

18.482/ITS/H/2003



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER

**TUGAS AKHIR
(KP 1701)**

**ANALISIS PEMILIHAN LAYOUT RUANG AKOMODASI
MENGUNAKAN TEKNIK *TOPOLOGICAL AND
NUMERICAL PATTERN MATCHING* PADA KAPAL TANKER**

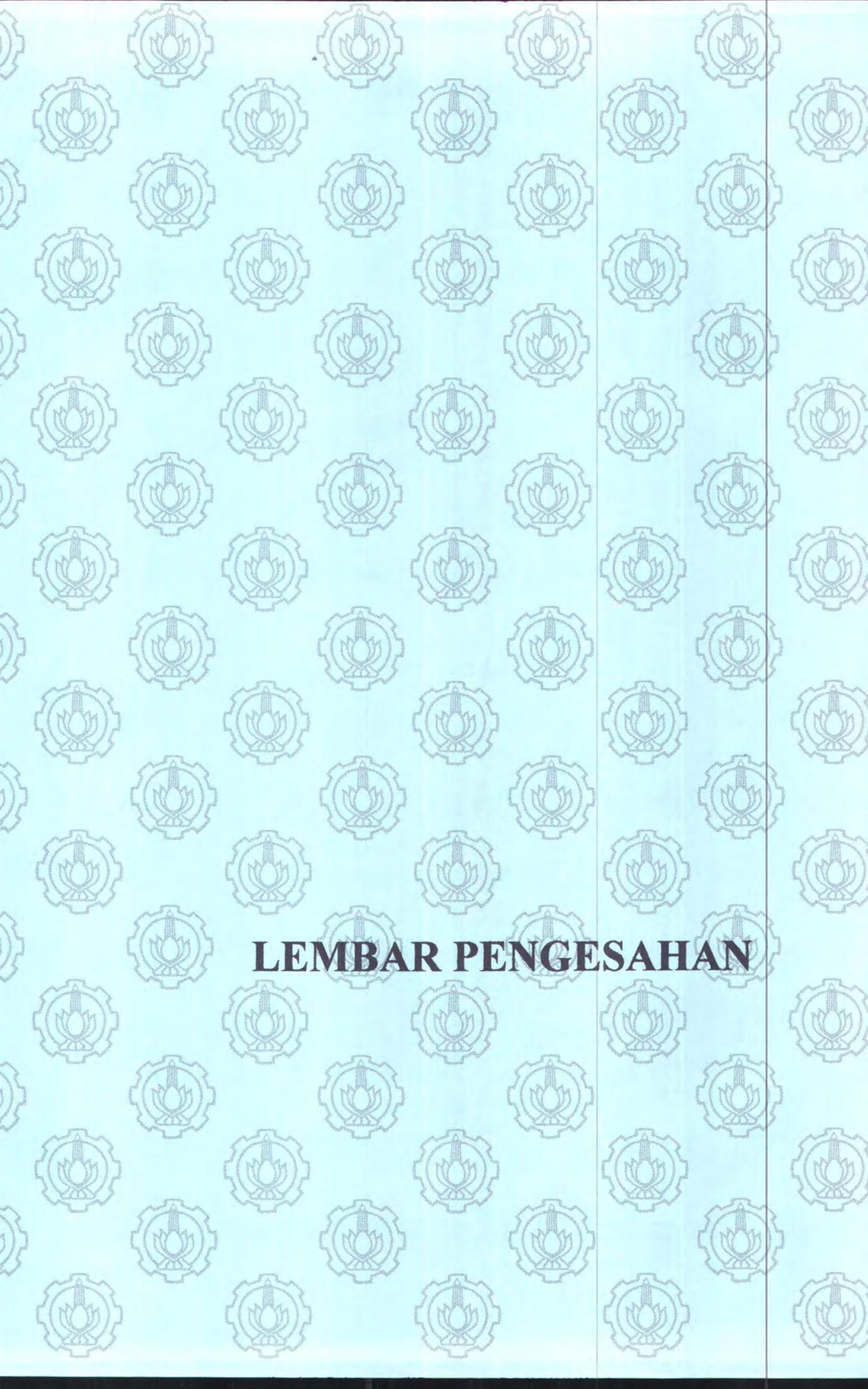


RSPe
623.85
Wiy
2-1
2003

Disusun oleh :
WAHYU WIYATI
NRP. 4197 100 066

PERPUSTAKAAN ITS	
Tgl. Terima	19-8-2003
Terima Dari	1/
No. Agenda Prep.	218784

**JURUSAN TEKNIK PERKAPALAN
FAKULTAS TEKNOLOGI KELAUTAN
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2003**



LEMBAR PENGESAHAN

LEMBAR PENGESAHAN

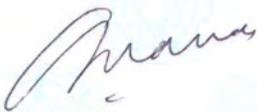
ANALISIS PEMILIHAN LAYOUT RUANG AKOMODASI MENGUNAKAN TEKNIK *TOPOLOGICAL AND NUMERICAL PATTERN MATCHING* PADA KAPAL TANKER

TUGAS AKHIR (KP 1701)

Diajukan Guna Memenuhi Sebagian
Persyaratan Untuk Memperoleh Gelar Sarjana Teknik Perkapalan
Pada
Jurusan Teknik Perkapalan
Fakultas Teknologi Kelautan
Institut Teknologi Sepuluh Nopember
Surabaya

Surabaya, 12-2-2003
Mengetahui dan Menyetujui,

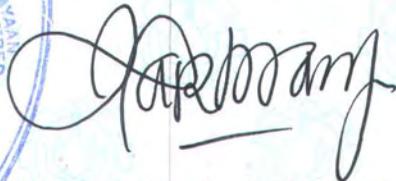
Dosen Pembimbing I



I. G. N. S. Buana, S.T., M.Eng
NIP. 132 085 800



Dosen Pembimbing II



Ir. Djauhar Manfaat, M.Sc., Ph.D.
NIP. 131 651 444

LEMBAR PENGESAHAN

ANALISIS PEMILIHAN LAYOUT RUANG AKOMODASI MENGUNAKAN TEKNIK *TOPOLOGICAL AND NUMERICAL PATTERN MATCHING* PADA KAPAL TANKER

TUGAS AKHIR

(KP 1701)

Telah Mengalami Revisi sesuai dengan Hasil Sidang Ujian Tugas Akhir

Pada

Jurusan Teknik Perkapalan

Fakultas Teknologi Kelautan

Institut Teknologi Sepuluh Nopember

Surabaya

Surabaya, 5 Pebruari 2003

Mengetahui dan Menyetujui,

Dosen Pembimbing I



I. G. N. S. Buana, S.T., M.Eng.

NIP. 132 085 800

Dosen Pembimbing II



Ir. Djauhar Manfaat, M.Sc., Ph.D.

NIP. 131 651 444





ABSTRAK

INSTITUT TEKNOLOGI SEPULUH NOPEMBER (ITS)

ABSTRAK

FAKULTAS TEKNOLOGI KELAUTAN

JURUSAN TEKNIK PERKAPALAN

Sarjana Teknik (S 1)

**ANALISIS PEMILIHAN LAYOUT RUANG AKOMODASI
MENGUNAKAN TEKNIK *TOPOLOGICAL AND NUMERICAL
PATTERN MATCHING* PADA KAPAL TANKER**

Oleh : Wahyu Wiyati

Pembimbing I : I. G. N. S. Buana, S.T., M.Eng.

Pembimbing II : Ir. Djauhar Manfaat, M.Sc., Ph.D.

Metode kapal pembanding yang lebih banyak memanfaatkan desain lama, saat ini sering digunakan untuk merencanakan suatu kapal. Terutama digunakan untuk mendesain Rencana Umum, yang intinya adalah merencanakan ruang-ruang akomodasi dan penempatannya pada geladak akomodasi. Hal ini sangat menguntungkan bagi desainer karena tidak harus merencanakannya dari awal, sehingga dapat menghemat waktu. Untuk merealisasikan pemanfaatan desain lama tersebut, dipakai suatu pendekatan *Case Based Reasoning (CBR) in Design* sampai dengan tahap penarikan (*recall*). Proses *recall* ini menggunakan teknik *Topological and Numerical Pattern Matching*. Yaitu cara mencari kesamaan hubungan tata letak (*topology*) dan kesamaan perhitungan matematis (*numeric*) antara desain baru dengan desain lama, kemudian menghitung dan mengurutkan nilai (*score*) dari masing-masing pasangan desain. Hasil akhir yang diperoleh ialah pasangan desain yang memiliki nilai kesamaan tertinggi akan di-*recall* dan ditampilkan dalam bentuk layout yang sederhana. Dengan dukungan komputer sebagai alat bantu dalam mengimplementasikan teknik di atas maka dibuat suatu *prototype software* komputer dengan nama *LEVINE system V1.0* yang akan mewujudkan pemanfaatan desain-desain lama sebagai solusi dalam proses pembuatan desain baru. Dari hasil yang diperoleh dalam pengujian terhadap *LEVINE system V1.0* dengan obyek layout ruang akomodasi kapal yang meliputi upper deck, poop deck, boat deck, bridge deck dan navigation deck, menunjukkan bahwa sistem ini mempunyai kemampuan dan konsistensi tinggi dalam pelaksanaan proses *pattern matching* antara *template* dengan data.

INSTITUT TEKNOLOGI SEPULUH NOPEMBER (ITS)

ABSTRACT

FACULTY OF OCEAN ENGINEERING

DEPARTMENT OF NAVAL ARCHITECTURE

Degree in Engineering (S 1)

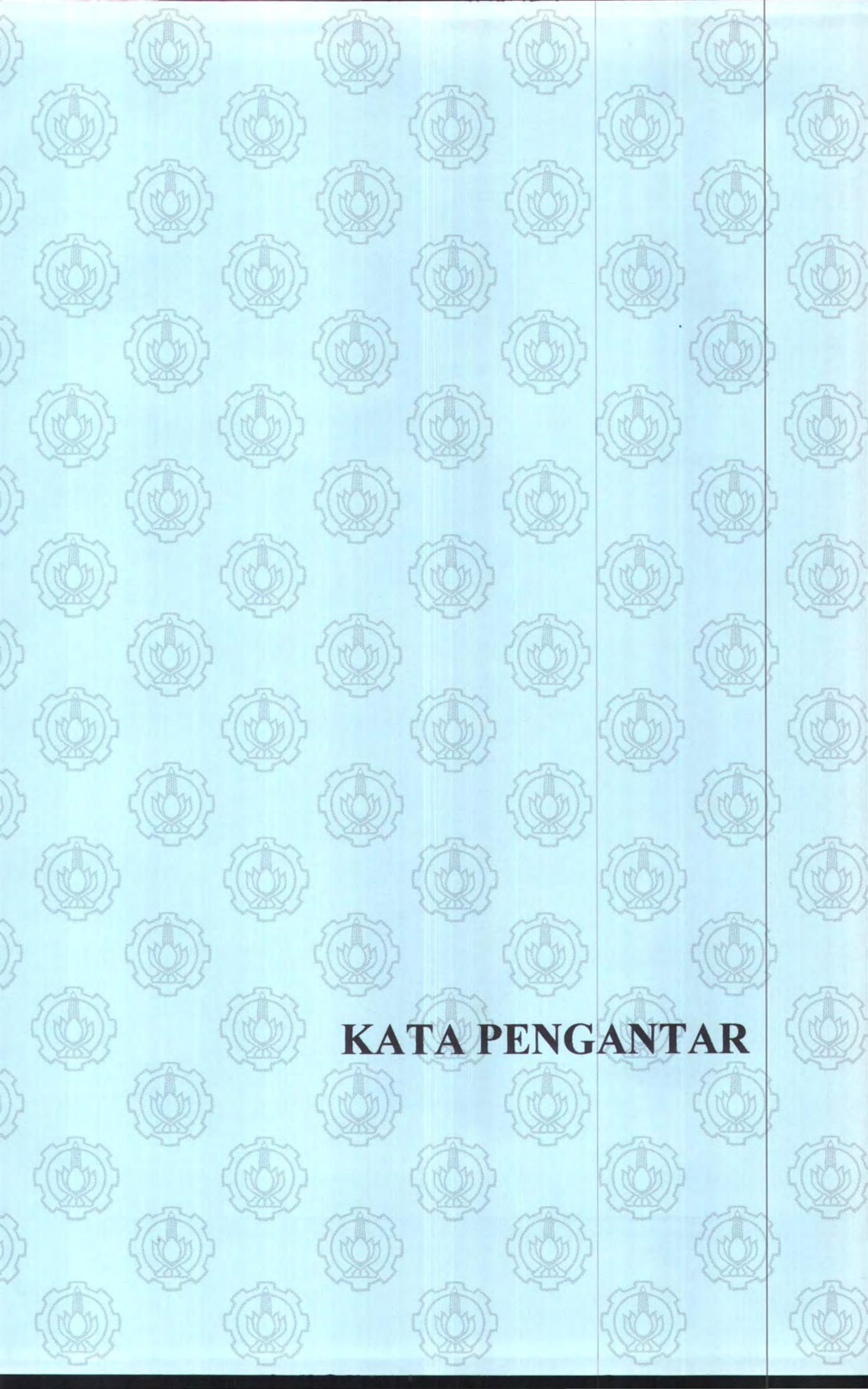
**AN ANALYSIS OF ACCOMMODATION LAYOUT RETRIEVAL
UTILIZING *TOPOLOGICAL AND NUMERICAL PATTERN MATCHING*
TECHNIQUE IN TANKER**

By: Wahyu Wiyati

First Supervisor: I. G. N. S. Buana, S.T., M.Eng.

