BAB V

PENGUJIAN DAN PEMBAHASAN

5.1 Pengujian

5.1.1 *Clock Cycle*

Penghitungan *clock cycle* pada penelitian ini memanfatkan instruksi *Read Time Stamp Counter* (RDTSC). Instruksi RDTSC mengembalikan 64-bit *time stamp counter* (TSC), yang meningkat pada setiap siklus *clock*. Penghitung ini merupakan pilihan yang tersedia pada arsitektur x86 (Yoshii, 2006). Peneliti bernama Kazutomo Yoshii yang berasal dari Argonne National Laboratory telah menyediakan sebuah *library* istruksi yang dapat digunakan untuk membaca TSC tersebut. Library tersebut digunakan penulis untuk membantu perolehan data performa dalam melakukan proses operasi dalam algoritma tertentu.

5.1.1 Variabel Pengujian

Variabel *input* penelitian dibatasi pada dua parameter karakteristik jaringan *mesh*, yaitu jumlah lapisan (*layer*) dan jumlah simpul per-lapisan (*sublayer*). Parameter karakteristik jaringan ini akan menentukan bentuk jaringan yang akan dibangkitkan menggunakan simulator. Jaringan *mesh* berlapis akan terbentuk dengan hubungan berupa kanal komunikasi antar lapisan simpul. Setiap kanal yang terbentuk memiliki bobot yang ditentukan oleh pembangkit angka acak (*random number generator*) dengan batasan atara 1 hingga 100. Bobot kanal tersebut menjadi penentu dalam proses pencarian jalur tependek.

Di sisi *output*, hasil pencarian yang diuji akan memperoleh data berupa diantaranya banyaknya penjumlahan (*addition*), pemasukan nilai (*assignment*), perbandingan (*comparison*), jumlah bobot kanal yang dilalui (*weight*), dan waktu simulasi (*time*). Algoritma Dijkstra, A*, dan Floyd-Warshall memiliki variabel yang terpisah sehingga penghitungan beban komputasi untuk setiap algoritma tersebut dapat dilakukan secara independen.

5.2 Hasil Pengujian

Pada pengujian dengan variabel waktu simulasi, ketelitian yang dapat diperoleh adalah 1 milisecond. Ketelitian ini dipengaruhi oleh fungsi *clock()* yang digunakan penulis dalam membaca waktu sistem dalam komputer menggunakan simulator OMNeT++. Hal ini menyebabkan pada pengujian waktu simulasi terdapat beberapa data yang bernilai 0. Kondisi tersebut menunjukkan bahwa hasil waktu simulasi yang diperoleh memiliki nilai lebih kecil dari 1000 milisecond.

Setiap jaringan *mesh* yang dibangkitkan akan dilakukan pengujian terhadap algoritma Dijkstra, A*, dan Floyd-Warshall secara berurutan berdasarkan penjadwalan pada simulator OMNeT++.

Hasil penjumlahan banyaknya operasi yang dilakukan dalam proses pencarian jalur dilakukan dalam program sehingga menghasilkan keluaran berupa angka non-negatif. Variabel *clock cycle* diperoleh dari selisih pencatatan *clock tick* prosesor pada awal dan akhir simulasi.

Berikut adalah rata-rata pengujian algoritma Dijkstra, A*, dan Floyd-Warshall yang dihasilkan menggunakan simulator OMNeT++.

Tabel 5.1 Hasil rata-rata pengujian algoritma Dijkstra

Node	Addition	Assignment	Comparison	Clock	Time (s)
10	28	111	40	55254	0
20	94	258	116	117151	0
40	242	567	284	276986	0
80	576	1273	663	763126	0
150	1864	3165	2016	2291992	0.0015
250	3140	5332	3392	5219431	0.0016
500	6340	10745	6842	17719206	0.0099
1000	12920	21629	13922	63747565	0.0316
2500	32522	50035	35024	377165510	0.189
5000	65136	100149	70138	1584615310	0.78
10000	130320	200333	140322	3337393788	5.975
20000	259960	399973	279962	2847891480	22.948

