BAB 4
IMPLEMENTASI DAN PENGUJIAN

Pada bab ini akan dilakukan implementasi dan pengujian terhadap aplikasi yang dibangun. Tahapan ini dilakukan setelah analisis dan perancangan selesai dilakukan dan selanjutnya akan diimplementasikan kedalam Bahasa pemograman. Aplikasi yang dibangun akan diimplementasikan untuk mengetahui apakah sistem tersebut berjalan sesuai dengan tujuannya atau tidak.

1.1. Implementasi

Tujuan implementasi adalah untuk menerapkan perancangan yang telah dilakukan terhadap sistem sehingga user dapat memberi masukan demi berkembangnya sistem yang telah dibangun sebagai simulasi penerapan algoritma whca* pada pencarian jalur terpendek dengan banyak agen. Perangkat keras dan perangkat lunak yang digunakan untuk melakukan implementasi dan pengujian aplikasi yaitu sebagai berikut:

a. Perangkat keras

Tabel 4.1. Perangkat Keras yang digunakan

Komputer	Spesifikasi
Prosesor	Intel(R) Core(TM) i5
Layar	Resolusi 1368 X 768
Monitor	LCD 14 inch
Harddisk	320 GB
Memori	4 GB

b. Perangkat lunak

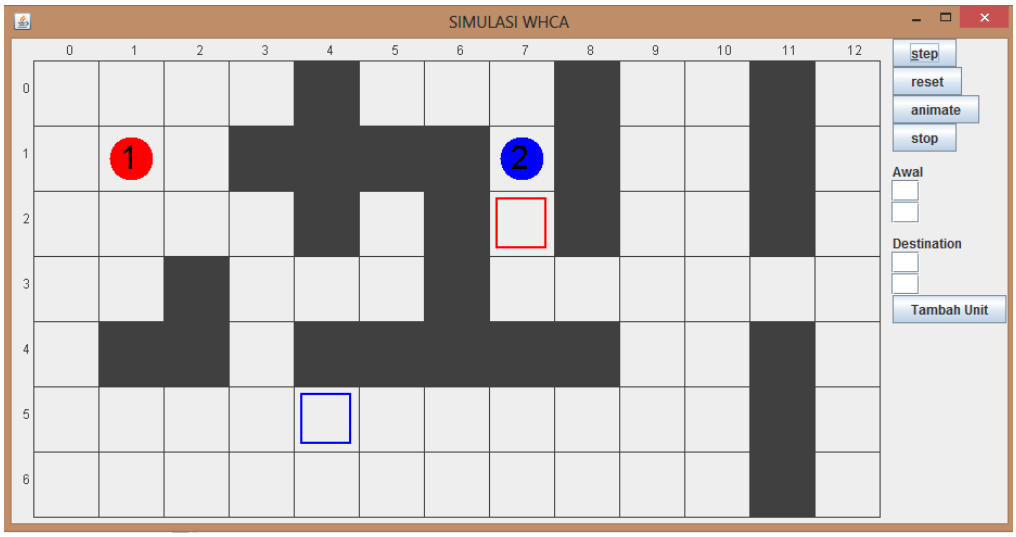
Perangkat lunak yang digunakan untuk mengimplementasikan sistem yaitu sebagai berikut:

Tabel 4.2. Perangkat Lunak yang digunakan

Komputer	Spesifikasi
Sistem Operasi	Microsoft Window 8
Bahasa Pemograman	Java
IDE	Eclipse Luna 4.4.1

1.1.1. Implementasi Antarmuka

Implementasi antarmuka untuk perancangan aplikasi tampilan utama yang dibangun dapat dilihat pada gambar 4.1.