Second Supervisor: Ir. Djauhar Manfaat, M.Sc., Ph.D.

Nowadays, Method of Comparison Ship which majority use design reuse is often used in ship design. Notably, in general arrangement that the point is accommodation and their position on the accommodation's deck. This is exceptionally beneficial to the designer because he doesn't have to plan it from scratch, so it can save time. Case Based Reasoning (CBR) in Design is used as an approach to realize the usage of design reuse. This approach is restricted up to recall and utilizing Topological and Numerical Pattern Matching technique. It is the way to find the similarity of the layout closeness relationship or topology and the similarity of the mathematic calculation or numeric between new designs and reuse one, then give score and put them in a series. The achievement is design matched, which has higher similarity score, will be recalled and presented into simple layout. LEVINE system V1.0 is a tool to implement this technique and it has been throughout the test by accommodation layout including upper deck, poop deck, boat deck, bridge deck and navigation deck. The result show us that LEVINE system V1.0 has capability and highest consistent to simulate data obtained from this study according to the pattern matching similar to manual works.



KATA PENGANTAR

KATA PENGANTAR

Assalamu'alaikum Wr. Wb.

Tiada kata yang dapat Penulis ucapkan selain puji syukur *Alhamdulillah* kehadirat Allah SWT, yang telah memberikan Rahmat, Taufik, Hidayah dan Inayah-Nya kepada Penulis, sehingga Penulis dapat menyelesaikan laporan Tugas Akhir dengan judul **“ANALISIS PEMILIHAN LAYOUT RUANG AKOMODASI MENGGUNAKAN TEKNIK *TOPOLOGICAL AND NUMERICAL PATTERN MATCHING* PADA KAPAL TANKER”**.

Adapun Tugas Akhir ini diajukan guna memenuhi salah satu syarat untuk memperoleh gelar Sarjana Teknik (S1) pada Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember, Surabaya.

Disamping itu, Penulis tak lupa menyampaikan terima kasih yang sebesar-besarnya kepada semua pihak yang telah membantu menyelesaikan Tugas Akhir ini, baik secara langsung maupun tidak langsung, terutama kepada :

- 1) Ibu, Bapak, Mbak Ika, Mbak Dwi dan Dik Naning tercinta yang telah memberikan bantuan moral maupun material yang tak pernah sarat. Doa untukmu semua selalu dalam nadiku.
- 2) Bapak Ir. Djauhar Manfaat, M.Sc., Ph.D., selaku Dosen Pembimbing yang telah meluangkan waktu untuk membantu menyelesaikan pemrograman Tugas Akhir ini.
- 3) Bapak I.G.N.S. Buana, S.T., M.Eng., selaku Dosen Pembimbing yang telah membantu menyelesaikan laporan Tugas Akhir ini.

- 4) Teman terbaik Totok yang senantiasa membantu setiap kesulitan dan memberi semangat di akhir TA ini.
- 5) Rekan seperjuangan yakni M. Thohir, Sofyan, 'Jabrix', Wahyu 'Gondrong', Gus Mat yang selalu bersama di Lab-Kom setiap malam. Perjuangan ini tiada akan pernah berakhir.
- 6) Rekan-rekan P-37, khususnya 'Unyil', Umar, Sugito, Habibur, Yosi, Bibie, Yusa, Agung 'Gendut', Lisza-Arief dan Arek-arek lantai 4 serta Mbak Arik yang selalu menemani dalam suka dan duka. Kalian semua akan selalu menjadi bagian dari hidup Penulis.
- 7) Joy dan Danny yang selalu memberi saran dan perhatian kepada Penulis.
- 8) Seluruh teman Kost di ARH 102 yang saling menolong.
- 9) Serta pihak-pihak yang tidak dapat Penulis sebutkan satu persatu.

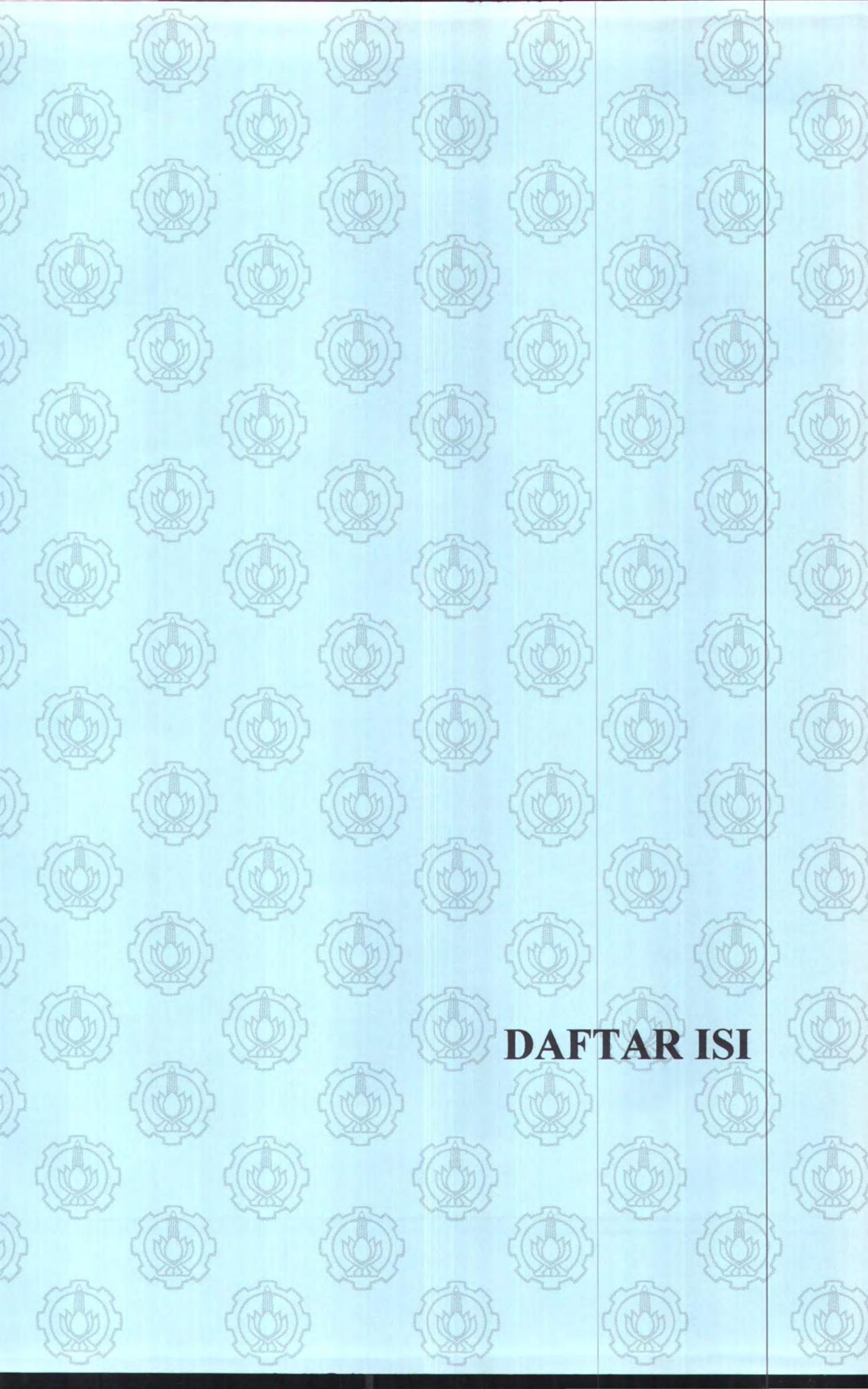
Sesuai dengan pepatah "Tiada Gading Yang Tak Retak", maka Penulis menyadari bahwa Tugas Akhir ini masih kurang sempurna. Oleh karena itu semua saran dan kritikan yang bersifat membangun akan Penulis terima dengan senang hati demi kesempurnaan Tugas Akhir ini.

Akhir kata, Penulis berharap semoga Tugas Akhir ini berguna bagi semua yang memerlukannya.

Wassalamu'alaikum Wr. Wb.

Surabaya, 28 Januari 2003

Penulis



DAFTAR ISI

DAFTAR ISI

	Halaman
HALAMAN JUDUL	
LEMBAR PENGESAHAN	
ABSTRAK	iv
KATA PENGANTAR	vi
DAFTAR ISI	viii
DAFTAR GAMBAR	xii
DAFTAR TABEL	xv
BAB I PENDAHULUAN	
1.1. Latar Belakang Masalah	I - 1
1.2. Perumusan Masalah	I - 4
1.3. Tujuan dan Manfaat	I - 4
1.4. Batasan Masalah	I - 5
1.5. Metodologi Penelitian	I - 6
1.6. Sistematika Penulisan Laporan	I - 8
BAB II TINJAUAN PUSTAKA	
2.1. Definisi Rencana Umum	II - 1
2.1.1. Definisi Ruang	II - 2
2.1.2. Ruang Akomodasi Kapal	II - 4
2.2. Layout Ruang Akomodasi Kapal	II - 7
2.3. Pemanfaatan Desain Lama (<i>Design Reuse</i>)	II - 8

