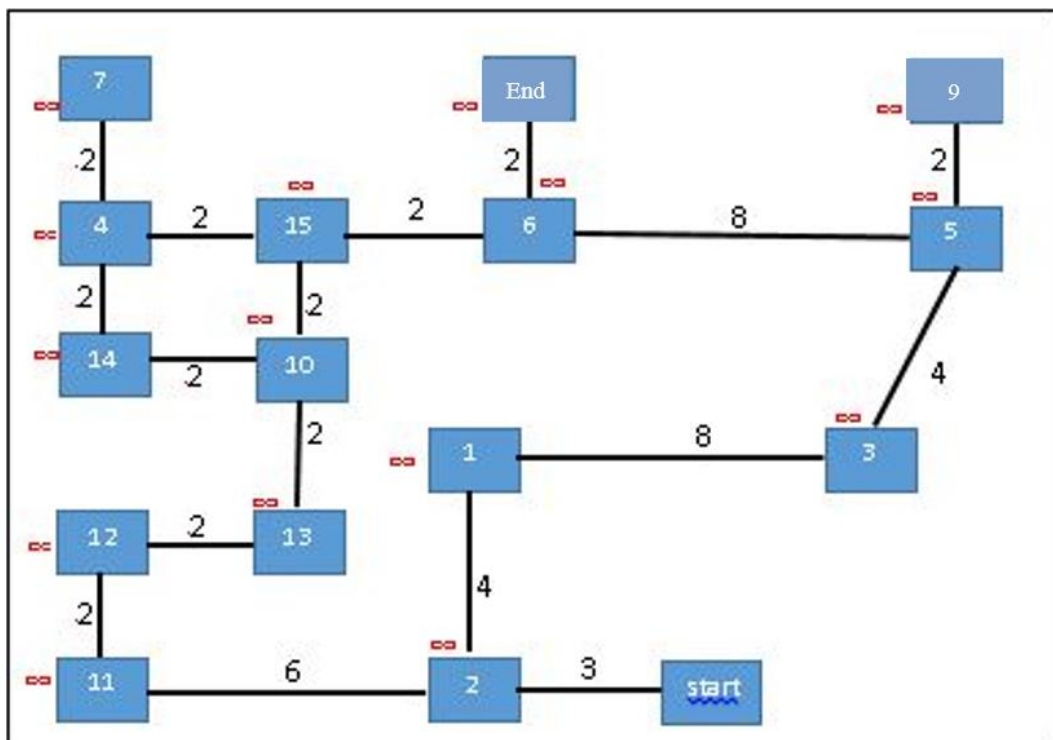
= Nilai yang dipilih untuk penentuan dijkstra

Hasil perhitungan :

S-0-4-6-5-E

4.2.1.2 Analisa Dijkstra Level 2

Selanjutnya, pemain masuk ke level 2 ketika berhasil melewati level 1. Adapun node-node yang terdapat pada graph Dijkstra level 2 digambarkan pada gambar IV.14



Gambar IV.14 Graph Dijkstra Level 2

Perhitungan Dijkstra yang didapatkan dari graph Dijkstra level 2 lebih jelas dijabarkan pada tabel IV-9.

Tabel IV-9 Perhitungan Dijkstra Level 2

V	S	1	2	3	4	5	6	7	9	10	11	12	13	14	15	E
S	0 _S	∞	3 _S	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2		7 ₂	3 _S	∞	∞	∞	∞	∞	∞	∞	9 ₂	∞	∞	∞	∞	∞
1		7 ₂		15 ₁	∞	∞	∞	∞	∞	∞	9 ₂	∞	∞	∞	∞	∞
11				15 ₁	∞	∞	∞	∞	∞	∞	9 ₂	11 ₁₁	∞	∞	∞	∞
12				15 ₁	∞	∞	∞	∞	∞	∞		11 ₁₁	13 ₁₂	∞	∞	∞
13				15 ₁	∞	∞	∞	∞	∞	15 ₁₃			13 ₁₂	∞	∞	∞
10				15 ₁	∞	∞	∞	∞	∞	15 ₁₃				17 ₁₀	17 ₁₀	∞
3				15 ₁	∞	19 ₃	∞	∞	∞					17 ₁₀	17 ₁₀	∞
15					19 ₁₅	19 ₃	19 ₁₅	∞	∞					17 ₁₀	17 ₁₀	∞
14					19 ₁₄	19 ₃	19 ₁₅	∞	∞					17 ₁₀		∞
4					19 ₁₄	19 ₃	19 ₁₅	21 ₄	∞							∞
5						19 ₃	19 ₁₅	21 ₄	21 ₅							∞
6							19 ₁₅	21 ₄	21 ₅							21 ₆
E								21 ₄	21 ₅							21 ₆

Keterangan :

S = Start (Node 0)

E = End (Node 9)