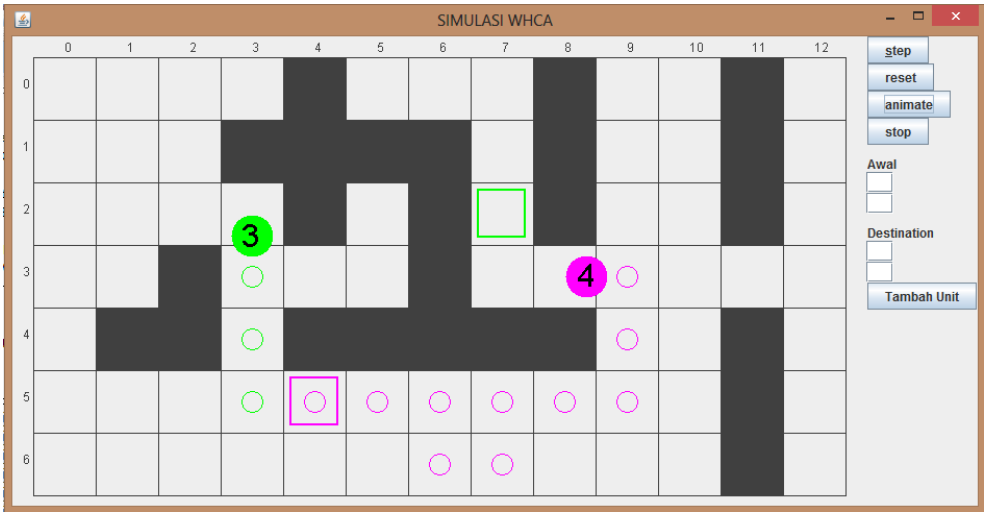


Gambar 4.1. Implementasi antarmuka

Di dalam aplikasi ini terdapat beberapa lima fungsionalitas yaitu tombol *step,* tombol *reset,* tombol *animate,* tombol stop, tombol tambah unit dan tambah rintangan.

1.1.2. Implementasi Contoh Kasus Jalur WHCA*

Salah satu contoh implementasi simulasi yang akan digunakan untuk menemukan jalur yaitu dengan menggunakan penghalang dan adanya kemungkinan tabrakan pada titik tertentu. Pertama setelah pengguna menempatkan titik awal dan titik akhir pada yang diinginkan dan sudah menempatkan beberapa penghalang pada map maka pencarian rute terpendek dapat dilakukan dengan menekan tombol step untuk melihat perlangkah dalam pencarian atau menekan tombol animate untuk melihat pencarian secara animasi. Implementasi jalur dengan algoritma WHCA* dengan menggunakan animasi dapat terlihat pada gambar 4.2 dengan rute reservasi pada iterasi pertama.



Gambar 4.2. Implementasi pencarian dengan menggunakan animasi

1.2. Pengujian

Pengujian pada sistem yang dibangun dilakukan dengan pengujian alpha (fungsional).

1.2.1. Pengujian blackbox

Metode yang digunakan dalam pengujian adalah metode blackbox yang fokus pada persyaratan fungsional dari perangkat lunak yang dibangun.

1.2.2. Rencana pengujian program simulasi

Rencana pengujian yang dilakukan pada perangkat lunak dengan menggunakan metode blakcbox dapat dilihat pada tabel 4.3.

Tabel 4.3. Rencana Pengujian program simulasi

Item Uji	Detail Pengujian
Pengaturan titik awal dan titik tujuan	Titik awal
	Titik tujuan
Penentuan penghalang	Posisi penghalang
Pencarian rute	Tombol *Step*
	Tombol *Animate*
	Tombol *Stop*
	Tombol *Reset*
Penentuan *Grid*	Nilai Baris
	Nilai Kolom

1.2.3. Kasus dan hasil pengujian

Berdasarkan rencana pengujian, maka dapat dilakukan pengujian alpha dengan blackbox pada simulasi pencarian jalur terpendek menggunakan algoritma WHCA* dengan banyak agen yang dijelaskan pada tabel 4.4.

Tabel 4.4. Kasus dan hasil pengujian (data normal)

Kasus dan hasil pengujian (data normal)			
Aksi	**Yang diharapkan**	**Pengamatan**	**Kesimpulan**
Memasukan titik awal	Menampilkan titik awal pada map	Map menampilkan titik awal	Diterima
Memasukan titik tujuan	Menampilkan titik tujuan pada map	Map menampilkan titik tujuan	Diterima
Memasukan data halangan	Menampilkan penghalang tembok pada map	Map menampilkan penghalang	Diterima
Menekan tombol *step*	Menampilkan rute reservasi pertama setiap unit pada map	Map menampilkan rute reservasi pertama	Diterima
Menekan tombol *Animate*	Menampilkan setiap unit bergerak dari titik awal menuju titik akhir	Sistem melakukan animasi perjalanan dari titik awal hingga titik tujuan	Diterima
Menekan tombol *Stop*	Menampilkan setiap unit pada posisi terakhir	Map menampilkan setiap unit pada posisi terakhir	Diterima
Menekan tombol *reset*	Menampilkan map dengan tanpa halangan dan unit	Map menghapus data halangan dan unit	Diterima
Memasukan data nilai baris Contoh: 13	Dapat terisi *number* untuk masukan data nilai baris	Jumlah baris pada map sesuai dengan yang diinputkan	Diterima