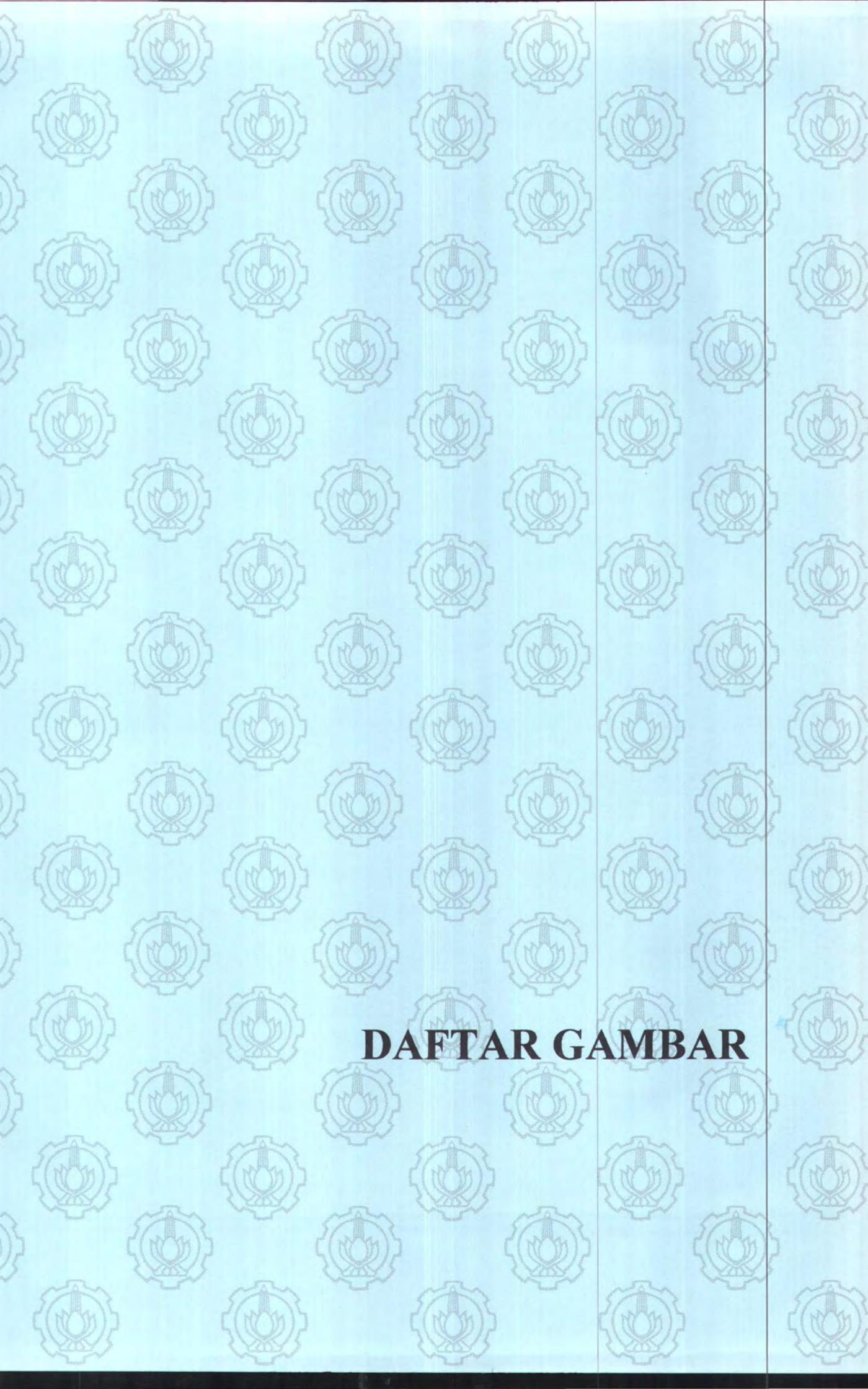
2.4. <i>Case Based Reasoning (CBR)</i>	II - 9
2.5. Pemanfaatan Komputer Untuk Menunjang	
<i>Design Reuse</i>	II-12
2.6. Representasi Kasus Desain (<i>DesignCases</i>)	II-14
2.7. Penarikan Kasus Desain	II-18
2.8. Adaptasi Kasus Desain	II-20
2.9. Pendekatan <i>Pattern Matching</i>	II-22
2.9.1. Klasifikasi <i>Pattern</i>	II-22
2.9.2. Metode <i>Matching</i>	II-25
2.9.3. Jenis Kesamaan	II-28
2.10. Sistem SPIDA	II-29
2.11. Pertimbangan Aspek Struktural dalam	
<i>Layout Design</i>	II-31
2.11.1. Penempatan Dinding/Sekat untuk	
Ruangan pada Bangunan Atas	II-31
2.11.2. Perencanaan Bukaan-bukaan	II-32
2.11.3. Sistem Konstruksi yang digunakan	
pada Ruang Akomodasi	II-32
2.11.4. Analisis Layout Ruang Akomodasi Kapal	
Berdasarkan Aspek Struktural	II-33

BAB III IMPLEMENTASI TEKNIK

3.1. Representasi Ruang	III - 1
-------------------------------	---------

3.2. Representasi Layout	III - 4
3.3. Proses <i>Pattern Matching</i>	III - 5
3.4. <i>Topological Pattern Matching</i>	III - 5
3.5. <i>Numerical Pattern Matching</i>	III - 7
3.5. Penyusunan Data	III - 8
3.5.1. Pengukuran Kesamaan Antara Contoh dan Masukan (<i>Input</i>)	III - 9
3.5.2. Normalisasi	III-14
3.5.3. Algoritma untuk Mendapatkan Ukuran Kesamaan Pola	III-16
3.6. Penyusunan Data	III-17
3.7. Prosedur Pengolahan Data	III-18
3.8. Implementasi Pendekatan <i>Case Based Reasoning</i> <i>in Design</i>	III-21
3.9. Algoritma dari <i>Topological and Numerical</i> <i>Pattern Matching</i>	III-22
BAB IV LEVINE SYSTEM V1.0	
4.1. Levine System V1.0	IV- 1
4.2. Pemodelan Sistem	IV- 2
4.3. Libraries	IV- 2
4.4. <i>Layout Viewer</i>	IV- 3
4.5. <i>Pattern Matching</i>	IV- 4
4.6. <i>Sistem Interface</i>	IV- 4

BAB V	ANALISIS HASIL	V - 1
BAB VI	KESIMPULAN DAN SARAN	
	6.1. Kesimpulan	VI - 1
	6.2. Saran	VI - 1
DAFTAR PUSTAKA		
LAMPIRAN		



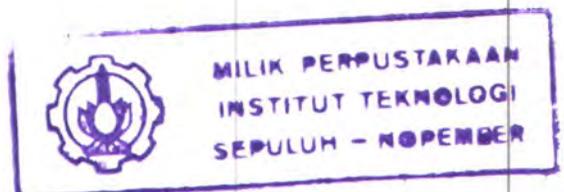
DAFTAR GAMBAR

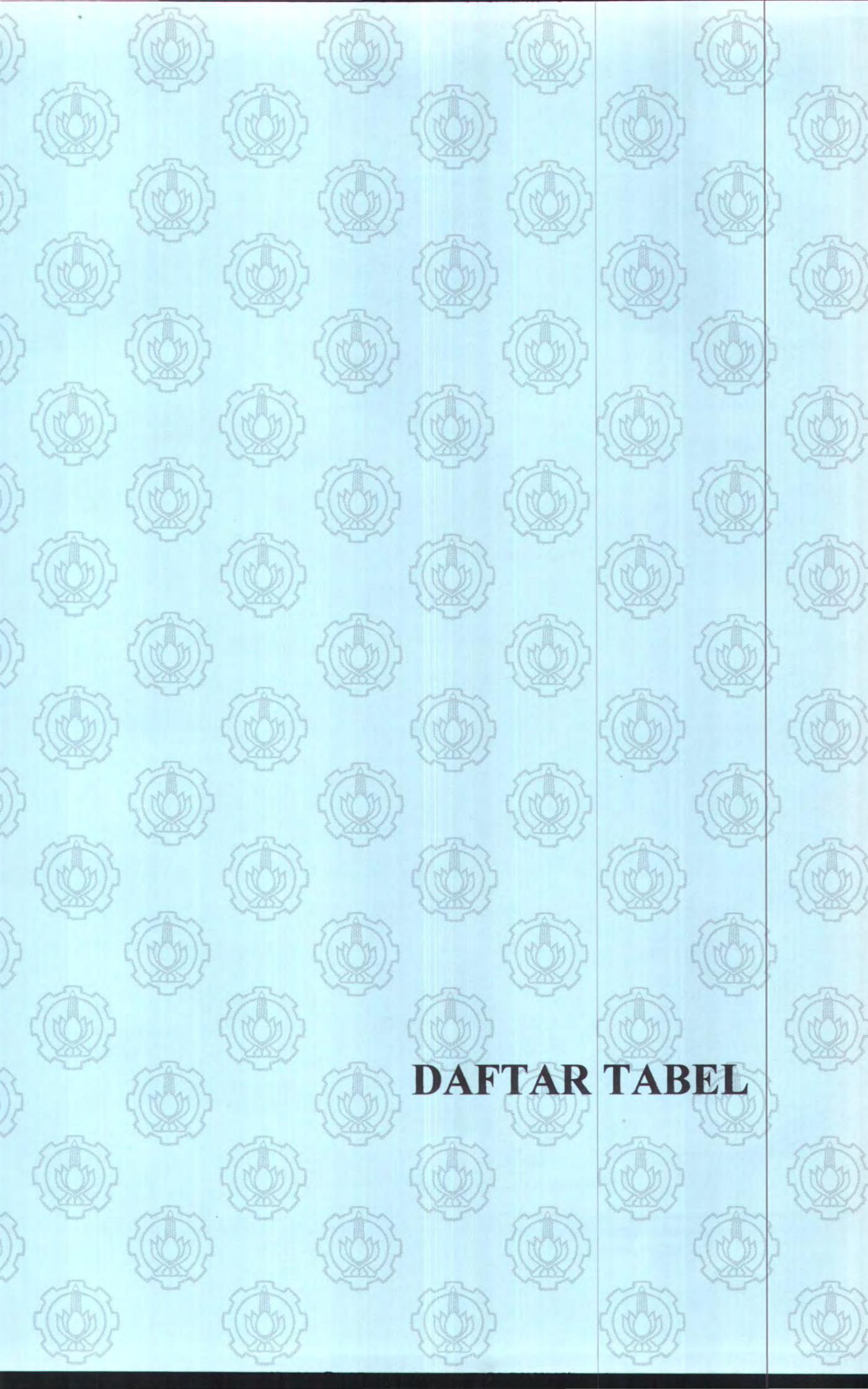
DAFTAR GAMBAR

		Halaman
Gambar	1.1. Bagan Aliran Metodologi Penelitian dan Analisis Masalah	I - 7
Gambar	2.1. Model sederhana dari <i>Case Based Reasoning</i> (CBR) (Maher dkk., 1995)	II - 12
Gambar	2.2. Tipe dasar dari <i>patterns</i> (Manfaat, 1998).....	II - 25
Gambar	3.1. Representasi ruang bentuk <i>polygon</i> (a), dan <i>rectangle</i> (b) (Manfaat, 1998)	III - 1
Gambar	3.2. Representasi sebuah <i>graph</i> (Ballard dan Brown, 1982)	III - 6
Gambar	3.3. Dua buah layout yang akan dicocokkan topologinya	III - 6
Gambar	3.4. Representasi layout dalam bentuk <i>graph</i>	III - 7
Gambar	3.5. Suatu <i>assosiation graph</i> yang dihasilkan dari <i>matching</i> dua layout dalam Gambar 3.3. dan 3.4.	III - 7
Gambar	3.6. Model kesamaan numerik antara layout input dengan layout di dalam <i>case base</i>	III - 8
Gambar	3.7. Contoh dari <i>pattern matrix</i>	III - 10
Gambar	3.8. Contoh dari <i>proximity matrix</i>	III - 11

Gambar	3.9. Metrik-metrik Minkowski (Jain dan Dubes, 1988)	III - 13
Gambar	3.10. Gambar tiga ruang yang didefinisikan kedekatannya	III - 20
Gambar	3.8. Algoritma dari <i>topological</i> dan <i>numerical</i> <i>pattern matching</i>	III - 23
Gambar	4.1. <i>Architecture</i> LEVINE System V1.0	IV - 2
Gambar	4.2. Tampilan <i>interface</i> utama (<i>main menu</i>)	IV - 4
Gambar	4.3. Menu pilihan dari tombol Deck Cases	IV - 5
Gambar	4.4. Dialog untuk menampilkan pilihan layout <i>cases</i>	IV - 6
Gambar	4.5. Pilihan layout untuk semua data layout pada geladak Upper Deck	IV - 6
Gambar	4.6. Dialog untuk menampilkan gambar layout	IV - 6
Gambar	4.7. Gambar layout yang telah dipilih	IV - 7
Gambar	4.8. Tampilan atribut untuk Quarter_Master1_ UD06s	IV - 8
Gambar	4.9. Menu pilihan dari tombol Templates	IV - 9
Gambar	4.10. Dialog untuk menampilkan pilihan layout <i>template</i>	IV - 9
Gambar	4.11. Pilihan layout untuk layout <i>template</i> pada Bridge Deck	IV - 9

Gambar	4.12. Dialog untuk menampilkan gambar layout <i>template</i>	IV - 10
Gambar	4.13. Gambar layout <i>template</i> yang telah dipilih, siap untuk diproses	IV - 10
Gambar	4.14. Pilihan proses <i>matching</i> pada suatu <i>template</i>	IV - 11
Gambar	4.15. Contoh hasil proses pilihan TOPOLOGY dengan hasil layout yang mempunyai tingkat kesamaan tertinggi secara topologi	IV - 12
Gambar	4.16. Contoh hasil proses pilihan TOPOLOGY-AND- NUMERICAL dengan hasil tiga layout yang mempunyai tingkat kesamaan tertinggi secara geometrik dan topologi	IV - 14
Gambar	4.17. Tampilan dialog untuk menutup program	IV - 14
Gambar	4.18. Isi dari About berupa penjelasan program	IV - 15





