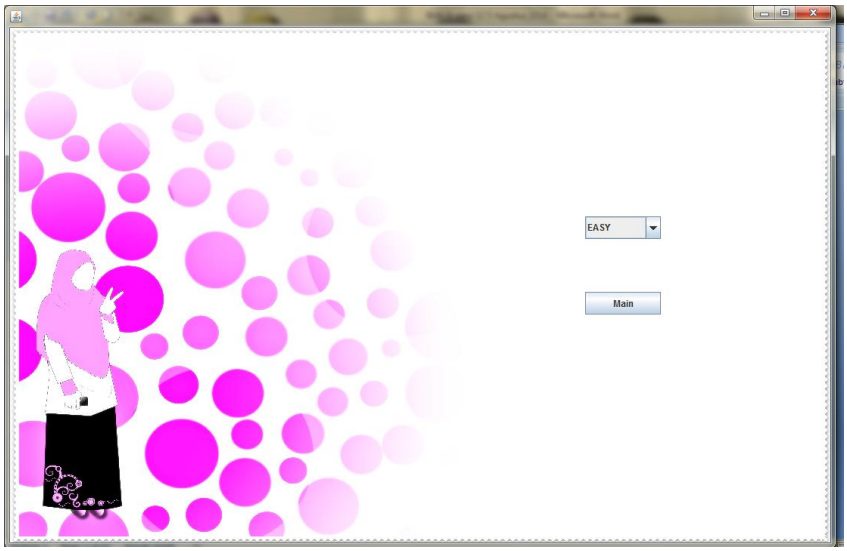
BAB 4
IMPLEMENTASI DAN PENGUJIAN

4.1 Implementasi Antarmuka

Tahap implementasi antarmuka merupakan tahap pembangunan *game* untuk menerapkan algoritma IDA* dan IDA* dengan SNC pada *game* labirin. Adapun implementasi antarmuka pada aplikasi *game* labirin adalah sebagai berikut :

1. Antarmuka Memilih Map

   Antarmuka memilih map merupakan tampilan awal ketika pemain membuka aplikasi permainan. Pilihan kerumitan labirin dalam *game* labirin terdiri dari *Easy*, *Medium* dan *Hard*. Tampilan dari antarmuka tingkat kerumitan labirin dapat dilihat pada Gambar 4.1.

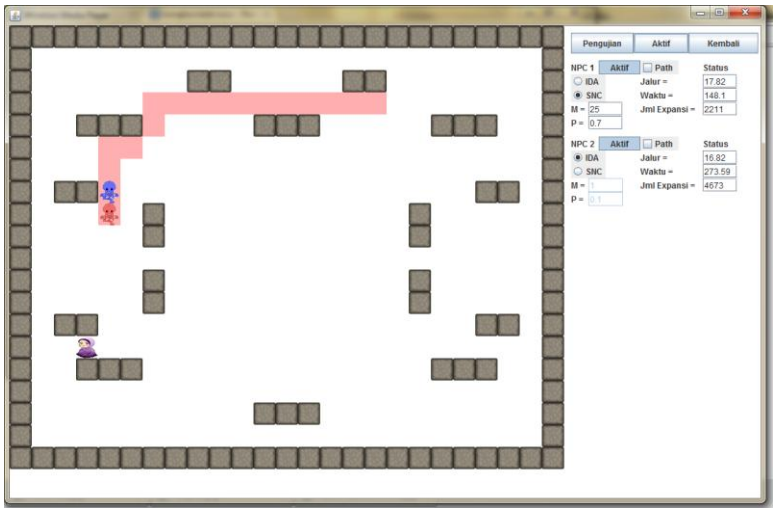


Gambar 4.1 Tampilan Antarmuka Memilih Map

2. Antarmuka Map *Easy*

   Antarmuka map *easy* merupakan tampilan pengujian ketika pemain dan musuh berada dalam *game* labirin pada map *easy* yang sudah ditentukan posisinya. Musuh yang dapat diaktifkan berjumlah 2 musuh, dimana masing-masing musuh harus diterapkan sebuah algoritma terlebih dahulu apakah algoritma IDA* atau IDA* dengan SNC. Jika ingin menerapkan algoritma IDA* dengan SNC, pemain harus menginputkan parameter *M* dan *p* yang akan

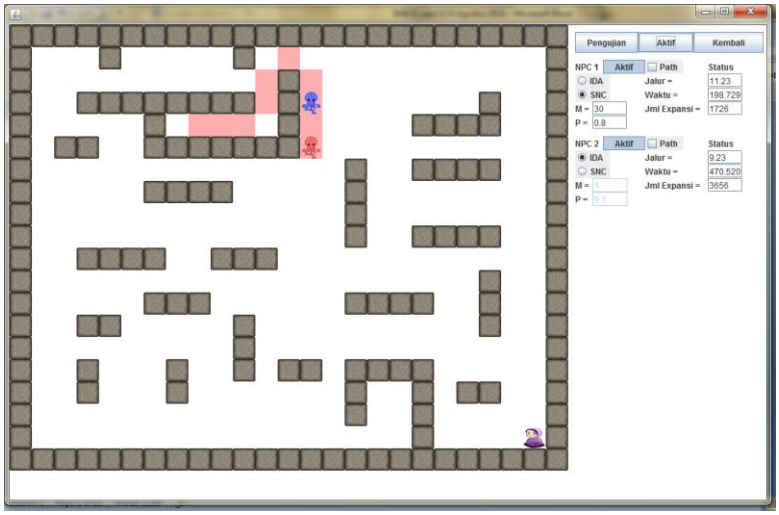
digunakan. Tampilan dari antarmuka bermain pada map *easy* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Tampilan Antarmuka Map *Easy*