Tabel 4.5. Kasus dan hasil uji (data normal) Lanjutan

Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data nilai kolom. Contoh: 10	Dapat terisi *number* untuk masukan data nilai kolom	Jumlah kolom pada map sesuai dengan yang diinputkan	Diterima
Klik tombol Grid Baru	Menampilkan *grid* yang sesuai dengan masukan nilai baris dan nilai kolom	Menampilkan map dengan jumlah baris dan kolom sesuai inputan	Diterima

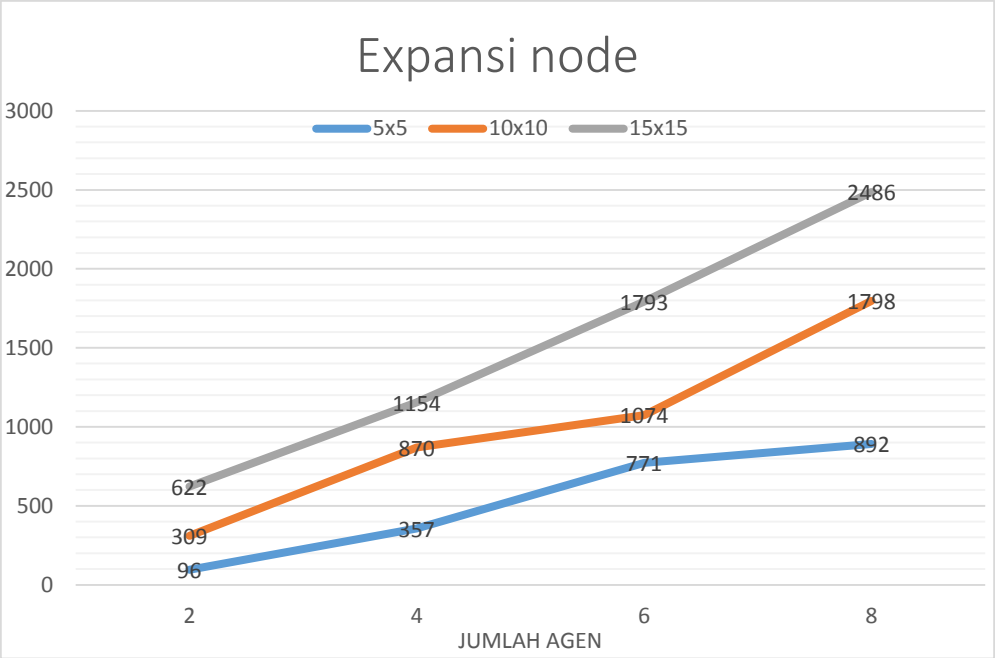
Tabel 4.6. Kasus dan hasil uji (data tidak normal)

Masukan	Yang diharapkan	Pengamatan	Kesimpulan
Memasukan data titik awal pada posisi halangan	Muncul pesan konfirmasi bahwa titik awal tidak dapat diinput	Hasil sesuai dengan yang diharapkan	Diterima
Memasukan data titik tujuan pada posisi halangan	Muncul pesan konfirmasi bahwa titik tujuan tidak dapat diinput	Hasil sesuai dengan yang diharapkan	Diterima
Memasukan data nilai kolom atau nilai baris melebihi batas grid	Muncul pesan konfirmasi bahwa data melebihi batas grid	Hasil sesuai dengan yang diharapkan	Diterima
Memasukan titik awal dan titik tujuan pada posisi yang sama	Muncul pesan bahwa titik tujuan harus berbeda dengan titik awal	Hasil sesuai dengan yang diharapkan	Diterima
Memasukan titik awal pada titik awal unit lain pada waktu yang sama	Muncul pesan bahwa titik awal sudah digunakan oleh unit lain	Hasil sesuai dengan yang diharapkan	Diterima
Memasukan titik tujuan pada titik tujuan unit lain pada waktu yang sama	Muncul pesan bahwa titik tujuan sudah digunakan oleh unit lain	Hasil sesuai dengan yang diharapkan	Diterima

1.2.4. Pengujian Performansi

Performasi yang akan diuji pada aplikasi yang telah dibangun yaitu waktu tempuh dari posisi awal hingga posisi akhir setiap unit dan jumlah simpul yang diperiksa pada setiap siklus.