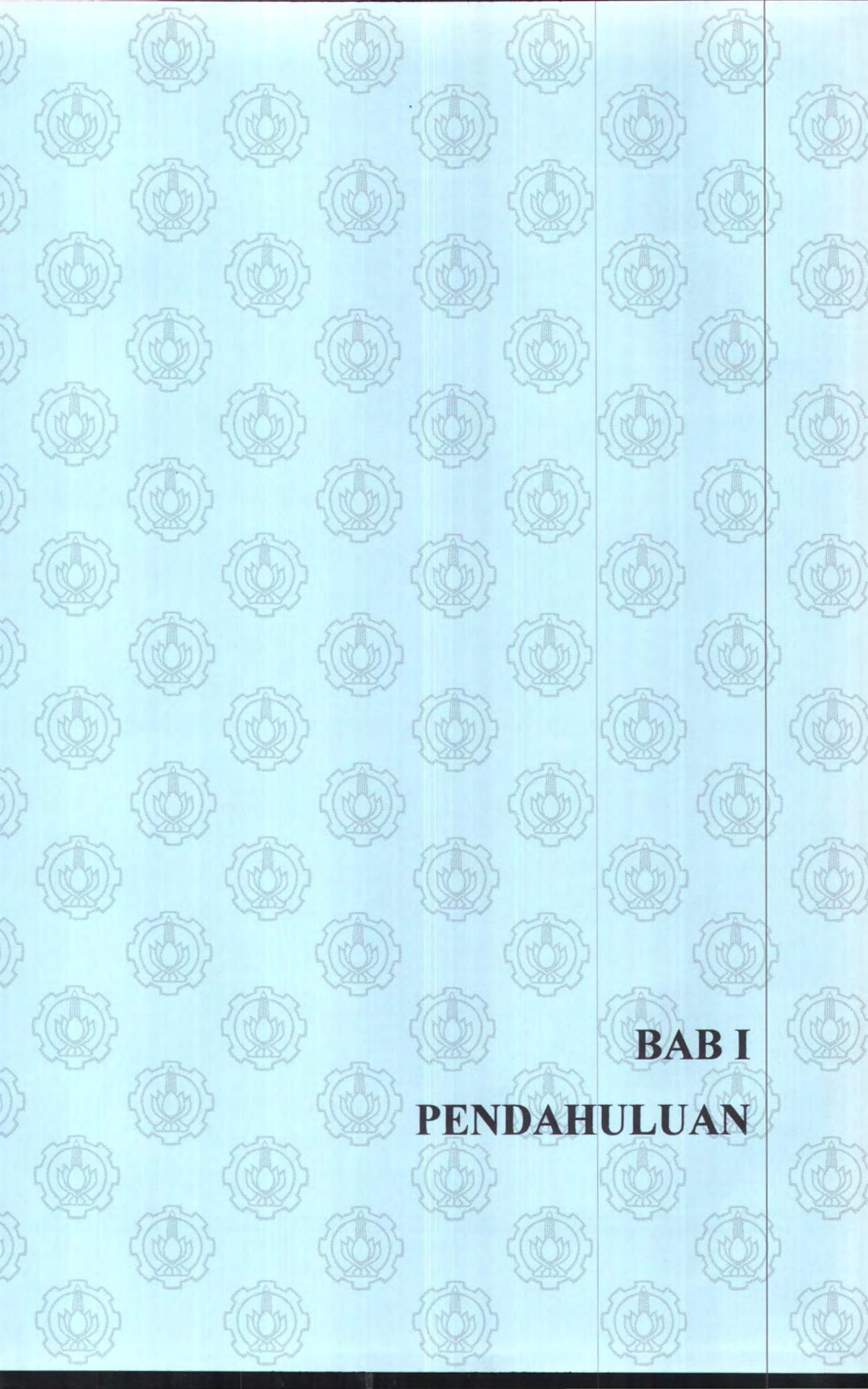
DAFTAR TABEL

DAFTAR TABEL

		Halaman
Tabel	2.1 Analisis Layout xxx Deck Berdasarkan Aspek Struktural	II-34
Tabel	3.1. Contoh urutan layout dari hasil <i>topological</i> dan <i>numerical pattern matching</i>	III-24
Tabel	5.1. Hasil <i>matching</i> antara data layout Template_06_Deck_RD dengan Ship_06_Deck_RD, Ship_04_Deck_RD dan Ship_05_Deck_RD	V - 2
Tabel	5.2. Hasil <i>matching</i> antara data layout Template_06_Deck_PD dengan Ship_06_Deck_PD, Ship_05_Deck_PD, Ship_04_Deck_PD Ship_03_Deck_PD, Ship_02_Deck_PD, dan Ship_01_Deck_PD	V - 5
Tabel	5.3. Hasil <i>matching</i> antara data layout Template_03_Deck_BD dengan Ship_03_Deck_BD, Ship_02_Deck_BD, Ship_01_Deck_BD, Ship_06_Deck_BD, Ship_05_Deck_BD dan Ship_04_Deck_BD	V - 6
Tabel	5.4. Hasil <i>matching</i> antara data layout Template_06_Deck_BD dengan Ship_06_Deck_BD, Ship_05_Deck_BD, Ship_04_Deck_BD, Ship_03_Deck_BD, Ship_02_Deck_BD dan Ship_01_Deck_BD	V - 7

Tabel 5.5. Hasil *matching* antara data layout Template_03_Deck_ND dengan Ship_06_Deck_ND, Ship_05_Deck_ND, Ship_04_Deck_ND, Ship_03_Deck_ND, Ship_02_Deck_ND dan Ship_01_Deck_ND V - 8

Tabel 5.6. Hasil *matching* antara data layout Template_03_Deck_ND dengan Ship_06_Deck_ND, Ship_05_Deck_ND, Ship_04_Deck_ND, Ship_03_Deck_ND, Ship_02_Deck_ND dan Ship_01_Deck_ND V - 9



BAB I
PENDAHULUAN



BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Pada saat ini terdapat 4 buah metode perencanaan kapal yang dikembangkan berdasarkan teori dan pengalaman, yaitu *Method of Comparison Ship*, *Method of Statistics*, *Method of Trial and Error*, dan *Method of Complex Solution*. Yang mana *Method of Comparison Ship* (Metode Kapal Pemanding) adalah yang paling sering digunakan untuk mendesain kapal baru. Dasar pemikiran digunakannya metode kapal pemanding tersebut adalah untuk merencanakan kapal yang diusahakan lebih baik dari kapal yang telah ada atau kapal pemanding (Phoels, 1982). Menurut Foz (1973); Akin (1978) dan Jones (1980) bahwa dalam membuat sebuah desain tataletak (*layout*), seorang desainer akan terhubung kembali dengan pengalaman dan keahlian yang dimiliki sebelumnya. Hal ini berarti baik sengaja maupun tidak desainer akan menggunakan kembali desain-desain lama yang telah ada (*design reuse*) untuk membuat desain baru.

Demikian halnya dalam membuat Rencana Umum (*General Arrangement*), desainer dapat memakai kembali desain-desain lama (*design reuse*) dari kapal pemanding. Khususnya pada penentuan tataletak (*layout*) ruang-ruang akomodasi dan penempatannya pada geladak akomodasi, sehingga akan lebih mudah. Karena inti dari pembuatan rencana umum adalah penentuan ruang akomodasi dan penempatannya pada geladak akomodasi (Taggart, 1980).



Pemanfaatan contoh-contoh desain lama (*design reuse*) dari kapal-kapal pemanding untuk merencanakan ruang akomodasi akan sangat menguntungkan bagi desainer. Hal ini seperti diungkapkan oleh Manfaat (1996) bahwa pemanfaatan desain-desain lama untuk proses pembuatan desain baru mempunyai beberapa keuntungan, antara lain :

a. Memperpendek proses desain

Dimana proses desain tidak dimulai dari awal karena desainer dapat menggunakan contoh-contoh desain yang sudah ada, sehingga mempercepat waktu penyelesaian dan memperpendek proses desain.

b. Mengurangi banyak tugas yang diperlukan untuk pembuatan desain, karena desainer tidak harus merancang nya dari awal (*from scratch*).

c. Mencapai hasil desain yang lebih memenuhi permintaan *owner*.

Untuk mewujudkan keuntungan diatas maka Maher, dkk. (1995) menggunakan suatu pendekatan *Case Based Reasoning* (CBR). Dimana *Case Based Reasoning* (CBR) yang diperkenalkan oleh Schank (1982); Kolodner, dkk. (1985) serta Hammond (1989) merupakan representasi dari *design reuse* yang dikumpulkan dalam suatu *case base* dan dua buah proses, yaitu penarikan (*recall*) serta proses modifikasi atau adaptasi (*adapt*).

Dalam konteks tugas akhir ini, penggunaan CBR untuk memilih desain *layout* ruang akomodasi kapal tanker dibatasi sampai pada proses penarikan (*recall*). Proses *recall* tersebut menggunakan teknik *topological and numerical pattern matching*. Sedangkan untuk mengimplementasikan proses *recall* tataletak ruang akomodasi kapal tanker tersebut akan dibuat sebuah *prototype* komputer,



yang akan memberikan urutan desain yaitu dari desain yang mempunyai kesamaan tertinggi sampai terendah. Selanjutnya desain dengan tingkat kesamaan tertinggi yang akan dijadikan sebagai pemecahan awal (patokan) dalam pembuatan desain tataletak (*layout*) ruang akomodasi kapal tanker yang baru.

Penggunaan komputer dalam proses *recall* mengingat bahwa sebagai manusia, desainer mempunyai keterbatasan-keterbatasan. Salah satu keterbatasan tersebut adalah keterbatasan ingatan, yang mengakibatkan tidak semua desain terkoordinasi dengan baik. Disini komputer dapat digunakan sebagai alat (*tools*) untuk merealisasikan hal tersebut. Keuntungan yang didapat dengan pemanfaatan komputer untuk menunjang *desain reuse* adalah:

- 1). Membantu dalam menyimpan dan merepresentasikan desain lama.
- 2). Mendukung pemberian *indeks* pada desain.
- 3). Membantu dalam mengelompokkan desain.
- 4). Membantu pencarian (*searching*) dari desain lama yang sesuai dengan permasalahan desain (*new problem*).
- 5). Mendukung kesesuaian pola antara permasalahan desain dengan desain lama.
- 6). Membantu dalam pengukuran kesamaan antara *new problem* dengan desain lama.
- 7). Mendukung dalam pemilihan desain lama yang mempunyai kesamaan terbanyak dengan permasalahan desain.

Keuntungan di atas, sangat jelas bahwa pemanfaatan desain lama yang didukung komputer dapat mengurangi beban kerja desainer.



1.2. Perumusan Masalah

Berdasarkan uraian diatas, pokok permasalahan yang harus dipecahkan dalam tugas akhir ini adalah :

- 1). Bagaimana mengimplementasikan *layout* ruang akomodasi kapal tanker ke dalam *interface* komputer yang menarik dan mudah dimengerti oleh pemakai.
- 2). Bagaimana mempresentasikan contoh-contoh *layout* ruang akomodasi kapal tanker yang sudah ada ke dalam suatu *data base* (*case base*).
- 3). Bagaimana menentukan kesamaan antara desain permintaan *owner* dengan desain-desain yang ada didalam *case base* menurut teknik *topological and numerical pattern matching*.
- 4). Bagaimana memberi nilai (*scoring*) derajat kesamaan antara desain-desain didalam *case base* dengan desain permintaan pemesan (*owner*).
- 5). Bagaimana memilih salah satu (atau lebih) desain didalam *case base* yang mempunyai nilai (*score*) derajat kesamaan tertinggi sebagai desain awal yang sesuai dengan permintaan *owner*.

1.3. Tujuan dan Manfaat

Penulisan tugas akhir ini bertujuan untuk memperbaiki proses desain tataletak (*layout*) ruang akomodasi kapal tanker dari desainer, sehingga diperoleh desain baru yang sesuai keinginan pemesan dalam waktu yang cepat. Tujuan utama ini dicapai melalui tujuan khusus, antara lain :

- 1). Untuk mempresentasikan desain tataletak ruang akomodasi kapal tanker yang sudah ada ke dalam *case base*.



- 2). Untuk mengembangkan teknik *topological and numerical pattern matching* dalam pemilihan desain tata letak ruang akomodasi kapal tanker.
- 3). Mengimplementasikan teknik-teknik diatas menjadi sebuah *prototype* komputer.