3. Antarmuka Map *Medium*

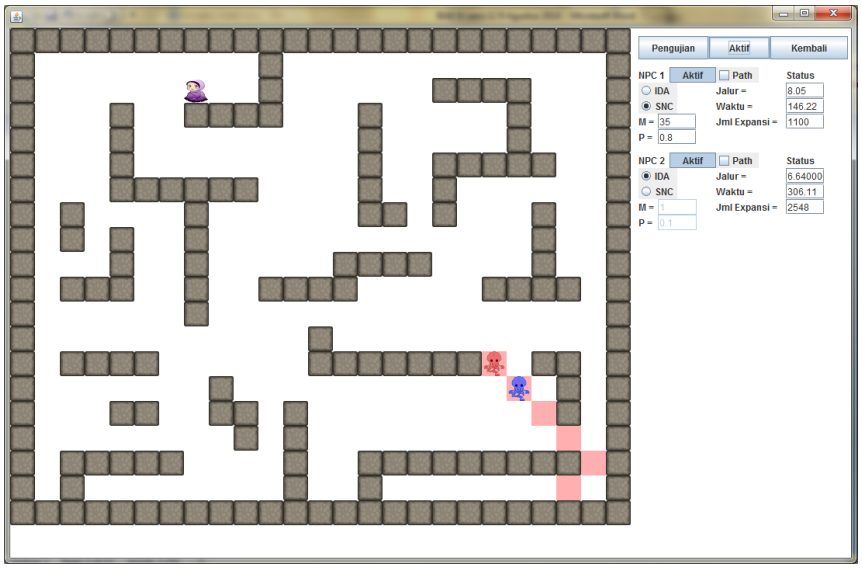
Antarmuka map *medium* merupakan tampilan pengujian ketika pemain dan musuh berada dalam *game* labirin pada map *medium* yang sudah ditentukan posisinya. Musuh yang dapat diaktifkan berjumlah 2 musuh, dimana masing-masing musuh harus diterapkan sebuah algoritma terlebih dahulu apakah algoritma IDA* atau IDA* dengan SNC. Jika ingin menerapkan algoritma IDA* dengan SNC, pemain harus menginputkan parameter *M* dan *p* yang akan digunakan. Tampilan dari antarmuka bermain pada map *medium* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tampilan Antarmuka Map *Medium*

4. Antarmuka Map *Hard*

Antarmuka map hard merupakan tampilan pengujian ketika pemain dan musuh berada dalam game labirin pada map hard yang sudah ditentukan posisinya. Musuh yang dapat diaktifkan berjumlah 2 musuh, dimana masing-masing musuh harus diterapkan sebuah algoritma terlebih dahulu apakah algoritma IDA* atau IDA* dengan SNC. Jika ingin menerapkan algoritma IDA* dengan SNC, pemain harus menginputkan parameter M dan p yang akan digunakan. Tampilan dari antarmuka bermain pada map hard dapat dilihat pada Gambar 4.4.



Gambar 4.4 Tampilan Antarmuka Map *Hard*

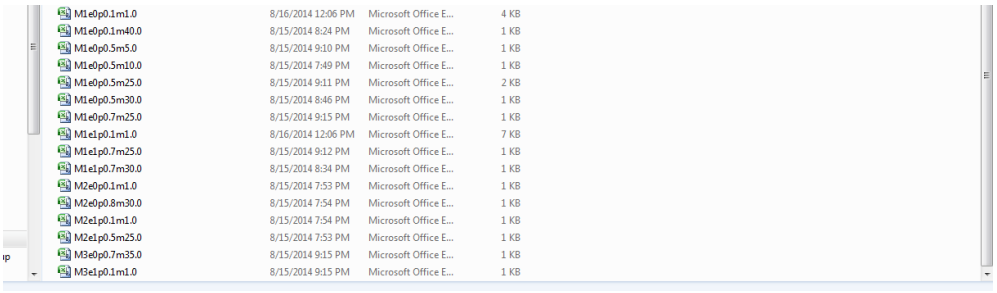
4.2 Pengujian

Pengujian pada *game* labirin dilakukan melalui tahap pengujian parameter algoritma IDA* SNC dan pengujian parameter aplikasi.

4.2.1 Pengujian Parameter IDA* SNC

Dalam pengujian parameter algoritma IDA* SNC dilakukan untuk mengetahui besar *cache* (*M*) dan probabilitas (*p*) terbaik agar memaksimalkan performa dari IDA* SNC. Karena pada awalnya inputan parameter untuk IDA* SNC belum diketahui inputan yang tepat. Pengujian parameter IDA* SNC akan dilakukan pada map *easy*, *medium* dan *hard* dengan posisi pemain, musuh dan tembok yang sudah ditentukan. Banyaknya pengujian yang dilakukan untuk setiap