Tabel 5.2 Hasil rata-rata pengujian algoritma A*

Node	Addition	Assignment	Comparison	Clock	Time (s)
10	8	57	8	42792	0
20	33	147	33	91004	0
40	85	333	85	260945	0
80	211	775	211	791467	0
150	672	1765	672	2182834	0
250	1124	2940	1124	4966576	0.0032
500	2400	6200	2400	18046241	0.0094
1000	4616	12189	4616	60442973	0.0292
2500	16239	58706	16239	363313650	0.187
5000	32546	117627	32546	1549557780	0.749
10000	65138	235403	65138	1498683678	5.054
20000	129958	469863	129958	351854100	21.715

Tabel 5.3 Hasil rata-rata pengujian algoritma Floyd-Warshall

Node	Addition	Assignment	Comparison	Clock	Time (s)
10	1756	28	1728	40968	0
20	10835	187	10648	239030	0
40	75076	988	74088	1661617	0.0017
80	722081	6150	715931	16016273	0.0099
150	3534311	22503	3511808	77552374	0.0418
250	16072770	69762	16003008	352096610	0.1794
500	126784083	278075	126506008	2811115784	1.4079
1000	1007219527	1207518	1006012008	871672345	11.2011
2500	2780720607	3092487	2777628120	1038874340	172.755
5000	6084433025	12434901	5968008424	2237366358	1372.49

Beban komputasi diperoleh dengan memberikan nilai pendekatan terhadap setiap operasi yang dilakukan algoritma. Nilai tersebut diperoleh dari besaran standar beban instrusi pada keluarga prosesor x86 (Fog, 2012). Operasi penjumlahan, perbandingan, dan pemasukan memiliki nilai pendekatan yang sama, yaitu 1 *clock cycle* (Fog, 2012). Dengan demikian dari hasil rata-rata ketiga algoritma tersebut dapat diperoleh beban komputasi sebagai berikut.

Tabel 5.4 Beban komputasi algoritma *shortest path*

Node	Dijkstra	A*	Floyd-Warshall
10	179	73	3512
20	468	213	21670
40	1093	503	150152
80	2512	1197	1444162
150	7045	3109	7068622
250	11864	5188	32145540
500	23927	11000	253568166
1000	48471	21421	2014439053
2500	117581	91184	5561441214
5000	235423	182719	12064876350
10000	470975	365679	-
20000	939895	729779	-