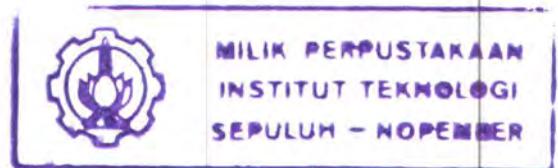
Sedangkan manfaat yang dapat diperoleh dari hasil proses *pattern matching* ini adalah sebagai berikut :

- 1). Mempercepat proses perencanaan suatu kapal, karena perancangan desain tataletak ruang akomodasi kapal membutuhkan waktu yang relatif lebih singkat.
- 2). Menghemat biaya produksi dalam pembuatan sebuah kapal.
- 3). Mengurangi tugas desainer, karena ia tidak perlu merancang dari awal.
- 4). Desain yang dihasilkan lebih sesuai (*match*) dengan desain permintaan pemesan.

1.4. Batasan Masalah

Untuk lebih mengarah pada pokok permasalahan dan mempersempit permasalahan maka perlu adanya batasan masalah, yaitu :

- 1). Data *layout* ruang akomodasi kapal yang digunakan untuk obyek studi adalah kapal Tanker dengan DWT 1000 – 6500 ton.
- 2). Aturan-aturan ruang akomodasi kapal tanker mengikuti aturan *International Labour Conference* (ILO).
- 3). Pendekatan *Case Based Reasoning* (CBR) dalam memilih ruang akomodasi kapal tanker dibatasi sampai dengan tahap penarikan hasil (*recall*).





- 4). Pada proses *matching* yang diperhatikan adalah kesamaan hubungan tataletak (*topology*) dan data numerik dari desain didalam *case base* menggunakan teknik *topological and numerical pattern matching*.

1.5. Metodologi Penelitian

Metode yang digunakan dalam penyusunan tugas akhir ini meliputi :

- 1). Library Research

Yaitu melakukan studi literatur dengan tujuan merangkum teori-teori dasar dalam merencanakan ruang akomodasi kapal tanker dan teori mengenai CBR serta acuan dalam penulisan tugas akhir ini.

- 2). Perumusan latar belakang

Pada tahap ini dilakukan perumusan latar belakang yang berkaitan dengan permasalahan yang ingin dipecahkan.

- 3). Hipotesis Awal

Yaitu tahap melakukan dugaan awal terhadap hasil akhir yang akan diperoleh didalam tugas akhir ini.

- 4). Perumusan permasalahan

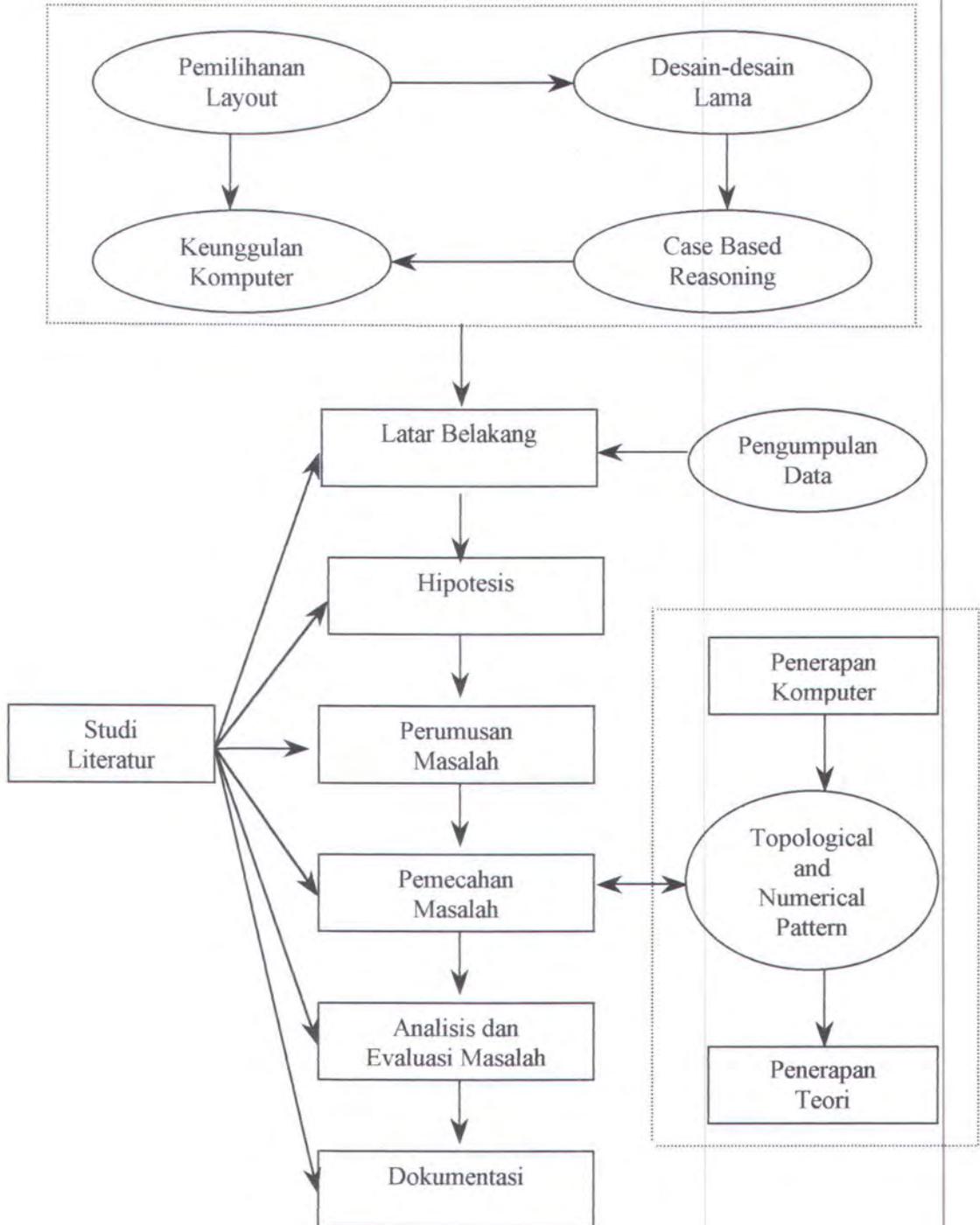
Pada tahap ini dilakukan perumusan masalah berdasarkan *library research*, latar belakang dan hipotesis awal yang telah dilakukan sebelumnya.

- 5). Pemecahan masalah

Tahap ini adalah mencari solusi terhadap permasalahan-permasalahan yang telah dirumuskan sebelumnya.



Aliran Metodologi Penelitian Dan Analisis Masalah



Gambar 1.1. Bagan Aliran Metodologi Penelitian dan Analisis Masalah.



6). Evaluasi

Yaitu tahap mengevaluasi hasil dari *pattern matching*. Kemudian menggunakannya sebagai pilihan desain yang sesuai dengan permintaan *owner*.

7). Dokumentasi

Merupakan tahap terakhir dalam tugas akhir ini, yaitu tahap penulisan tugas akhir terhadap proses penelitian yang telah dilakukan di atas.

1.6. Sistematika Penulisan Laporan

Untuk memperoleh hasil laporan tugas akhir yang sistematis dan tidak menyimpang dari pokok permasalahan maka dibuat sistematika sebagai berikut :

BAB I PENDAHULUAN

Bab ini meliputi latar belakang masalah, perumusan masalah, tujuan dan manfaat, batasan masalah, metodologi penelitian serta sistematika penulisan laporan tugas akhir.

BAB II TINJAUAN PUSTAKA

Bab ini berisi tentang tinjauan pustaka atau dasar teori mengenai konsep perancangan ruang akomodasi kapal tanker. Selain itu juga mencakup konsep *Case Based Reasoning (CBR) in design* untuk menyelesaikan permasalahan yang ada.

BAB III TEKNIK IMPLEMENTASI

Bab ini meliputi tahapan proses perencanaan tata letak ruang akomodasi kapal tanker, mulai dari data awal, penyamaan



hubungan antar ruang akomodasi dan hubungan numeriknya serta metode yang digunakan dalam memilih desain tata letak ruang akomodasi sesuai permintaan pemesan.

BAB IV IMPLEMENTASI SISTEM

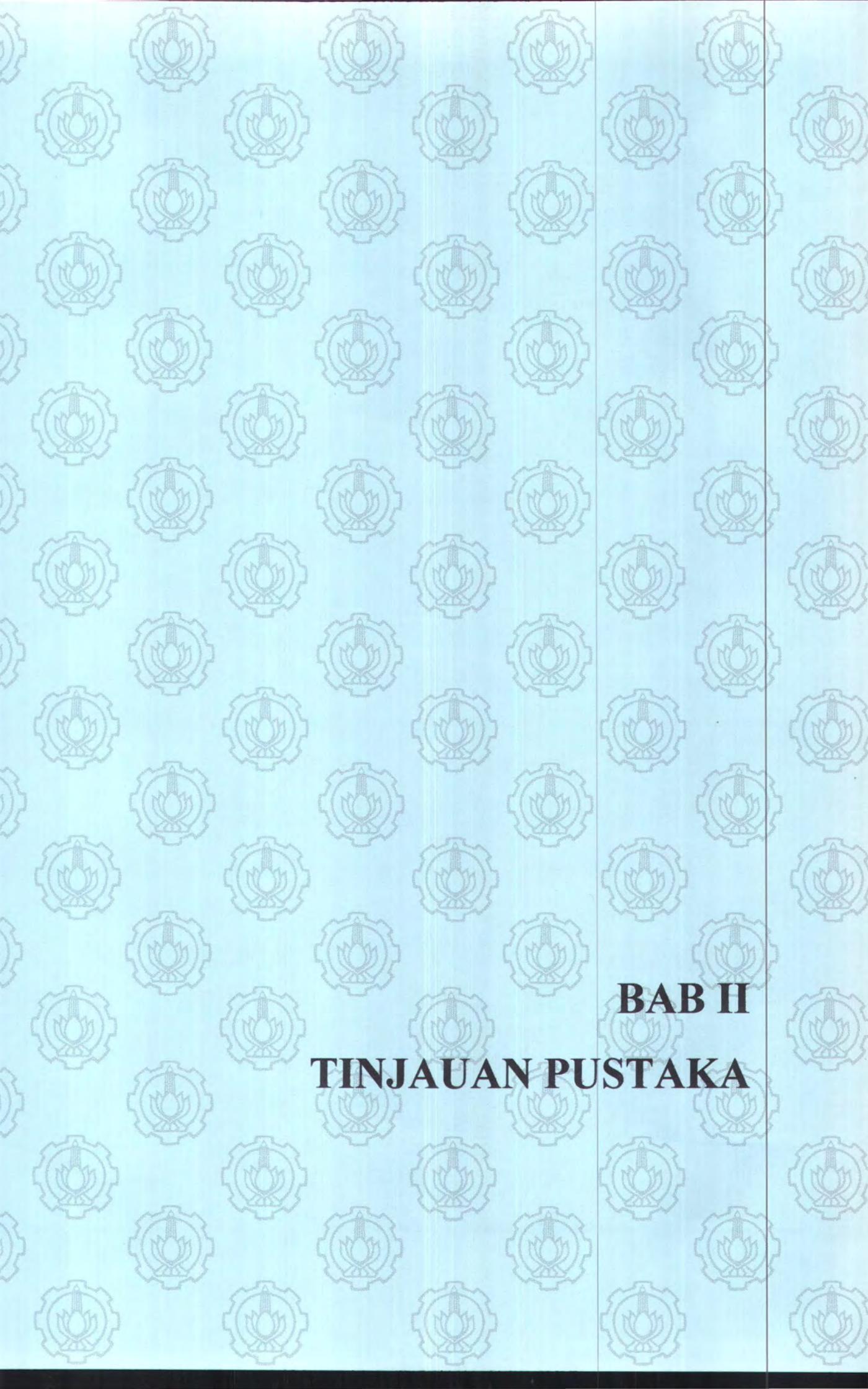
Bab ini berisi tentang penjelasan sistem yang digunakan dalam proses pattern matching.

BAB V ANALISIS HASIL

Bab yang membahas mengenai hasil dari program yang telah dibuat, yaitu apakah hasil yang diperoleh sesuai dengan apa yang diinginkan atau tidak.