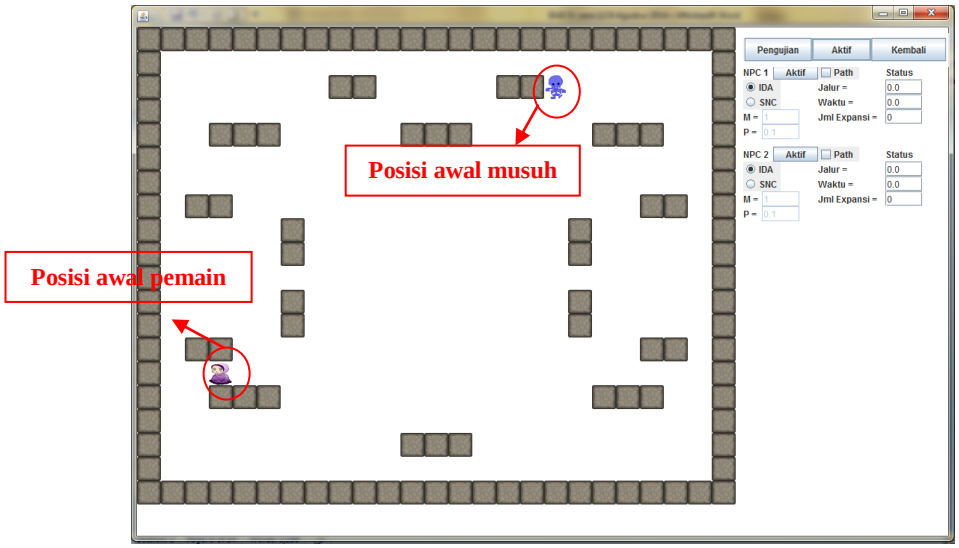
map yaitu berjumlah 900 kali percobaan dengan mengubah inputan nilai *M* dari 15 s/d 60. Sedangkan nilai *p* sendiri diset 0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; dan 0.9 (pengujian terlampir). Gambar 4.5 merupakan hasil pengujian yang telah dilakukan dan langsung tersimpan di dalam folder penyimpanan program *game* labirin dalam bentuk file .**csv** kemudian penulis buat dalam bentuk Tabel yang sudah terlampir pada Lampiran B, Lampiran C dan Lampiran D.



Gambar 4.5 Hasil Pengujian dalam Bentuk File .csv

1. Map *Easy*

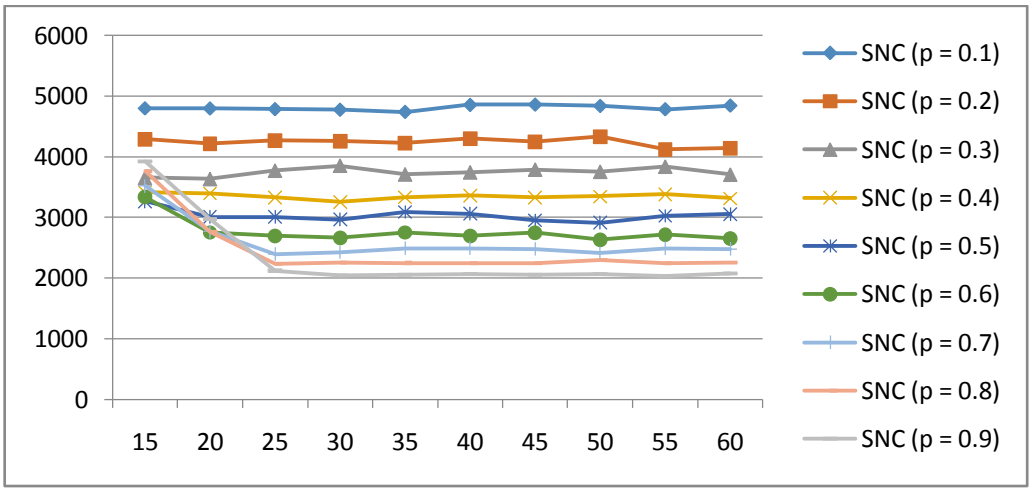
Pengujian pada map *easy* dilakukan dengan percobaan untuk 1 NPC dengan algoritma IDA* SNC dan mengubah nilai *M* dari 15 s/d 60. Sedangkan nilai *p* sendiri diset 0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; dan 0.9. Adapun posisi awal NPC dan pemain pada map *easy* dapat dilihat pada Gambar 4.6. Percobaan yang dilakukan mengasumsikan posisi awal NPC dan pemain dalam keadaan diam. Detail pengujian terdapat pada Lampiran B. Hasil pengujian kemudian dirata-rata dan dibuat dalam Tabel 4.1 dan Gambar 4.7.



Gambar 4.6 Posisi NPC dan Pemain pada Map *Easy*

Tabel 4.1 Hasil Pengujian Parameter IDA* SNC Terhadap Ekspansi Simpul pada Map *Easy*

NPC Map *Easy*									
p / M	(0.1)	(0.2)	(0.3)	(0.4)	(0.5)	(0.6)	(0.7)	(0.8)	(0.9)
15	4796	4291	3653	3416	3264	3336	3520	3761	3925
20	4794	4220	3635	3397	3006	2752	2769	2758	2971
25	4783	4268	3771	3326	3005	2698	2394	2235	2121
30	4773	4261	3852	3253	2962	2666	2422	2258	2046
35	4737	4224	3710	3333	3087	2752	2489	2249	2056
40	4853	4298	3743	3361	3058	2697	2487	2245	2063
45	4858	4247	3788	3330	2949	2745	2479	2242	2053
50	4836	4332	3753	3346	2911	2631	2411	2295	2063
55	4781	4121	3836	3382	3024	2719	2490	2248	2032
60	4842	4144	3706	3314	3054	2655	2479	2251	2076



Gambar 4.7 Grafik Pengujian Parameter IDA* SNC Terhadap Ekspansi Simpul pada Map *Easy*

