

DAFTAR ISI

Halaman

HALAMAN JUDUL	
PERNYATAAN KEASLIAN TUGAS AKHIR	
PERSETUJUAN DOSEN PEMBIMBING	
PERSETUJUAN TIM PENGUJI TUGAS AKHIR	
ABSTRACT	v
ABSTRAK	vi
KATA PENGANTAR.....	vii
DAFTAR ISI.....	ix
DAFTAR GAMBAR.....	xii
DAFTAR TABEL	xv
DAFTAR DEFINISI	xvi
BAB I PENDAHULUAN.....	1
1.1 Latar belakang	1
1.2 Perumusan masalah	2
1.3 Batasan masalah	3
1.4 Tujuan.....	3
1.5 Metodologi	4
1.6 Sistematika penulisan	5
BAB II LANDASAN TEORI	7
2.1 <i>Set</i>	7
2.2 <i>Graph</i>	8
2.3 <i>Adjacency list</i> dan <i>adjacency matrix</i>	9
2.4 <i>Hypergraph</i>	10
2.5 <i>Depth First Search (DFS)</i>	11
2.6 Relasi objek komposisi.....	16
2.7 <i>Partially Ordered Set</i>	16
2.8 <i>Hasse Diagram</i>	17
2.9 <i>Transitive closure</i>	18

2.10	Algoritma DFS <i>transitive closure</i>	19
2.11	Algoritma <i>Warshall transitive closure</i>	19
2.12	<i>Connected component</i>	20
2.13	<i>Standard Template Library (STL)</i>	22
2.14	<i>White box testing</i>	22
BAB III PERANCANGAN SISTEM		24
3.1	Pendahuluan	24
3.2	<i>Flowchart</i> mekanisme kerja program	26
3.3	Identifikasi objek.....	27
3.3.1	<i>Input dan output</i>	27
3.3.2	Tahap identifikasi objek	28
3.4.	Pembentukan <i>hasse diagram</i> dengan algoritma <i>growing</i>	29
3.4.1	<i>Input dan output</i>	29
3.4.2	Tahap pembentukan <i>hasse diagram</i> dengan algoritma <i>growing</i>	30
3.5.	Pembentukan <i>transitive closure</i>	38
3.5.1	<i>Input dan output</i>	38
3.5.2	Tahap pembentukan <i>transitive closure</i> dengan <i>Warshall</i>	38
3.5.3	Tahap pembentukan <i>transitive closure</i> dengan DFS	40
3.6	Identifikasi komponen.....	41
3.6.1	<i>Input dan output</i>	42
3.6.2	Tahap eksekusi identifikasi komponen	42
BAB IV IMPLEMENTASI DAN PENGUJIAN		45
4.1	Implementasi program.....	45
4.2	Implementasi STL pada program	49
4.2.1	<i>Container</i>	49
4.2.2	<i>Algorithm</i>	51
4.3	Pengujian program	53
4.3.1	Objek Geometri	56
4.3.2	Objek Durga	59
4.3.3	<i>Dummy data</i>	65
BAB V KESIMPULAN DAN SARAN.....		69