BAB VI KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan yang diperoleh dari penyelesaian masalah dan tujuan yang ingin dicapai serta saran-saran terhadap upaya perbaikan dalam menyelesaikan masalah.



BAB II
TINJAUAN PUSTAKA



BAB II

TINJAUAN PUSTAKA

2.1. Definisi Rencana Umum

Tapscott (1980) mendefinisikan Rencana umum suatu kapal (*General Arrangement*) sebagai pemberian ruang untuk semua fungsi dan peralatan yang disyaratkan, pengkoordinasian yang tepat untuk lokasi dan akses. Dan seperti telah dijelaskan pada bab sebelumnya bahwa inti dari pembuatan rencana umum adalah penentuan ruang akomodasi dan penempatannya pada geladak akomodasi (Taggart, 1980).

Selain itu Tapscott (1980) juga mengemukakan bahwa ada empat langkah yang berurutan, yang harus dilakukan dalam rangka membuat rencana umum sebuah kapal, yaitu:

- 1). Penyediaan ruang utama
- 2). Pemberian batas ruangan (sekat) pada ruang pribadi
- 3). Pemilihan dan penempatan peralatan didalam ruangan serta pemasangan perabot pada dinding pembatas
- 4). Penyediaan akses yang saling menghubungkan antar ruang (pintu)

Dimana langkah-langkah tersebut diatas dimulai dari hal yang menyeluruh, meliputi penempatan ruangan-ruangan untuk ABK, tangki-tangki, ruang muat, dan sebagainya, sampai hal yang terperinci seperti perencanaan dari sebuah ruangan ABK, sirkulasi udara pada ruang muat dan sebagainya.



2.1.1. Definisi Ruang

Kamus Besar Bahasa Indonesia (1990) memberikan dua pengertian terhadap kata “ruang”. Yang pertama, ruang adalah rongga yang terbatas atau terlengkup oleh bidang. Sedangkan yang kedua, ruang adalah rongga yang tidak terbatas, tempat segala yang ada. Secara harfiah “ruang” dapat diartikan sebagai alam semesta yang dibatasi oleh atmosfer dan tanah dimana kita berpijak, sedangkan secara sempit “ruang” berarti suatu kondisi yang dibatasi empat dinding, yang dapat diraba dan dirasakan keberadannya (Suptandar, 1999).

Penempatan bidang pembatas pada keempat sisi ruang bisa menimbulkan kesan bahwa ruang terasa sempit, luas, lebar, menyenangkan, menakutkan, formal dan sebagainya. Ruang adalah unsur penting dalam desain sebagai tempat kehidupan manusia dalam melakukan tugas dan kewajibannya. Ruang bagi manusia adalah segala-galanya sebagai tempat tinggal, sebagai status diri dan status sosial.

Menurut Suptandar (1999) secara garis besar ruang dibedakan atas :

1). Ruang nyata

Yaitu ruang yang dapat diukur secara nyata dan bisa dirasakan keberadaannya karena bentukan dari beberapa bidang atau komponen tertentu. Ruang nyata dibagi menjadi dua yaitu ruang terbuka dan ruang tertutup. Ruang yang berhubungan langsung dengan bagian luar disebut ruang terbuka.

2). Ruang abstrak

Yaitu ruang yang tidak ada batasannya dan tidak ada fakta yang nyata dan tidak mudah dipahami secara visual oleh setiap orang. Implikasi



permasalahan membutuhkan kesadaran, kejelian atau kearifan manusia yang bersangkutan oleh karena ruang abstrak tidak nyata dan tidak dapat diukur, tetapi dapat dikhayalkan keberadannya.

Goodban (1979) telah mengklasifikasikan ruangan untuk hunian menjadi tiga golongan berdasarkan fungsinya, yaitu :

1). Area pemukiman (*living area*)

Merupakan daerah yang digunakan sebagai ruangan-ruangan yang berfungsi untuk tempat bersosialisasi antara anggota keluarga dan untuk melakukan kegiatan-kegiatan yang bersifat rekreasi.

2). Area peristirahatan (*sleeping area*)

Merupakan daerah yang digunakan oleh manusia untuk beristirahat secara total, artinya waktu istirahat yang digunakan lebih lama. Yang termasuk dalam golongan ini adalah ruang tidur, kamar mandi dan penyimpanan pribadi.

3). Area pelayanan (*service/working area*)

Merupakan daerah yang berhubungan dengan penyediaan dan penyimpanan makanan. Selain itu area ini juga berfungsi sebagai tempat penyimpanan peralatan pemeliharaan rumah atau penyimpanan benda-benda yang bersifat umum. Contoh dari ruangan ini adalah dapur, baik dapur kering maupun dapur basah, gudang peralatan perawatan rumah dan gudang penyimpanan perlengkapan hobi.



2.1.2. Ruang Akomodasi Kapal

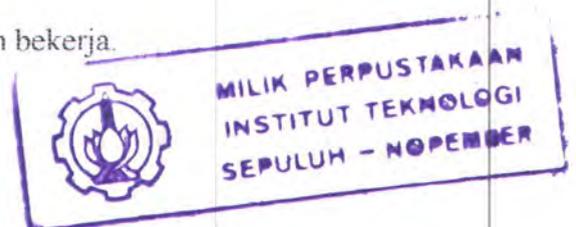
Kapal merupakan sebuah struktur yang berhubungan dengan manusia sebagai pelaksana operasionalnya, seperti halnya tempat tinggal atau hunian atau rumah. Demikian juga dengan ruang akomodasi di kapal, yang diartikan sebagai ruangan-ruangan yang digunakan oleh Anak Buah Kapal (ABK) untuk tinggal di kapal selama kapal berlayar. Dimana didalam ruang akomodasi tersebut ABK saling berinteraksi dan berhubungan satu sama lain. Selain itu, ruang akomodasi di kapal juga didefinisikan sebagai ruangan yang digunakan untuk ruangan publik, koridor, kamar mandi, kabin, kantor, klinik, ruang rekreasi dan lain-lain.

Ruang akomodasi ABK merupakan sebuah lingkungan dimana ABK saling berinteraksi dan berhubungan satu sama lain. Sehingga hal yang perlu diperhatikan dalam perencanaan ruang akomodasi kapal adalah keterkaitan antar individu, rekan kerja dan lingkungan tempat bekerja. Untuk itu perencanaan ruang akomodasi di kapal perlu memperhatikan hal-hal yang berhubungan dengan aspek psikologis sosial, yakni aspek yang menunjukkan bahwa suatu lingkungan sangat berpengaruh terhadap tingkah laku seseorang.

Aspek-aspek yang perlu diperhatikan dalam perencanaan ruang akomodasi kapal adalah (Smith dan Hatfield, 1977) :

1). Perlindungan dan Keamanan (*Shelter and security*)

Aspek ini merupakan aspek yang paling penting dan merupakan dasar kebutuhan dari tiap manusia. Dengan adanya keamanan maka seseorang akan merasa nyaman untuk beraktifitas maupun bekerja.





2). Hubungan Sosial (*Social contact*)

Perencanaan ruang akomodasi harus dapat memaksimalkan kontak antar individu agar terjadi interaksi dalam sebuah kelompok yang memiliki kesamaan minat dan kepentingan. Sehingga yang perlu diperhatikan adalah pemenuhan kebutuhan antara pribadi dan kontak sosial bagi individu.

3). Simbol Identifikasi (*Symbolic identification*)

Dilakukan untuk menunjukkan perbedaan tingkat social. Hal ini dapat dilakukan dengan pengaturan tata letak, sehingga tiap tingkatan ABK memiliki daerah sendiri.

4). Alat Bantu kerja (*Task instrumentality*)

Dalam perencanaan, hal yang perlu diperhatikan adalah efektifitas dalam bekerja. Hal ini dapat dicapai dengan kemudahan akses ke tempat kerja, pencahayaan dan mechanical handling.

5). Keindahan dan Kesenangan (*Beauty and pleasure*)

Akomodasi di kapal hendaknya juga memiliki aspek keindahan. Selain itu ruangan-ruangan di kapal dapat memenuhi kebutuhan ABK untuk berekreasi.

6). Perkembangan (*Growth and development*)

Pada aspek ini desain harus menciptakan aktifitas alternatif sehingga dapat memaksimalkan kontak social dan menyebabkan perkembangan antar individu.



Ruangan akomodasi kapal secara umum dapat dibagi dalam :

- 1). Ruang umum, adalah bagian dari ruang akomodasi kapal yang digunakan sebagai ruang makan, ruang duduk atau pada dasarnya adalah ruangan yang berfungsi sebagai tempat-tempat umum.
- 2). Ruang saniter, yaitu ruangan yang digunakan untuk :
 - Mandi (Kamar berendam, kamar mandi dengan memakai shower)
 - Kamar-kamar kecil (biasanya satu kamar dengan kamar mandi)
 - Kamar penatu
 - Kamar cuci
 - Fasilitas-fasilitas saniter lainnya.
- 3). Ruang private, yaitu ruang pribadi ABK yang biasanya terdiri dari ruang tidur dan ruang santai, atau ruang tidur dan ruang santai menjadi satu atau ruang tidur saja.
- 4). Kantor, yang merupakan ruangan yang digunakan khusus sebagai kelengkapan Administrasi baik yang berhubungan dengan kapal itu sendiri maupun yang berasal dari luar kapal.
- 5). Lorong-lorong, yaitu yang digunakan untuk jalan-jalan yang menghubungkan satu tempat ke tempat lainnya.
- 6). Ruang makan dan ruang persiapan makan. Ruangan ini biasanya harus ada pada setiap kapal dan ruangan ini terdiri dari :
 - Galley (dapur), yaitu ruangan untuk keperluan masak.
 - Pantry, yaitu ruangan yang digunakan untuk keperluan mempersiapkan makanan yang akan dihidangkan.



- Messroom, yaitu ruangan yang digunakan untuk makan-makan dan minum-minum.
- 7). Ruangan rekreasi dan sport, yaitu ruangan yang digunakan oleh para ABK untuk berolah raga atau untuk bersantai.

2.2. Layout Ruang Akomodasi Kapal

Layout (tataletak) dapat didefinisikan sebagai pengaturan/peletakan *subspaces* dalam satu *space* (Manfaat, 1992). Atau dalam tata letak ruang akomodasi kapal, tata letak dapat didefinisikan sebagai pengaturan ruangan-ruangan akomodasi dalam suatu geladak akomodasi.

Banyak prosedur yang dibangun dalam merancang suatu tata letak. Dalam perkembangannya teori perencanaan tata letak saat ini banyak menggunakan metode *Systematic Layout Planning* (SLP), yang dikembangkan oleh Muther (Francis, 1992).

Adapun persyaratan-persyaratan yang dipertimbangkan dalam perancangan tata letak ruang akomodasi ABK adalah :

1). Aliran pergerakan ABK

Dalam pembuatan tata letak ruang akomodasi kapal, aliran pergerakan ABK harus baik dan teratur agar tidak mengganggu aktifitas dari ABK.

2). Hubungan aktifitas ABK

Hubungan antar ABK dari segi aktifitas atau tugas yang harus dilakukan juga harus diperhatikan.



3). Penentuan kebutuhan dan ketersediaan ruangan

Kebutuhan minimum ruang akomodasi untuk semua ABK dapat ditentukan dengan memperhatikan ketentuan-ketentuan yang telah ditetapkan baik dalam ILO maupun *Merchant Shipping Regulation*. Selanjutnya dengan memperhatikan kebutuhan minimum ruang dan luas daerah total yang ada maka kita dapat mengalokasikan luas untuk masing-masing ruang.

4). Pertimbangan modifikasi

Sebuah kapal memiliki peluang untuk dikembangkan setelah jangka waktu tertentu mengikuti perkembangan dalam bidang perkapalan termasuk dalam tata letak ruang akomodasi. Untuk itu tata letak yang akan direncanakan diusahakan memiliki kemudahan dalam pemodifikasian.

5). Batasan-batasan

Dari peraturan yang digunakan (*Merchant Shipping Regulation* dan ILO), maka akan dapat dilihat batasan-batasan terhadap ruangan-ruangan maupun penempatan ruangan tersebut. Hal ini dijadikan pertimbangan dalam menyusun tata letak.