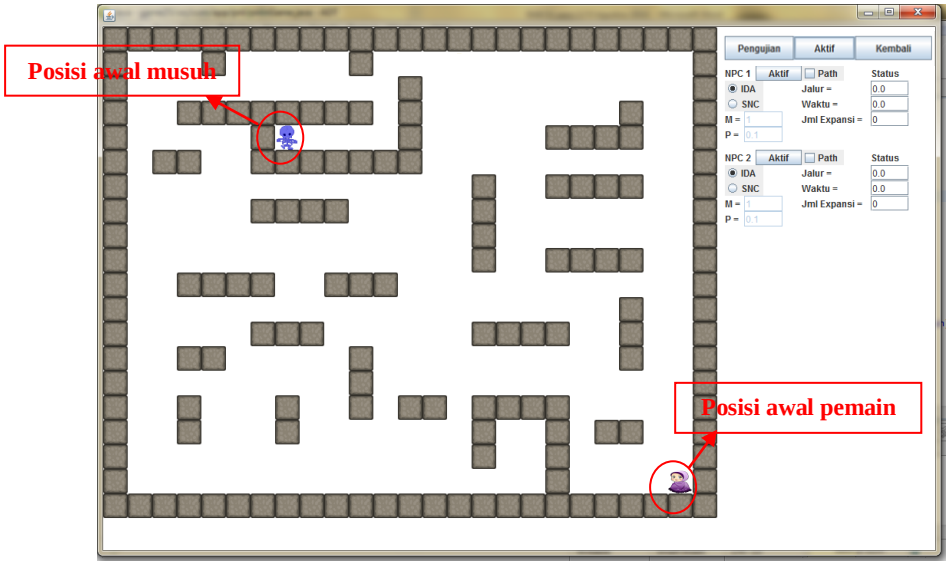
Pada Tabel 4.1 dan Gambar 4.7 terlihat dari pengujian untuk map *easy* terlihat semakin tinggi nilai *cache* (*M*), semakin menurun rata-rata jumlah simpul yang diekspansi. Hal ini dikarenakan semakin besar ukuran *cache* maka akan lebih banyak simpul yang sering dikunjungi yang bisa disimpan. Ini mengakibatkan jumlah total ekspansi simpul selama proses pencarian menjadi berkurang.

Dari hasil percobaan terlihat pada nilai *cache* dan probabilitas tertentu (*M* = 25, *p* = 0.7) menghasilkan rata-rata jumlah ekspansi simpul yang diperiksa semakin stabil dikarenakan besar *cache* mulai 25 dan probabilitas 0.7 sudah cukup

untuk menyimpan sebagian simpul yang sering dikunjungi. Penambahan ukuran *cache* tidak akan menurunkan jumlah ekspansi simpul secara signifikan lagi. Jika *p* diset lebih rendah dari 0.7 maka ekspansi simpul juga cenderung meningkat karena kemungkinan simpul yang sering dikunjungi untuk disimpan dalam *cache* menjadi lebih kecil.

2. Map *Medium*

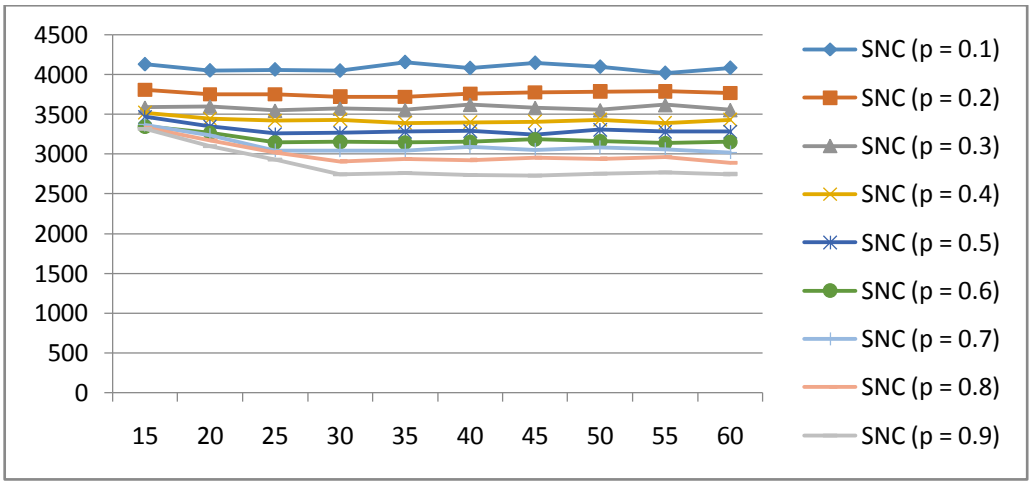
Pengujian pada map *medium* dilakukan dengan percobaan untuk 1 NPC dengan algoritma IDA* SNC dan mengubah nilai *M* dari 15 s/d 60. Sedangkan nilai *p* sendiri diset 0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; dan 0.9. Adapun posisi awal NPC dan pemain pada map *medium* dapat dilihat pada Gambar 4.8. Percobaan yang dilakukan mengasumsikan posisi awal NPC dan pemain dalam keadaan diam. Detail pengujian terdapat pada Lampiran C. Hasil pengujian kemudian dirata-rata dan dibuat dalam Tabel 4.2 dan Gambar 4.9.



Gambar 4.8 Posisi NPC dan Pemain pada Map *Medium*

Tabel 4.2 Hasil Pengujian Parameter IDA* SNC Terhadap Ekspansi Simpul pada Map *Medium*

NPC Map *Medium*									
p / M	(0.1)	(0.2)	(0.3)	(0.4)	(0.5)	(0.6)	(0.7)	(0.8)	(0.9)
15	4130	3810	3587	3523	3470	3350	3370	3333	3316
20	4053	3756	3602	3445	3347	3269	3240	3171	3101
25	4065	3756	3549	3421	3257	3146	3051	3024	2931
30	4051	3724	3576	3431	3271	3157	3052	2913	2747
35	4157	3722	3556	3391	3286	3149	3052	2943	2764
40	4083	3760	3623	3400	3292	3156	3095	2929	2741
45	4148	3776	3584	3411	3244	3186	3060	2959	2732
50	4099	3790	3554	3435	3307	3164	3087	2944	2756
55	4023	3791	3619	3391	3286	3138	3061	2964	2770
60	4086	3769	3557	3433	3287	3157	3024	2893	2752