(-) pengujian tidak dapat dilakukan

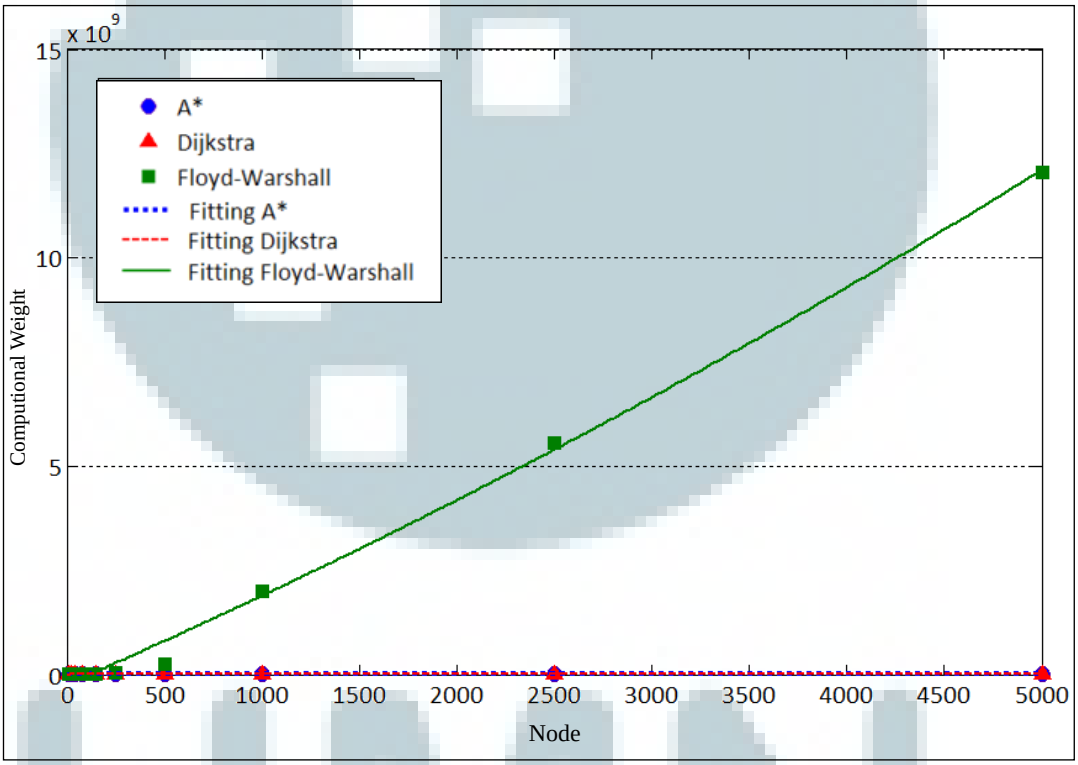
Dari segi penggunaan memori, nilai diperoleh dengan menjumlahkan penggunaan data dalam memproses algoritma. Setiap algoritma membutuhkan variabel penampung dengan tipe yang beragam, diantaranya *integer*, *double*, dan *boolean*. Tipe data tertentu memiliki besaran memori (*byte*) yang berbeda. Penggunaan memori pada ketiga algoritma dapat dilihat pada tabel berikut.

Tabel 5.5 Penggunaan memori algoritma *shortest path*

Node	Dijkstra	A*	Floyd-Warshall
10	171	180	800
20	341	360	3200
40	681	720	12800
80	1361	1440	51200
150	2551	2700	180000
250	4251	4500	500000
500	8501	9000	2000000
1000	17001	18000	8000000
2500	42501	45000	50000000
5000	85001	90000	200000000