2.3. Pemanfaatan Desain Lama (*Design Reuse*)

Dalam membuat sebuah desain tata letak, seorang desainer akan terhubung kembali dengan pengalaman dan keahlian yang dimiliki sebelumnya (Foz, 1973; Akin, 1978; Jones, 1980). Hal ini berarti desainer baik secara sengaja atau tidak akan menggunakan desain-desain lama yang telah ada (*design reuse*) untuk membuat sebuah desain baru.



Menurut Manfaat (1996), pemanfaatan desain-desain lama untuk diproses menjadi desain baru mempunyai beberapa keuntungan :

1). Memperpendek proses desain

Dalam hal ini proses desain tidak dimulai dari awal, sehingga dapat mempercepat waktu perancangan.

2). Mengurangi tugas yang diperlukan untuk pembuatan desain, karena desainer tidak perlu merancang desain dari awal (*from scratch*).

3). Mencapai hasil desain yang lebih memenuhi permintaan pemesan (*owner*) atau persyaratan.

Pendekatan yang dapat merealisasikan pemanfaatan desain lama adalah *Case Based Reasoning* (CBR) yang akan dibahas dalam sub bab 2.4.

2.4. *Case Based Reasoning* (CBR)

Case Based Reasoning merupakan salah satu bentuk kecerdasan buatan (*Artificial Intelligence*) yang diperkenalkan oleh J.F. Gero dan Sudweeks. Dalam tugas akhir ini penulis akan menggunakan pendekatan *Case Based Reasoning in Design* (Maher, dkk., 1995). *CBR in Design* dipertimbangkan sebagai model desain yang difokuskan pada penggunaan kembali pengalaman desain yang merupakan bentuk dari beberapa episode yang mengacu pada situasi-situasi yang telah ada sebelumnya (Maher dan Zhang, 1991). Dalam model ini, pertimbangan analogi digunakan untuk memilih dan merubah desain-desain lama menjadi solusi-solusi untuk permasalahan desain yang baru. Sebagai contoh, pada



pembuatan desain interior ruang akomodasi kapal, salah satu bentuk akan mengingatkan pada desain interior dari rumah.

Langkah-langkah dalam pendekatan *Case Based Reasoning* adalah sebagai berikut:

1). Adanya permasalahan desain baru (*new problem*)

Permasalahan desain baru ini merupakan input dari *user* yang berupa gambar desain layout akomodasi ABK.

2). Desainer mengumpulkan desain-desain yang sudah ada (*design reuse*) dalam *case base*.

3). Tahap penarikan salah satu desain di dalam *case base* (*recall*).

Dalam tahap ini dilakukan penarikan salah satu desain lama dalam *case base* yang sesuai dengan desain yang diinginkan *owner* (*new problem*). Sedangkan tahapan dalam *recall* adalah :

a). Pencarian (*Searching*).

Merupakan salah satu proses pencarian desain-desain yang sesuai dengan keinginan *owner* (*user input*) dari dalam *case base*.

b). Mencari kesamaan (*Matching*).

Dalam tahap ini dilakukan pencarian kesamaan antara permasalahan desain (*new problem*) dengan desain yang diperoleh dari tahap *searching*.

c). Penilaian kesamaan (*Measuring Similarities*).

Setelah didapatkan kesamaan-kesamaan dari permasalahan desain (*new problem*) dengan desain-desain lama (*design reuse*) yang dihasilkan dari tahap *searching*, selanjutnya dilakukan pengukuran/penilaian kesamaan,



yaitu pemberian *score* dari desain-desain yang mempunyai kesamaan tersebut. Desain dengan *score* tertinggi merupakan desain yang mempunyai kesamaan terbanyak dengan permasalahan desain.

d). Penarikan (*Retrieving*).

Dalam tahap ini dilakukan proses penarikan desain-desain yang mempunyai *score* kesamaan tertinggi yang dihasilkan dari tahap *measuring similarities*.

e). Pemilihan (*Selecting*)

Proses yang terakhir dari tahap *recall* ini merupakan pemilihan desain (*selection*), dimana dalam hal ini desainer tidak harus memilih desain yang mempunyai *score* kesamaan tertinggi yang dihasilkan dari proses *retrieving*.

4). *Adaptation*, yaitu tahap memodifikasi desain yang sudah dihasilkan dari tahap *recall*.

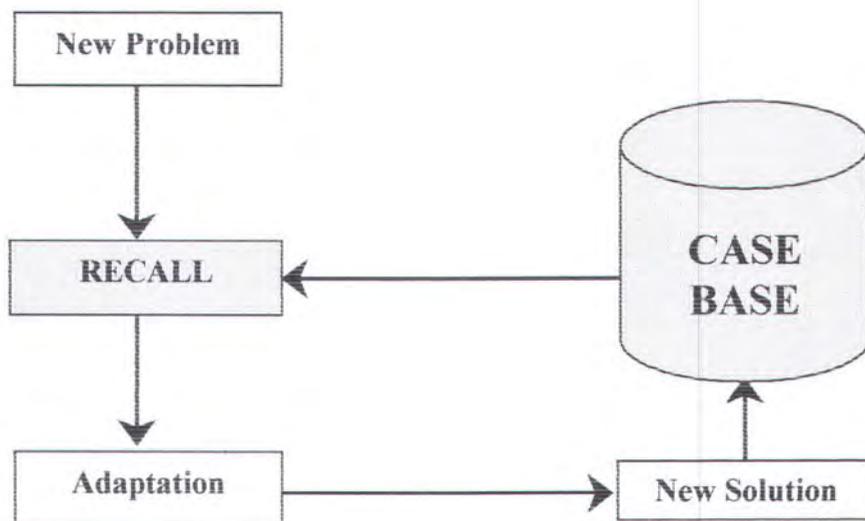
Pada proses adaptasi ini, dibuat keputusan bagian mana dari desain lama yang perlu diubah dan bagian mana yang dibiarkan tetap sama (Maher, dkk., 1995).

5). Didapatkan *new solution* yang sesuai dengan keinginan pemesan (*owner*).

New solution ini dapat dipakai/dimasukkan dalam *case base* sebagai referensi untuk siklus proses berikutnya.

Gambar 2.1. merupakan ilustrasi sederhana dari pemodelan CBR (Maher, dkk., 1995). Dalam gambar ini, model CBR mempunyai representasi desain-desain lama (*case base*) dan dua buah proses, yaitu proses penarikan (*recall*) dan

proses modifikasi/adaptasi desain lama (*adapt*). Secara singkat, dalam model ini jika diberikan masalah baru (*new problem*) sebagai input, maka desain lama akan ditarik dan selanjutnya akan diadaptasi untuk menghasilkan desain baru (*new solution*). Desain baru tersebut kemudian dapat disimpan kedalam *case base* sebagai desain lama untuk siklus proses desain berikutnya. Representasi dari desain lama dalam *case base* disediakan sebagai dasar untuk penggunaan kembali desain lama dengan sistem CBR dan bagaimana pertimbangan dapat dilakukan. Dengan kata lain, efisiensi dari proses penarikan (*recall*) dari desain lama tergantung dari bagaimana desain-desain lama tersebut direpresentasikan dalam *case base*.



Gambar 2.1. Model sederhana *Case Based Reasoning* (Maher dkk. 1995).

2.5. Pemanfaatan Komputer Untuk Menunjang *Design Reuse*

Dalam pemanfaatan desain lama yang telah ada (*design reuse*) desainer biasa menggunakan cara manual, yaitu dengan mengandalkan ingatan dari



desainer. Hal ini mempunyai kekurangan tidak semua desain dapat terkoordinasi dengan baik karena adanya keterbatasan ingatan desainer. Disini komputer dapat digunakan sebagai *tools* untuk merealisasikan hal tersebut, yaitu dengan menggunakan memori komputer untuk menyimpan desain-desain lama dan desainer dapat memanggilnya setiap waktu.

Adapun pemanfaatan komputer untuk menunjang *design reuse* dapat diuraikan sebagai berikut :

- 1). Membantu dalam penyimpanan dan representasi desain lama
- 2). Mendukung pemberian indeks pada desain
- 3). Membantu dalam pengelompokan desain
- 4). Membantu pencarian (*searching*) dari desain lama yang sesuai dengan permasalahan desain (*new problem*).
- 5). Mendukung kesesuaian pola antara permasalahan desain dengan desain lama
- 6). Membantu dalam pengukuran kesamaan antara *new problem* dengan desain lama
- 7). Mendukung dalam pemilihan desain lama yang mempunyai kesamaan terbanyak dengan permasalahan desain.

Dengan adanya dukungan komputer seperti diuraikan diatas, maka pemanfaatan komputer memberikan banyak keuntungan bagi desainer untuk menyelesaikan pembuatan desainnya. Desainer tidak perlu bingung dengan arsip desain yang dimilikinya, karena telah tersimpan dalam komputer dan jika suatu waktu dibutuhkan dapat ditampilkan secara cepat, rapi dan akurat. Selain itu komputer dapat membantu dalam pemilihan kesamaan antara desain input dengan



desain lama yang tersedia. Program komputer (matematis, simulasi dan sebagainya) merupakan alat yang sangat berguna dan mempunyai kemampuan besar dalam membandingkan pilihan susunan suatu daerah kegiatan dalam batasan kriteria yang terpilih dan data yang tersedia (Apple, 1997). Dari uraian diatas, sangat jelas bahwa pemanfaatan desain lama yang didukung oleh komputer dapat mengurangi beban kerja desainer.

Dalam tugas akhir ini penulis menggunakan program aplikasi *Harlequin Lisp Work* (The Harlequin Group Limited, 1994) yang berbasis pada Windows. Tidak berbeda dengan bahasa program lainnya, *Harlequin Lisp Work* juga berisi tentang data awal permasalahan, struktur program untuk pemecahan masalah dan bagaimana menambahkan input serta menampilkan hasil akhir.

2.6. Representasi Kasus Desain (*Design Cases*)

Menurut Maher, dkk., (1995), perkembangan memori *case base* didasarkan pada hal-hal sebagai berikut :

- Isi dari kasus desain
- Representasi dari isi kasus desain
- Pengumpulan kasus desain dalam memori *case base*

a). *Isi dari kasus desain*

Kasus desain mempunyai beberapa bentuk yang berbeda, diantaranya yaitu sejarah desain, persyaratan dan penyelesaian desain, perbedaan pendapat



dalam permasalahan desain dan proses penyelesaiannya. (Maher dan Garza, 1997).

Sejarah desain dapat digunakan desainer untuk mempelajari situasi/kondisi yaitu kondisi baik dan buruk dari penyelesaian desain-desain lama yang telah ada, dan pengambilan keputusan dalam penyelesaian kasus desain lama tersebut. Selain itu, sejarah desain dapat digunakan untuk membantu desainer dalam mengambil keputusan ketika memecahkan masalah desain yang baru, atau masalah desain yang hampir sama.

Persyaratan desain lama yang berkaitan dengan kasus desain dapat digunakan untuk mendaftar ciri-ciri yang digunakan dalam menyelesaikan kasus yang ada. Hal tersebut dapat langsung dibandingkan dengan persyaratan desain baru untuk mendefinisikan gambaran umumnya. Sebagai tambahan, dapat diambil bentuk dari batasan desain yang dihubungkan dengan permasalahan desain. Penyelesaian desain lama (dalam bentuk penggambaran desain) dapat digunakan sebagai penyelesaian awal untuk masalah desain baru (*new design problem*). Untuk penyelesaian kembali kasus desain yang baru, penyelesaian desain lama disusun secara khusus dalam sebuah *file* yang disimpan dalam *case base*. (Maher, dkk., 1995).

Perbedaan pendapat dari masalah desain mengacu pada perbedaan representasi dari desain lama. Ada dua teori yang mendukung hal tersebut, yaitu : *accessibility theory* (teori kemampuan) yang merepresentasikan hubungan topologi antar ruang pada susunan tata letak berdasarkan kedekatannya, dan *night*



MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER



zone distribution theory yang merepresentasikan hubungan pertukaran antar ruangan dalam sebuah layout.

Dalam sistem Dave, dkk., permasalahan desain direpresentasikan dalam gambar geometri yang dilengkapi dengan informasi simbolik dan hubungan topologi antar ruang dalam tata letak yang didasarkan pada hubungan kedekatannya.