Gambar 4.9 Grafik Pengujian Parameter IDA* SNC Terhadap Ekspansi Simpul pada Map *Medium*

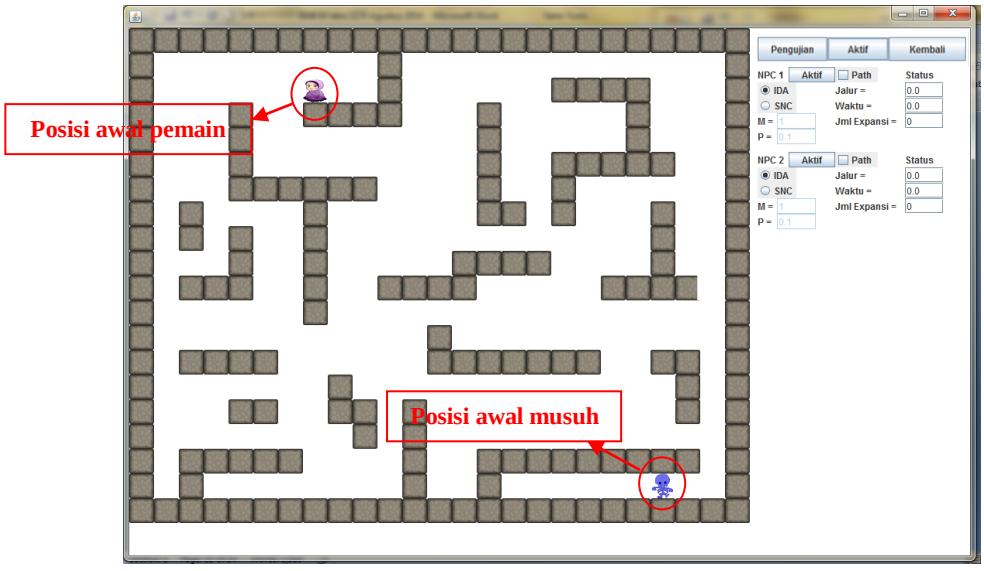
Pada Tabel 4.2 dan Gambar 4.9 terlihat dari pengujian untuk map *medium* terlihat semakin tinggi nilai *cache* (*M*), semakin menurun rata-rata jumlah simpul yang diekspansi. Hal ini dikarenakan semakin besar ukuran *cache* maka akan lebih banyak simpul yang sering dikunjungi yang bisa disimpan. Ini mengakibatkan jumlah total ekspansi simpul selama proses pencarian menjadi berkurang.

Dari hasil percobaan terlihat pada nilai *cache* dan probabilitas tertentu (*M* = 30, *p* = 0.8) menghasilkan rata-rata jumlah ekspansi simpul yang diperiksa semakin stabil dikarenakan besar *cache* mulai 30 dan probabilitas 0.8 sudah cukup

untuk menyimpan sebagian simpul yang sering dikunjungi. Penambahan ukuran *cache* tidak akan menurunkan jumlah ekspansi simpul secara signifikan lagi. Jika *p* diset lebih rendah dari 0.8 maka ekspansi simpul juga cenderung meningkat karena kemungkinan simpul yang sering dikunjungi untuk disimpan dalam *cache* menjadi lebih kecil.

3. Map *Hard*

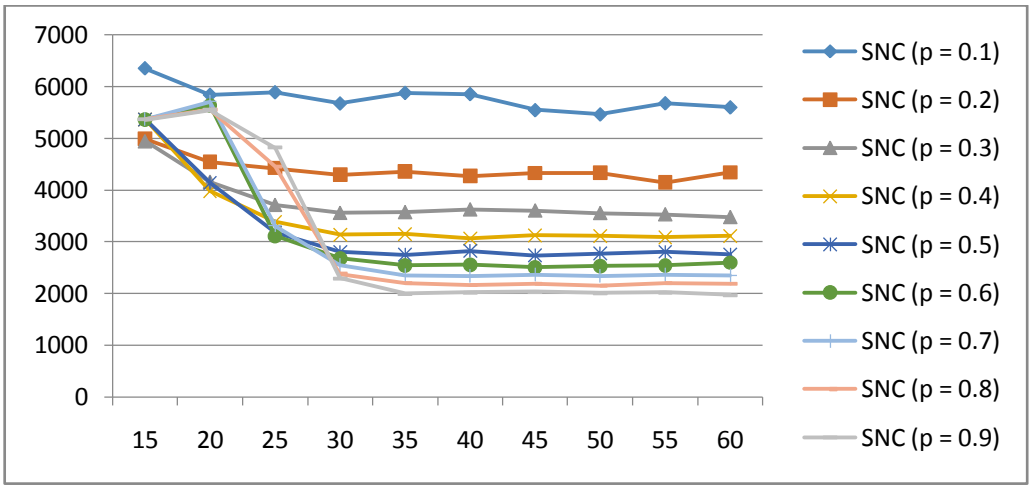
Pengujian pada map *hard* dilakukan dengan percobaan untuk 1 NPC dengan algoritma IDA* SNC dan mengubah nilai *M* dari 15 s/d 60. Sedangkan nilai *p* sendiri diset 0.1 ; 0.2 ; 0.3 ; 0.4 ; 0.5 ; 0.6 ; 0.7 ; dan 0.9. Adapun posisi awal NPC dan pemain pada map *hard* dapat dilihat pada Gambar 4.10. Percobaan yang dilakukan mengasumsikan posisi awal NPC dan pemain dalam keadaan diam. Detail pengujian terdapat pada Lampiran D. Hasil pengujian kemudian dirata-rata dan dibuat dalam Tabel 4.3 dan Gambar 4.11.