Performansi yang akan diuji pada aplikasi yang telah dibangun yaitu simpul yang diperiksa dan waktu yang ditempuh yang dihasilkan dengan Implementasi ordo minimal 5 x 5 tanpa penghalang dengan jumlah agen dari 2 hingga 8.

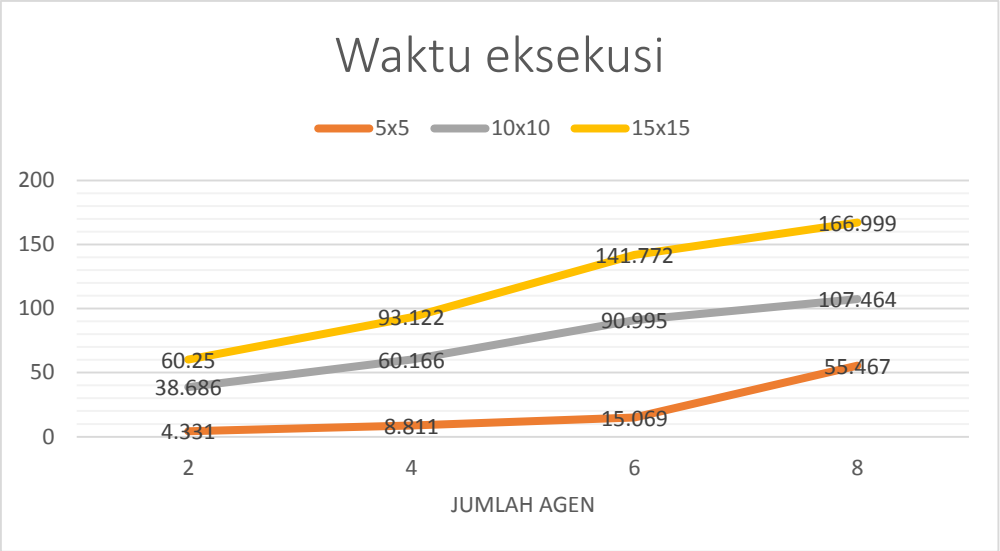


Gambar 4.3. Hasil pengujian ekspansi node

Hasil pengujian ekspansi node pada ordo 5x5, 10x10 dan 15x15 tanpa penghalang dengan jumlah agen dari 2 – 8 dapat dilihat pada tabel 4.7.

Tabel 4.7. Tabel hasil pengujian ekspansi node

Ordo \ jumlah unit	2	4	6	8
5X5	96 ms	357 ms	771 ms	892ms
10X10	309 ms	870 ms	1074 ms	1798 ms
15X15	622 ms	1154 ms	1793 ms	2486 ms



Gambar 4.4. Hasil pengujian waktu eksekusi

Hasil pengujian waktu eksekusi pada ordo 5x5, 10x10 dan 15x15 tanpa penghalang dengan jumlah agen dari 2 – 8 dapat dilihat pada tabel 4.8.

Tabel 4.8. Tabel hasil pengujian hasil waktu eksekusi

Ordo \ jumlah unit	2	4	6	8
5X5	4.331 ms	8.811 ms	15.069 ms	55.467 ms
10X10	38.686 ms	60.166 ms	90.995 ms	107.464 ms
15X15	60.25 ms	93.122 ms	141.772 ms	166.999 ms

1.2.5. Kesimpulan Pengujian Blackbox

Hasil pengujian dari pengujian blackbox yang telah dilakukan, menunjukkan bahwa aplikasi yang dibangun sudah memenuhi persyaratan fungsional, akan tetapi pada prosesnya masih memungkinkan untuk terjadinya kesalahan, namun frekuensi kesalahan masih relatif kecil. Secara fungsional sistem yang telah dibangun sudah dapat menghasilkan keluaran yang diharapkan. Jalan yang dihasilkan merupakan jalan terpendek karena simpul yang diperiksa relatif banyak dan memerlukan waktu pencarian yang relatif lama pula. Semakin banyak agen maka semakin banyak simpul yang diperiksa