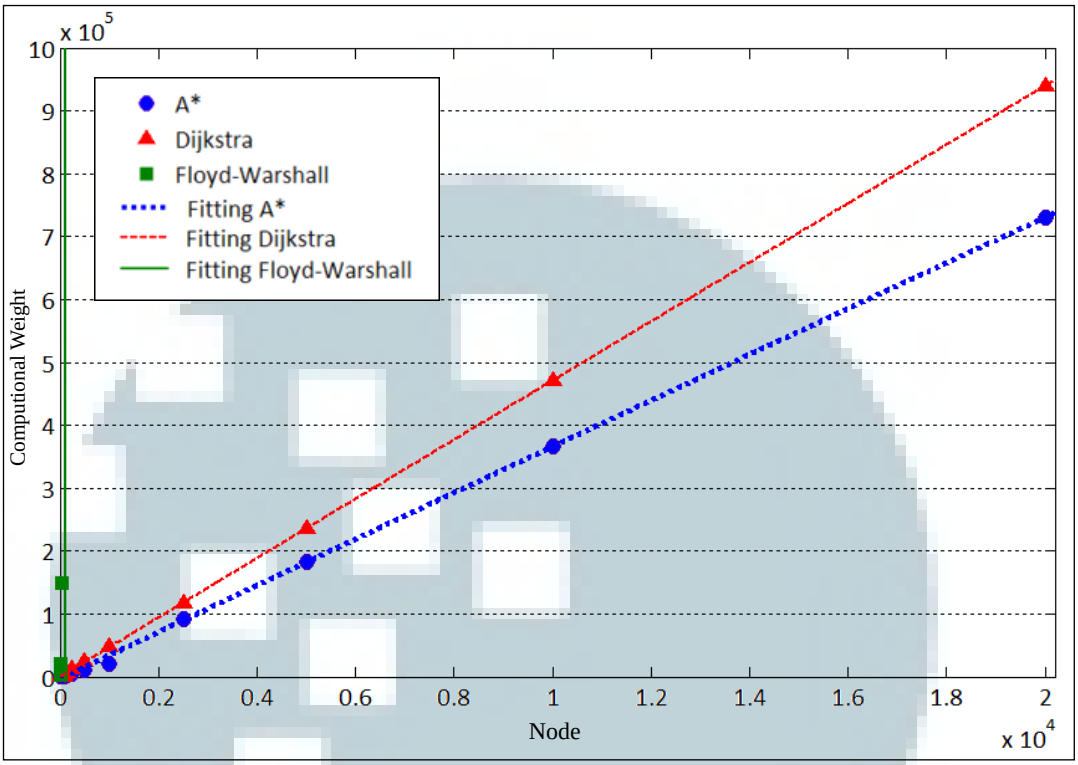
5.3 Pembahasan

Hasil pengujian yang dilakukan terhadap algoritma Dijkstra, A*, dan Floyd-Warshall dapat dilakukan pendekatan menggunakan fungsi kuadrat, Pendekatan tersebut dapat diperoleh grafik perbandingan antara ketiga algoritma yang diuji dengan parameter operasi penjumlahan, perbandingan, dan pemasukan nilai sebagai berikut.



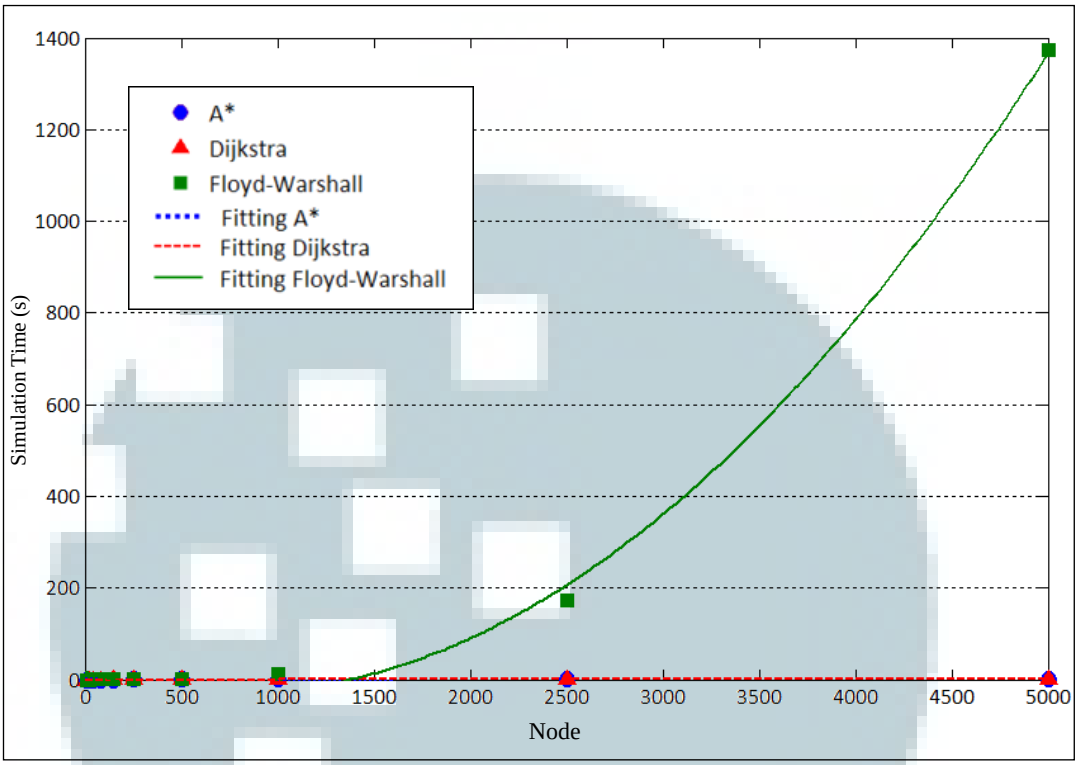
Gambar 5.1 Perbandingan beban komputasi

Berdasarkan Gambar 5.1 diketahui bahwa algoritma Floyd-Warshall memiliki nilai beban komputasi yang jauh berbeda dengan algoritma Dijkstra dan A*. Untuk dapat mengetahui tingkat perbedaan antara algoritma Dijkstra dan A* maka grafik perbandingan dibatasi pada tingkat komplesitas maksimum 1 juta.