5.1. Kesimpulan.....	69
5.2. Saran.....	70
DAFTAR PUSTAKA.....	71

DAFTAR GAMBAR

Gambar 2.1 <i>Bipartite graph</i> (a), $K_{4,3}$ (b), $K_{3,3}$ (c), $K_{1,4}$ (d).....	9
Gambar 2.2 Contoh <i>hypergraph</i> (a) dan <i>incidence matrix</i> (b).....	11
Gambar 2.3 <i>Pseudocode</i> dengan prosedur DFS dan DFS-Visit	12
Gambar 2.4 DFS pada <i>directed graph</i> dengan <i>timestamped</i> (a), <i>interval</i> antara <i>discovery time</i> dan <i>finishing time</i> setiap <i>vertex</i> (b)	14
Gambar 2.5 <i>Directed graph</i> dengan <i>tree</i> , <i>back</i> , <i>forward</i> , dan <i>cross edges</i>	15
Gambar 2.6 Contoh <i>hasse diagram</i> dari <i>poset</i> $(P(S), \subseteq)$	17
Gambar 2.7 Cuplikan <i>source code</i> dari algoritma <i>Warshall</i>	20
Gambar 2.8 Cuplikan <i>source code</i> untuk identifikasi <i>connected component</i>	21
Gambar 3.1 <i>Flowchart</i> mekanisme kerja program.	26
Gambar 3.2 <i>File incident</i> objek Geometri.....	27
Gambar 3.3 <i>Flowchart</i> fungsi <i>executeGrowing</i> dalam program.....	34
Gambar 3.4 <i>Flowchart</i> fungsi <i>dfsGrowing</i> dalam program.....	35
Gambar 3.5 Ilustrasi konstruksi <i>graph</i> P_m dengan algoritma <i>growing</i> tahap $i - iv$ berdasarkan <i>feature value</i> objek Geometri	36
Gambar 3.6 Ilustrasi konstruksi <i>graph</i> P_m dengan algoritma <i>growing</i> tahap $v - vi$ berdasarkan <i>feature value</i> objek Geometri	36
Gambar 4.1 Tampilan utama program identifikasi relasi objek komposisi	45
Gambar 4.2 Tampilan program untuk memilih metode <i>transitive</i>	45
Gambar 4.3 Tampilan program untuk memilih <i>input object</i>	46
Gambar 4.4 Hasil identifikasi seluruh objek Geometri.....	46
Gambar 4.5 Hasil identifikasi relasi <i>feature value</i> objek Geometri.....	48
Gambar 4.6 Hasil identifikasi relasi <i>feature structure</i> objek Geometri	49
Gambar 4.7 Cuplikan kode <i>struct obj</i> dalam program.....	50
Gambar 4.8 Algoritma <i>includes</i> STL berdasarkan <i>feature value</i>	51
Gambar 4.9 Fungsi <i>sort</i> dengan <i>binary predicate sortByFv</i>	52

Gambar 4.10 <i>Sorting criterion</i> berdasarkan $ Fv $ dalam <i>vector of struct</i> objek K .	52
Gambar 4.11 Cuplikan kode program identifikasi relasi objek komposisi	54
Gambar 4.12 Cuplikan kode program membentuk objek dari kelas <i>identification</i>	55
Gambar 4.13 Cuplikan kode program untuk atribut dari dari kelas <i>Identification</i>	55
Gambar 4.14 Cuplikan kode program untuk periksa relasi objek komposisi	56
Gambar 4.15 Hasil algoritma <i>growing</i> berdasarkan <i>feature value</i> objek seluruh objek Geometri.....	57
Gambar 4.16 Hasil algoritma <i>transitive closure</i> dengan DFS berdasarkan <i>feature value</i> seluruh objek Geometri	57
Gambar 4.17 Hasil algoritma <i>transitive closure</i> dengan <i>Warshall</i> berdasarkan <i>feature value</i> seluruh objek Geometri	57
Gambar 4.18 Hasil <i>transitive closure</i> berdasarkan <i>feature value</i> seluruh objek Geometri.....	57
Gambar 4.19 Hasil identifikasi komponen berdasarkan <i>feature value</i> seluruh objek Geometri.....	57
Gambar 4.20 Visualisasi relasi seluruh objek Geometri berdasarkan <i>feature value</i>	58
Gambar 4.21 Hasil algoritma <i>growing</i> berdasarkan <i>feature structure</i> seluruh objek Geometri.....	58
Gambar 4.22 Hasil algoritma <i>transitive closure</i> dengan DFS berdasarkan <i>feature structure</i> seluruh objek Geometri	59
Gambar 4.23 Hasil algoritma <i>transitive closure</i> dengan <i>Warshall</i> berdasarkan <i>feature structure</i> seluruh objek Geometri	59
Gambar 4.24 Hasil <i>transitive closure</i> berdasarkan <i>feature structure</i> seluruh objek Geometri.....	59
Gambar 4.25 Hasil identifikasi komponen berdasarkan <i>feature structure</i> seluruh objek Geometri.....	59
Gambar 4.26 Visualisasi relasi seluruh objek Geometri <i>feature structure</i>	59
Gambar 4.27 Hasil algoritma <i>growing</i> berdasarkan <i>feature value</i> objek Durga...	60