Gambar 4.10 Posisi NPC dan Pemain pada Map *Hard*

Tabel 4.3 Hasil Pengujian Parameter IDA* SNC Terhadap Ekspansi Simpul pada Map *Hard*

NPC Map *Hard*									
p / M	(0.1)	(0.2)	(0.3)	(0.4)	(0.5)	(0.6)	(0.7)	(0.8)	(0.9)
15	6353	4987	4947	5367	5367	5367	5367	5367	5367
20	5841	4545	4164	3982	4138	5642	5706	5553	5548
25	5890	4424	3716	3381	3177	3114	3307	4451	4819
30	5674	4300	3561	3138	2805	2682	2545	2376	2288
35	5878	4357	3575	3144	2746	2540	2344	2194	1990
40	5855	4274	3628	3060	2825	2558	2340	2164	2023
45	5545	4326	3607	3122	2736	2504	2358	2181	2032
50	5463	4333	3558	3111	2772	2526	2334	2148	2009
55	5678	4146	3533	3084	2805	2542	2362	2201	2022
60	5599	4338	3480	3111	2755	2597	2345	2186	1970



Gambar 4.11 Grafik Pengujian Parameter IDA* SNC Terhadap Ekspansi Simpul pada Map *Hard*

Pada Tabel 4.3 dan Gambar 4.11 terlihat dari pengujian untuk map *hard* terlihat semakin tinggi nilai *cache* (*M*), semakin menurun rata-rata jumlah simpul yang diekspansi. Hal ini dikarenakan semakin besar ukuran *cache* maka akan lebih banyak simpul yang sering dikunjungi yang bisa disimpan. Ini mengakibatkan jumlah total ekspansi simpul selama proses pencarian menjadi berkurang.

Dari hasil percobaan terlihat pada nilai *cache* dan probabilitas tertentu (*M* = 35, *p* = 0.7) menghasilkan rata-rata jumlah ekspansi simpul yang diperiksa

semakin stabil dikarenakan besar *cache* mulai 35 dan probabilitas 0.7 sudah cukup untuk menyimpan sebagian simpul yang sering dikunjungi. Penambahan ukuran *cache* tidak akan menurunkan jumlah ekspansi simpul secara signifikan lagi. Jika *p* diset lebih rendah dari 0.7 maka ekspansi simpul juga cenderung meningkat karena kemungkinan simpul yang sering dikunjungi untuk disimpan dalam *cache* menjadi lebih kecil.

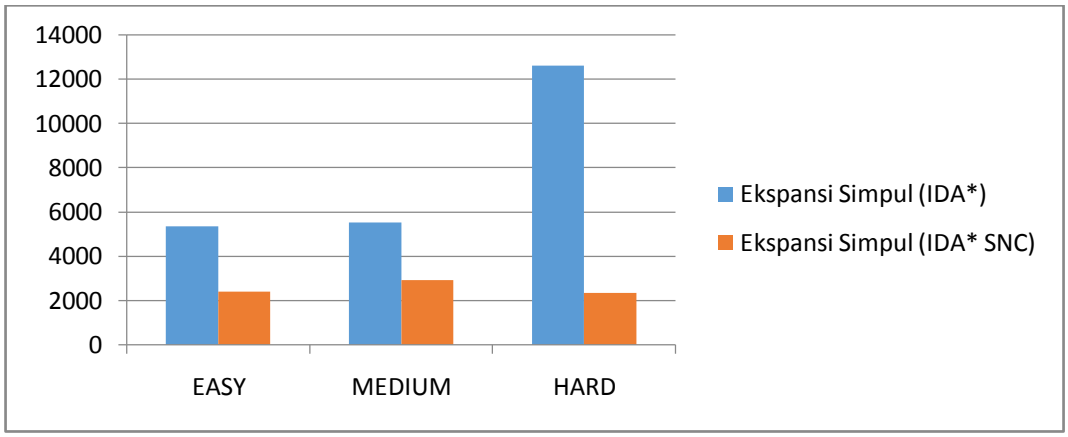
4.2.2 Pengujian Paramter Aplikasi

Pengujian parameter aplikasi dilakukan untuk mengetahui hasil dari pengujian algoritma IDA* dan IDA* SNC yang telah diiplementasi pada *game* labirin. Pengujian dilakukan untuk 2 NPC yang ada di map *easy, medium,* dan *hard* dengan NPC pertama menggunakan algoritma IDA* dan NPC kedua IDA* SNC. Sedangkan posisi awal kedua NPC diletakkan pada posisi yang sama. Untuk jumlah ekspansi simpul pada IDA* SNC, digunakan nilai parameter *M* dan *p* terbaik berdasarkan hasil pengujian dari poin 4.2.1.