V = Titik node saat ini

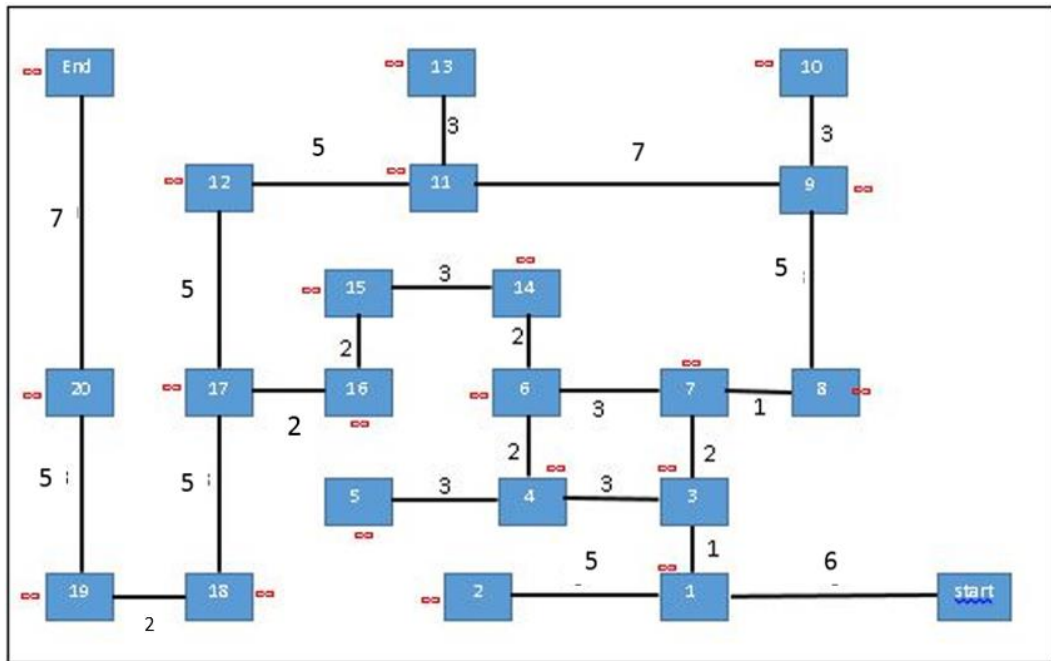
= Nilai yang dipilih untuk penentuan dijkstra

Hasil perhitungan :

S-2-11-12-13-10-15-6-E

4.2.1.3 Analisa Dijkstra Level 3

Labirin terakhir yang harus diselesaikan oleh pemain yaitu labirin pada level 3 dimana terdapat rangkaian node pada gambar IV.15 sebagai berikut:



Gambar IV.15 Graph Dijkstra Level 3

Adapun perhitungan Dijkstra yang didapatkan dari graph Dijkstra level 3 yang tertera pada tabel IV-10 adalah sebagai berikut:

Tabel IV-10 Perhitungan Dijkstra Level 3

V	S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	E
S	0 _s	6 _s	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
1		6 _s	11 ₁	7 ₁	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
3			11 ₁	7 ₁	10 ₃	∞	∞	9 ₃	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
7			11 ₁		10 ₃	∞	12 ₄	9 ₃	10 ₇	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
4			11 ₁		10 ₃	13 ₄	12 ₄		10 ₇	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
8			11 ₁			13 ₄	12 ₄		10 ₇	15 ₈	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
2			11 ₁			13 ₄	12 ₄			15 ₈	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
6						13 ₄	12 ₄			15 ₈	∞	∞	∞	∞	14 ₆	∞	∞	∞	∞	∞	∞	∞
5						13 ₄				15 ₈	∞	∞	∞	∞	14 ₆	∞	∞	∞	∞	∞	∞	∞
14										15 ₈	∞	∞	∞	∞	14 ₆	∞	∞	∞	∞	∞	∞	∞
9										15 ₈	18 ₉	22 ₉	∞	∞	∞	17 ₁₄	∞	∞	∞	∞	∞	∞
15										18 ₉	22 ₉	22 ₉	∞	∞	∞	17 ₁₄	19 ₁₅	∞	∞	∞	∞	∞
10										18 ₉	22 ₉	22 ₉	∞	∞	∞	19 ₁₅	19 ₁₅	∞	∞	∞	∞	∞
16											22 ₉	∞	∞	∞	∞	19 ₁₅	∞	∞	∞	∞	∞	∞
17											22 ₉	27 ₁₇	27 ₁₇	∞	∞	∞	21 ₁₆	27 ₁₇	27 ₁₇	∞	∞	∞
11											22 ₉	27 ₁₇	25 ₁₁	∞	∞	∞	∞	27 ₁₇	27 ₁₇	∞	∞	∞
13												27 ₁₇	25 ₁₁	∞	∞	∞	∞	27 ₁₇	27 ₁₇	∞	∞	∞
18												27 ₁₇						27 ₁₇	29 ₁₈	29 ₁₈	∞	∞
12												27 ₁₇							27 ₁₇	29 ₁₈	∞	∞
19																			27 ₁₇	29 ₁₈	34 ₁₉	∞
20																				29 ₁₈	34 ₁₉	41 ₂₀
E																						41 ₂₀

Keterangan :

S = Start (Node 0)

E = End (Node 21)

V = Titik node saat ini

= Nilai yang dipilih untuk penentuan dijkstra

Hasil perhitungan :

S-1-3-4-6-14-15-16-17-18-19-20-E