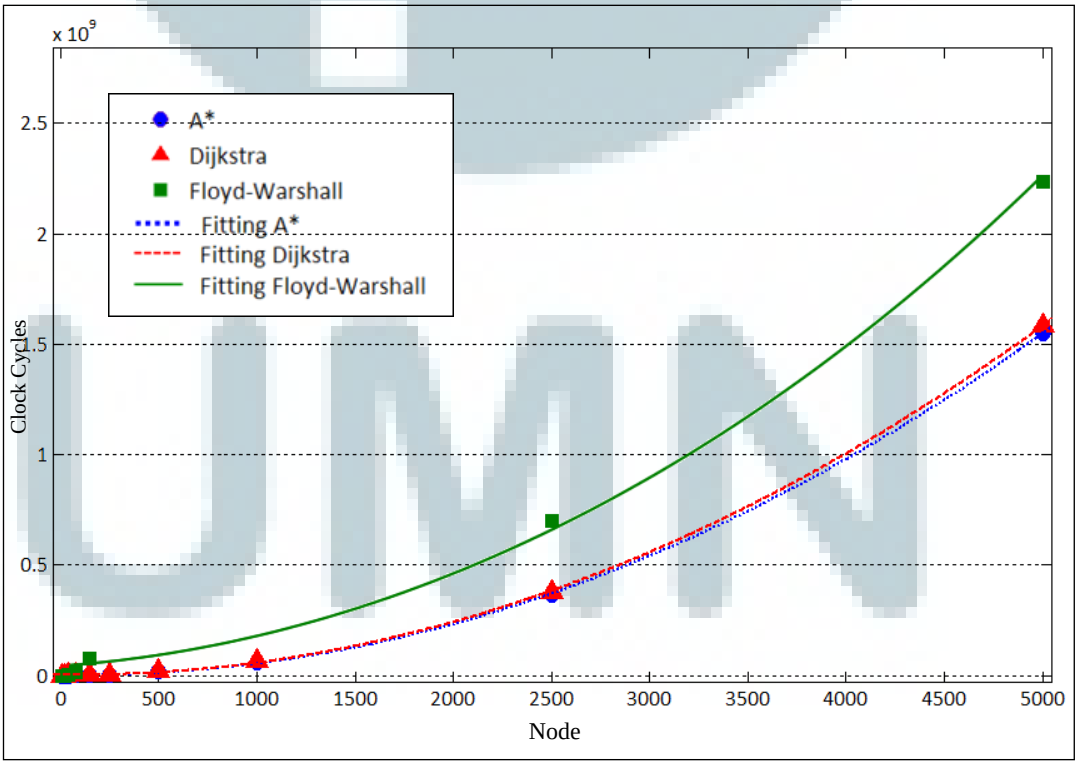


Gambar 5.2 Perbandingan beban komputasi dengan batasan

Selain membandingkan dari segi komputasi, dapat diperoleh grafik perbandingan algoritma yang diuji dengan parameter *clock cycle* dan waktu simulasi sebagai berikut.