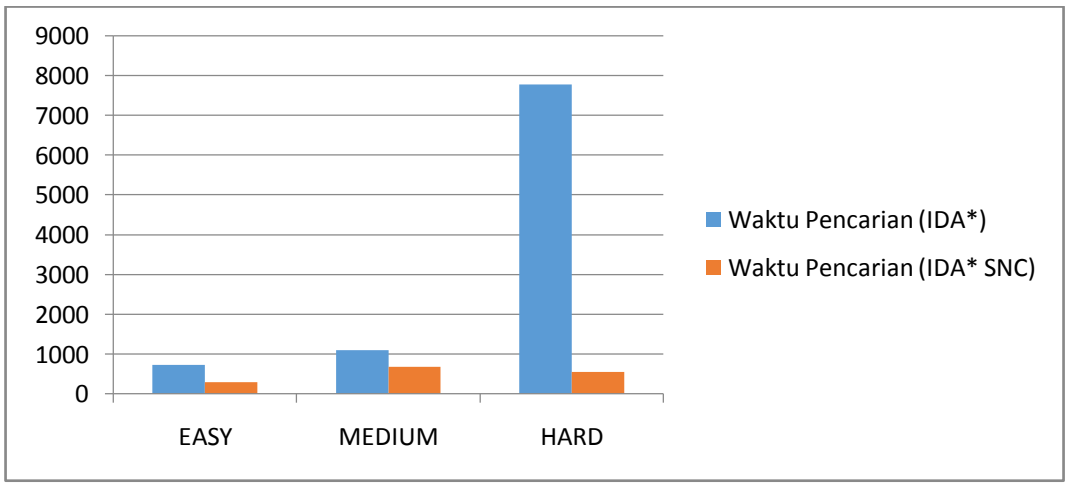
Performansi yang akan diuji pada aplikasi yang telah dibangun yaitu waktu pencarian, jumlah ekspansi simpul yang diperiksa dan langkah yang dihasilkan dengan parameter IDA* SNC. Adapun prioritas yang akan diambil dari performansi yang diujikan tersebut pada *game* labirin untuk pencarian jalan IDA* dan IDA* SNC adalah pertama dilihat dari jumlah ekspansi simpul yang diperiksa paling sedikit, karena semakin kecil jumlah simpul yang diperiksa maka semakin cepat langkah pencarian musuh. Sehingga waktu pencarian yang semakin sedikit menyebabkan NPC lebih cepat mengejar dan menangkap pemain. Kemudian yang terakhir dilihat dari panjang jalur yang merupakan jumlah jalur/*path* yang dilewati musuh dari titik awal sampai tiitk akhir. Berdasarkan hasil pengujian parameter aplikasi, diperoleh hasil perbandingan rata-rata ekspansi simpul yang diperiksa, waktu dan panjang jalur. Hasil pengujian performansi dapat dilihat pada Tabel 4.4, Gambar 4.12 dan Gambar 4.13.

Tabel 4.4 Hasil Pengujian Performansi IDA* dan IDA* SNC

Map	IDA*			IDA* SNC			
	Jumlah Ekspansi Simpul	Pjg Jalur	Waktu (ms)	Parameter Inputan	Jumlah Ekspansi Simpul	Pjg Jalur	Waktu (ms)
Easy	5351	24.23	727.21	M = 25 p = 0.7	2394	24.23	290.28
Medium	5535	33.64	1087.45	M = 30 p = 0.8	2913	33.64	682.49
Hard	12614	27.74	7775.68	M = 35 p = 0.7	2344	27.74	544.58