Gambar 4.28 Hasil algoritma <i>transitive closure</i> dengan DFS berdasarkan <i>feature value</i> objek Durga	60
Gambar 4.29 Hasil algoritma <i>transitive closure</i> dengan <i>Warshall</i> berdasarkan <i>feature value</i> objek Durga.....	61
Gambar 4.30 Hasil <i>transitive closure</i> berdasarkan <i>feature value</i> objek Durga	61
Gambar 4.31 Hasil identifikasi komponen berdasarkan <i>feature value</i> seluruh objek Durga.....	61
Gambar 4.32 Visualisasi relasi objek Durga <i>feature value</i>	62
Gambar 4.33 Hasil algoritma <i>growing</i> berdasarkan <i>feature structure</i> objek Durga	63
Gambar 4.34 Hasil algoritma <i>transitive closure</i> dengan DFS berdasarkan <i>feature structure</i> objek Durga	63
Gambar 4.35 Hasil algoritma <i>transitive closure</i> dengan <i>Warshall</i> berdasarkan <i>feature structure</i> objek Durga	63
Gambar 4.36 Hasil <i>transitive closure</i> berdasarkan <i>feature structure</i> seluruh objek Durga.....	64
Gambar 4.37 Hasil identifikasi komponen berdasarkan <i>feature structure</i> objek Durga.....	64
Gambar 4.38 Visualisasi relasi objek Durga berdasarkan <i>feature structure</i> objek Durga.....	65
Gambar 4.39 Grafik identifikasi relasi <i>feature value</i> dan <i>feature structure</i> dari Tabel 4.3.....	66
Gambar 4.40 Grafik algoritma <i>Warshall</i> dan DFS dari Tabel 4.4	67
Gambar 4.41 Grafik algoritma <i>Warshall</i> dan DFS dari Tabel 4.3	68

DAFTAR TABEL

Tabel 3.1 Hasil identifikasi objek Geometri dalam koleksi K.....	28
Tabel 4.1 <i>Container STL</i> yang digunakan dalam program	51
Tabel 4.2 Algoritma <i>STL</i> yang digunakan dalam program	52
Tabel 4.3 <i>Runtime</i> algoritma <i>growing</i> berdasarkan <i>feature value</i> dan <i>feature structure</i>	65
Tabel 4.4 time Algoritma <i>Warshall</i> dan <i>DFS</i> berdasarkan <i>feature value</i>	67
Tabel 4.5 <i>Runtime</i> Algoritma <i>Warshall</i> dan <i>DFS</i> berdasarkan <i>feature structure</i> .	68

DAFTAR DEFINISI

Edge adalah relasi yang menghubungkan dua *vertex* dalam *graph*.

Feature structure adalah struktur fitur yang dimiliki suatu objek

Feature value adalah nilai fitur yang dimiliki suatu objek berupa pasangan

Graph adalah sekumpulan *vertex* (objek) dan *edge* (relasi), dimana setiap *edge* menghubungkan dua *vertex*.

Hasse diagram (Minimum spanning subgraph) adalah *graph* yang memetakan relasi dalam poset dengan sifat transitive dan reflexive yang telah direduksi

Poset adalah relasi yang membentuk *transitive*, *reflexive*, atau *antisymmetric* pada suatu objek atau set.

Partially similiar adalah adanya kesamaan antara dua objek yang berbeda melalui himpunan ciri yang seragam.

Proper superset adalah relasi yang terbentuk jika A superset dari B, dimana seluruh elemen dari B ada pada A, dan A minimal memiliki satu elemen yang tidak ada pada B. Sehingga A *proper superset* B

Set adalah sekumpulan objek tanpa memperhatikan urutan.

Runtime adalah lamanya eksekusi algoritma dalam program

Subset adalah relasi yang terbentuk jika seluruh elemen dari B ada pada A, sehingga B subset A.

Superset adalah relasi yang terbentuk jika seluruh elemen dari B ada pada A, sehingga A superset B

STL adalah *Standard Template Library* dari *library* C++ yang berbentuk template

dan terdiri dari tiga komponen yaitu *container*, *algorithm*, dan *iterator*.

Transitive Closure adalah relasi dalam *graph* yang dibentuk dari objek A ke objek C, karena A memiliki relasi dengan objek B dan objek B memiliki relasi dengan objek C.

Vertex adalah objek dalam *graph*.