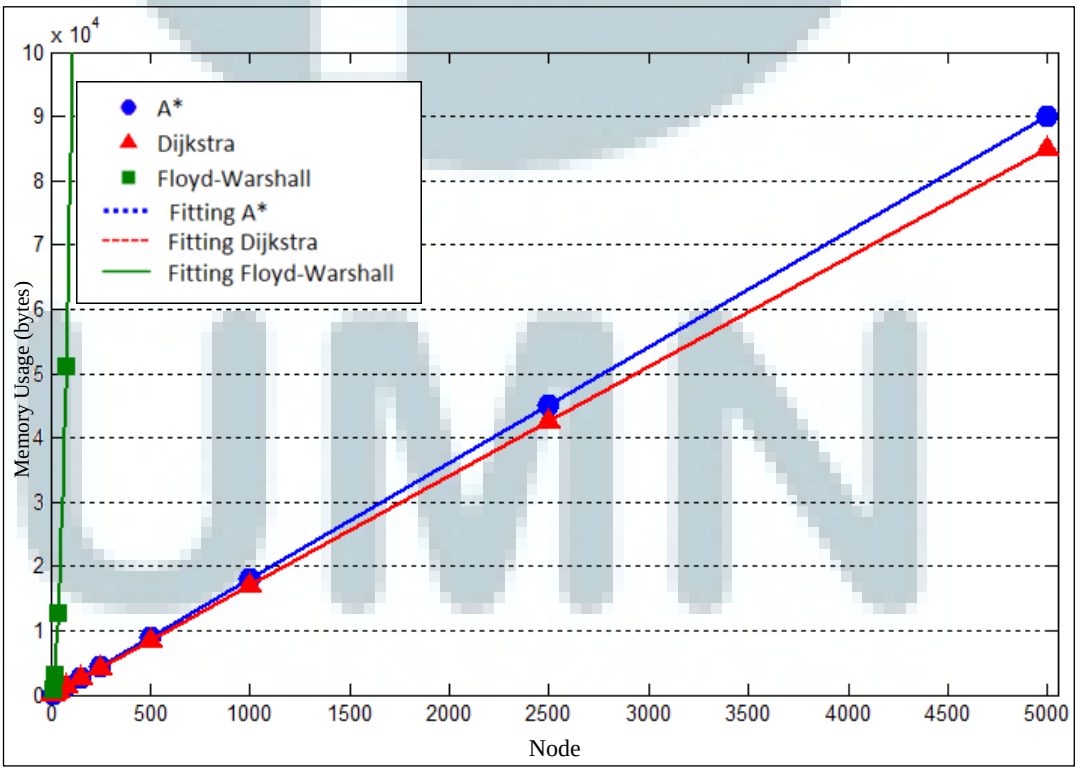
Gambar 5.3 Perbandingan waktu simulasi



Gambar 5.4 Perbandingan *clock cycle*

Pengujian terhadap algoritma Floyd-Warshall hanya dilakukan hingga batasan simpul sebanyak 5000. Hal tersebut disebabkan oleh keterbatasan simulator untuk memproses algoritma Floyd-Warshall. Hasil penelitian menunjukkan bahwa algoritma Floyd-Warshall memiliki jumlah operasi yang paling banyak dibandingkan algoritma Dijkstra dan A*. Selain itu, waktu yang diperlukan untuk mensimulasikan algoritma Floyd-Warshall jauh lebih besar.

Nilai beban komputasi yang diperoleh algoritma A* lebih unggul dibandingkan Dijkstra karena terdapat fungsi heuristik, di mana dapat diperkirakan jarak dari suatu simpul ke tujuan. Hal tersebut menyebabkan algoritma ini tidak perlu melakukan pengecekan terhadap semua simpul yang terdapat pada jaringan, sehingga operasi yang dilakukan akan semakin kecil.



Gambar 5.5 Perbandingan penggunaan memori

Dari segi penggunaan memori dapat diperoleh grafik perbandingan seperti yang dapat dilihat pada Gambar 5.5. Penggunaan memori pada algoritma Dijkstra lebih unggul dari A* karena Dijkstra lebih sedikit membutuhkan variabel penampung. Penelitian ini lebih difokuskan kepada beban komputasi, sehingga algoritma A* yang digunakan untuk implementasi pada perangkat seluler berbasis Android.

Penggunaan memori untuk masing-masing algoritma dapat dirumuskan sebagai berikut:

Dijkstra : $f(x) = 2an + a + n$ (i)

A* : $f(x) = an + n$ (ii)

Floyd-Warshall : $f(x) = an^2$ (iii)

a : besar variabel penampung (byte)

n : jumlah simpul