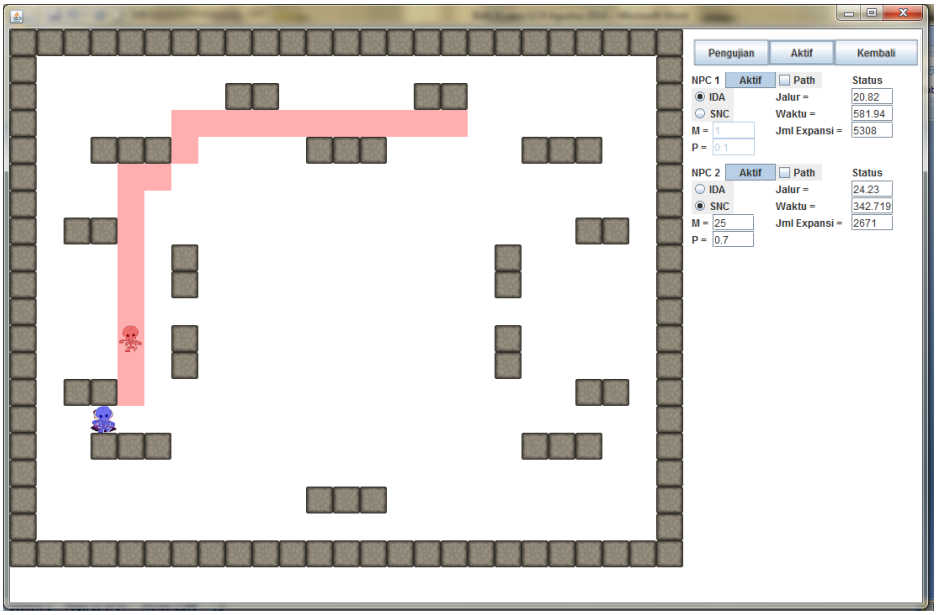


Gambar 4.12 Grafik Hasil Pengujian Ekspansi Simpul IDA* dan IDA* SNC

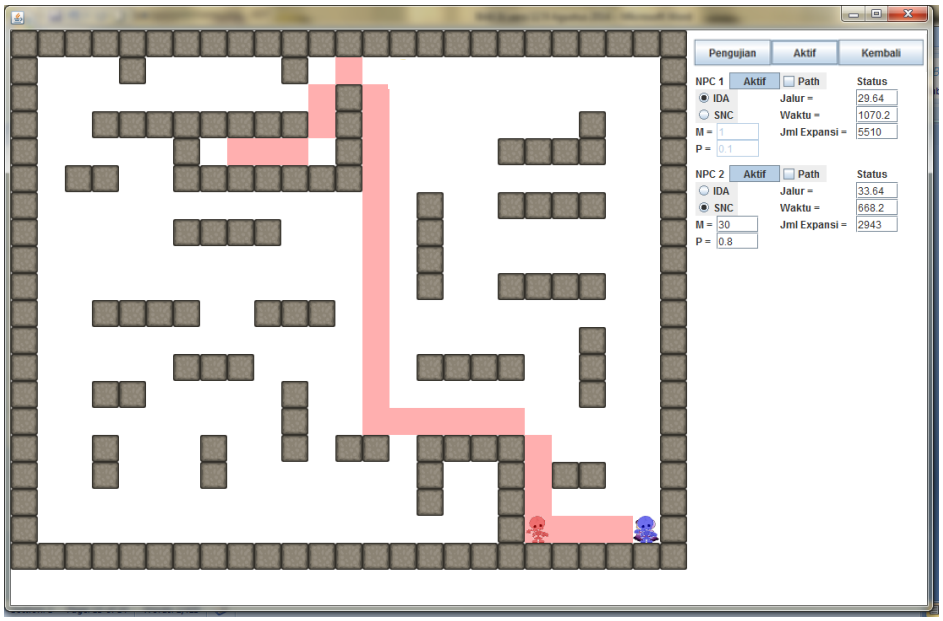


Gambar 4.13 Grafik Hasil Pengujian Waktu Pencarian IDA* dan IDA* SNC

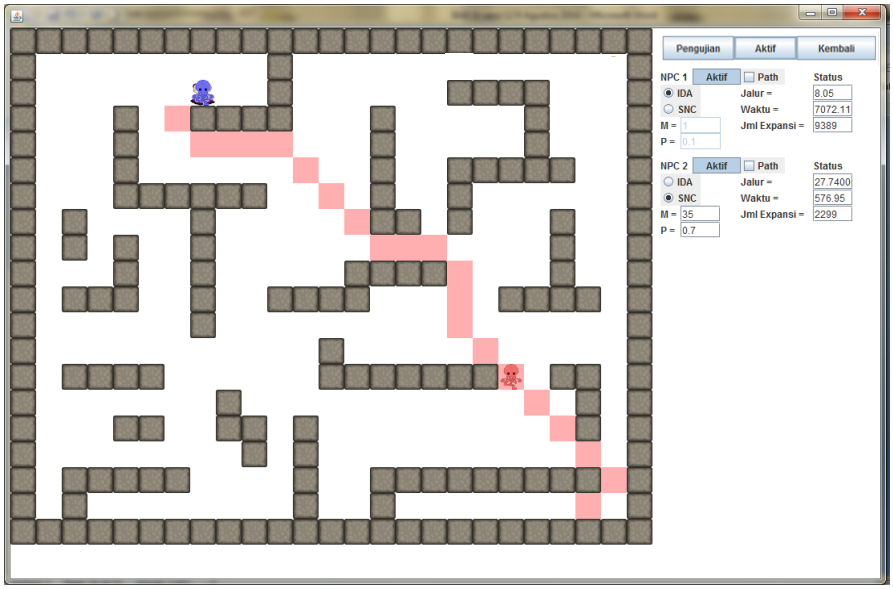
Untuk langkah yang dihasilkan oleh NPC dengan IDA* atau IDA* SNC menghasilkan langkah dan panjang jalur yang sama. Jalur/*path* dari setiap simpul yang dipilih setiap map dari titik awal musuh sampai mengejar dan menemukan pemain dapat dilihat pada Gambar 4.13, Gambar 4.14 dan Gambar 4.15.



Gambar 4.14 Jalur yang dilewati NPC pada Map *Easy*



Gambar 4.15 Jalur yang dilewati NPC pada Map *Medium*



Gambar 4.16 Jalur yang dilewati NPC pada Map *Hard*

Berdasarkan hasil analisis pada Tabel 4.4, Gambar 4.12 dan Gambar 4.13 hasil pengujian performansi terlihat bahwa NPC yang menggunakan IDA* SNC menghasilkan perbedaan jumlah ekspansi simpul yang signifikan daripada performansi NPC yang menggunakan IDA*. Karena IDA* SNC mengekspansi simpul yang lebih sedikit, maka waktu pencarian semakin cepat pula. Untuk total pajang jalur yang dipilih pada setiap NPC menghasilkan panjang jalur yang sama antara IDA* dengan IDA* SNC. Pada Gambar 4.14, Gambar 4.15 dan Gambar 4.16 juga terlihat dalam jalur yang dilewati, NPC yang menggunakan IDA* SNC lebih dulu mengejar dan menangkap pemain. Hal ini juga membuktikan bahwa waktu pencarian yang dibutuhkan NPC dengan IDA* SNC lebih sedikit daripada NPC dengan IDA.

Pemilihan untuk inputan nilai parameter juga dapat meningkatkan performa eksekusi algoritma IDA* SNC yaitu dengan men-*set* nilai inputan parameter *M* dan *p* yang tepat pada setiap NPC-nya. Misalnya pada map *hard*, nilai inputan parameter terbaik diperoleh ketika $M = 35$ dan $p = 0.7$, ini sudah cukup membuat IDA* SNC menjadi lebih optimal. Sehingga dengan parameter yang sudah di-*set*, IDA* SNC sudah menghasilkan jumlah ekspansi simpul yang jauh berbeda dengan jumlah ekspansi simpul IDA* yaitu sebanyak 2344 simpul

untuk IDA* SNC dan 12614 simpul untuk IDA*. Perbedaan ekspansi simpulnya terlihat sangat jauh bahkan untuk map *hard*, persentase penurunan ekspansi simpul antara IDA* dan IDA* SNC yaitu mencapai 81,42%. Sedangkan untuk waktu pencarian IDA* dan IDA* SNC pada map *hard* menghasilkan persentase penurunan yaitu mencapai 93%. Dari kedua hasil persentase tersebut, terlihat penggunaan IDA* SNC sangat berpengaruh ketika diimplementasikan pada *game* labirin.