Proses penyelesaian yang digunakan dalam menyelesaikan permasalahan desain dapat digunakan kembali untuk menyelesaikan permasalahan desain yang baru.

b). Representasi dari isi kasus desain

Representasi dari isi kasus desain mengacu pada bentuk dimana isi dari kasus desain disimpan dalam *case base*. Kemungkinan akan didapat bentuk yang berbeda, bentuk yang paling umum adalah sepasang nilai atribut (*attribute-value pairs*). (Kolodner, 1993; Giretti, dkk., 1994; Vop, dkk., 1994; Maher, dkk., 1995; Flemrning, dkk., 1997). Bentuk yang lain meliputi text (Domeshek dan Kolodner, 1992; Kolodner, 1993; Gross, dkk., 1994), persamaan dan pertidaksamaan matematika (*mathematical-equalities and inequalities*) (Hua, dkk., 1996; Faltings, 1997), penggambaran berdasarkan grafik (Giretti, dkk., 1994; Vop, dkk.; Dave, dkk., 1994), *symbol/icon* (Giretti, dkk., 1994; Vop, dkk., 1994) dan representasi multimedia (Domeshek dan Kolodner, 1992; Gross, dkk., 1994).

Kasus desain yang direpresentasikan dengan sepasang nilai atribut dijelaskan dengan atribut-atribut dimana masing-masing atribut memiliki nilai sendiri. Sebagai contoh sebuah bangunan gedung dapat digambarkan dengan



informasi geometriknya (seperti panjang, lebar, luas dll) sebagai atributnya, dimana masing- masing mempunyai nilai sendiri.

Text dapat digunakan untuk menggambarkan sejarah desain (Kolodner, 1993) yang menjelaskan arsitektur kasus desain. Sistem ARCHIE dan Gross, dkk. menggunakan representasi ini.

Persamaan dan pertidaksamaan matematika digunakan untuk menggambarkan batasan desain yang berada dalam kotak desain. Sebagai contoh, dalam CADRE dan FAMING, persamaan dan pertidaksamaan matematika digunakan untuk menggambarkan satu set batasan menurut angka dalam dimensi rencana dasar kasus desain.

Grafik dapat digunakan untuk menggambarkan hubungan atribut-atribut atau bagian dari kasus desain, dimana *node* menggambarkan sebuah atribut atau bagian dari busur hubungan atau ketergantungan antara dua *nodes*. Sistem ASA, FABEL dan Dave, dkk. menggunakan grafik untuk menggambarkan *topology* (hubungan kedekatan antar ruang) dari sebuah layout.

Symbol/icon digunakan untuk memberikan gambaran khusus dari layout, seperti yang digunakan dalam ASA dan FABEL. Dalam ASA, simbol digunakan untuk menggambarkan teori *night zone distribution*.

Representasi multimedia merupakan kombinasi dari sejumlah kasus desain. Sistem Gross, dkk. menggunakan bentuk ini untuk mempermudah pemahaman terhadap permasalahan dengan memberikan beberapa model pemahaman.



c). Pengorganisasian kasus desain dalam memori case base

Kasus desain dapat diatur dalam memori *case base* sebagai kumpulan, dimana desainer dapat meninjau kembali dan menyeleksi setiap permasalahan secara langsung. Sistem CADRE, FAMING dan Dave, dkk. menggunakan pengaturan memori *case base* jenis ini. Fokus utama dari sistem ini adalah adaptasi permasalahan, menyeleksi sisa-sisa kasus desain untuk desainer. Pengumpulan kasus desain dalam memori *case base* dapat disusun sesuai rencana pengurutan (*indexing scheme*) (Maher, dkk., 1995). Dalam pengorganisasian jenis ini, masing-masing kasus desain diberi *indeks* sendiri-sendiri dan disimpan dalam *case base*.

Dalam sistem ARCHIE, masing-masing desain berisi sejarah desain. Tiap sejarah desain diurutkan dengan sekumpulan kombinasi ciri-ciri desain, seperti persoalan pokok desain (seperti keselamatan, kebebasan, kegaduhan dan kemampuan), komponen struktural (seperti lantai, ruangan dan bangunan samping) dan sistem fungsional (seperti sirkulasi, perpipaan dan listrik). Dalam hal ini, sejarah desain dari salah satu permasalahan desain saling terhubung satu dengan lainnya, dengan demikian hubungan antar permasalahan desain mengijinkan desainer untuk melihat dari satu permasalahan ke permasalahan lainnya.

2.7. Penarikan Kasus Desain

Penarikan kasus desain mengacu pada pencarian permasalahan-permasalahan dimana ciri-cirinya mempunyai kesamaan atau bersangkutan



dengan ciri-ciri dari permasalahan desain baru. Hal tersebut dihasilkan dengan mendapatkan kembali permasalahan menggunakan metode penarikan permasalahan.

Metode ini dapat dikelompokkan dalam dua kategori:

- Langsung ditentukan oleh desainer (*user*), yaitu penyelesaian permasalahan tanpa menggunakan proses *pattem matching*.
- Berdasarkan pada proses *pattern malching*.

a). ***Desainer langsung menyelesaikan permasalahan desain tanpa pattern matching***

Pada kategori pertama dari metode penyelesaian ini, desainer (*user*) mempunyai kontrol dalam memilih kasus desain dari *case base*, karena itu mereka dapat langsung mengambil/memilih penyelesaian kasus yang sesuai dengan *design problem* yang ada. Sistem ARCHIE, FAMING dan Dave, dkk. menggunakan metode penarikan desain yang sesuai dengan kategori ini. Selanjutnya, untuk membantu desainer (*user*) menyelesaikan kasus desain menjadi lebih efisien, ARCHIE menggunakan metode ini berdasarkan pada urutan *indeks*, dimana sejarah desain diurutkan dengan menggunakan gambaran/penjelasan desain yang ada.

b). ***Penyelesaian permasalahan tata letak berdasarkan proses pattern matching***

Kategori kedua dalam metode penyelesaian permasalahan meliputi proses pencarian untuk permasalahan, penyamaan permasalahan, penilaian atau



pendefinisian kesamaan antara permasalahan desain baru dan kasus desain, dan pemilihan desain yang sesuai.

Pattern matching dipertimbangkan sebagai proses kunci dalam kategori ini (Maher, dkk., 1995). Pola dari ciri-ciri permasalahan desain baru disamakan dengan pola ciri-ciri dari kasus desain dalam *case base*. Sistem Gross, dkk. adalah contoh-contoh sistem yang sesuai dengan kategori ini, dimana pola penyamaannya menggunakan simbol.

2.8. Adaptasi Kasus Desain

Adaptasi kasus desain dititikberatkan pada modifikasi pilihan desain yang didapatkan dari *case base* untuk mendapatkan penyelesaian desain baru yang memenuhi persyaratan desain. Ada beberapa metode adaptasi kasus yang berbeda yang digunakan dalam sistem penggambaran yang dapat dikategorikan menjadi metode informal dan formal.

Metode informal mengacu pada metode yang dalam permasalahan proses adaptasi ini dibuat langsung oleh desainer (*user*). ARCHIE dan Gross, dkk. menggunakan metode yang termasuk dalam kategori ini, tetapi fokus dari sistem-sistem tersebut adalah untuk menyajikannya sebagai suatu penjelajah kasus (*browser case*) yang menekankan penyampaian informasi tentang kasus desain kepada para desainer (*user*) yang lebih baik daripada adaptasi kasus desain yang diberikan kepada deainer (*user*) secara langsung.

Metode formal mengacu pada metode yang menggunakan algoritma-algoritma untuk menunjang adaptasi desain kasus. Metode ini terdiri dari batasan



pemenuhan persyaratan, teknik perubahan topologi, pergerakan dan pengujian, perencanaan ulang desain.

Metode batasan pemenuhan persyaratan diterapkan untuk adaptasi kasus yang digambarkan sebagai permasalahan batasan pemenuhan persyaratan (Maher, dkk., 1995; Hua, dkk., 1996). Karena itu adaptasi kasus desain bertujuan untuk memenuhi seperangkat batasan yang diinginkan dari permasalahan desain yang baru. Jenis-jenis adaptasi kasus dimana diterapkan metode batasan pemenuhan persyaratan meliputi adaptasi *dimensional* dan adaptasi desain obyek dari permasalahan desain layout. Adaptasi *dimensional* mengacu pada modifikasi dimensi kasus desain layout. Sistem Dave, dkk. menggunakan metode batasan pemenuhan persyaratan untuk adaptasi *dimensional* dari kasus desain layout. FABEL menggunakan metode batasan pemenuhan persyaratan untuk menyelesaikan permasalahan penempatan desain obyek dalam kasus desain layout yang diadaptasi.

Teknik-teknik perubahan topologi digunakan untuk adaptasi topologi, yaitu modifikasi dari hubungan topologi antar elemen kasus desain layout, seperti contoh : kedekatan antar ruang.

Metode mencoba dan menghasilkan digunakan dalam FABEL. Dalam metode ini diberikan susunan layout, penarikan kembali kasus desain layout dievaluasi untuk mengetahui apakah penempatan obyek dalam susunan layout bertentangan dengan elemen kasus desain atau tidak. Apabila terjadi perbedaan, penempatan dari obyek desain dimodifikasi sesuai dengan permasalahan desain. Modifikasi dari susunan layout ini kemudian digunakan untuk menyelesaikan



kembali kasus desain yang lain. Keseluruhan proses ini diulang sampai penyelesaian layout yang sesuai dengan persyaratan didapatkan. Operasi pengulangan tersebut memungkinkan desainer untuk menentukan penambahan, pemindahan, pengumpulan obyek desain dan atributnya.

2.9. Pendekatan *Pattern Matching*

Pattern matching merupakan inti dari *Case Based Reasoning* (CBR), dimana *pattern matching* ini digunakan untuk menentukan tingkat kesamaan hubungan antara input dari desainer dengan data-data desain yang ada dalam *case base*.

2.9.1. Klasifikasi *Pattern*

Pendekatan *pattern matching* difokuskan pada empat teknik yang berbeda, yaitu : *geometric pattern*, *topological relations*, *distributed pattern* dan *semantic/symbolic pattern* (Manfaat, dkk., 1998). Keempat teknik ini menunjukkan adanya perbedaan dasar dalam representasi *pattern* yang digunakan dalam teknik *pattern matching*.

a). *Geometric pattern*

Dalam teknik ini *pattern* didefinisikan dalam sistem koordinat, dimana representasi dari teknik ini adalah kumpulan titik, *dimensional graph* dan gambar.

Kumpulan titik didefinisikan sebagai kumpulan dari sejumlah koordinat dalam sistem koordinasi dua dimensi. *Dimensional graph* menunjukkan *graph*



yang memiliki elemen-elemen *vertices*, *edges* dan *faces* yang didefinisikan dalam sistem koordinat. *Graphs* tersebut dapat berbentuk dua dimensi (*planar graphs*) atau tiga dimensi (*polyhedra*). Sedangkan gambar dalam *geometric pattern* terdiri dari bermacam-macam bentuk, yaitu : *lines*, *arc*, *circles*, *polygons*, *curves* dan *characters*.

b). *Topological relations*

Teknik ini menggambarkan *pattern* sebagai bentuk hubungan antara elemen-elemen dari sebuah *pattern*. *Topological relations* ini direpresentasikan dalam bentuk *graph* yang tersusun atas *vertices*, *edges* dan *faces*. *Vertices* digambarkan sebagai elemen dari *pattern*, *edges* digambarkan sebagai hubungan antar elemen dan *faces* merupakan kumpulan kedekatan dari *edges*.

c). *Distributed pattern*

Teknik *distributed pattern* merupakan teknik yang mendistribusikan *pattern* ke dalam bentuk matriks (*grid units* atau *pixels*). Contoh representasi teknik ini adalah *spatial patterns*, *spatial relationship patterns*, dan *graphical patterns*.

Spatial patterns menghubungkan pola-pola yang memiliki bentuk *shapes* dari *faces*. Masing-masing *grid unit* atau *pixels* dari *pattern* merepresentasikan bagian *shape* dari *space*.

Spatial relationship patterns menghubungkan pola-pola yang merepresentasikan hubungan antara *spaces* pada *spatial layout*, dimana beberapa *grid unit* atau *pixel* merepresentasikan hubungan kedekatan antara dua ruang dan merepresentasikan posisi dari ruang tertentu yang memiliki kedekatan.



Graphical patterns ini pada dasarnya merepresentasikan bagian dari pola, tetapi tidak dalam bentuk *spatial pattern*.

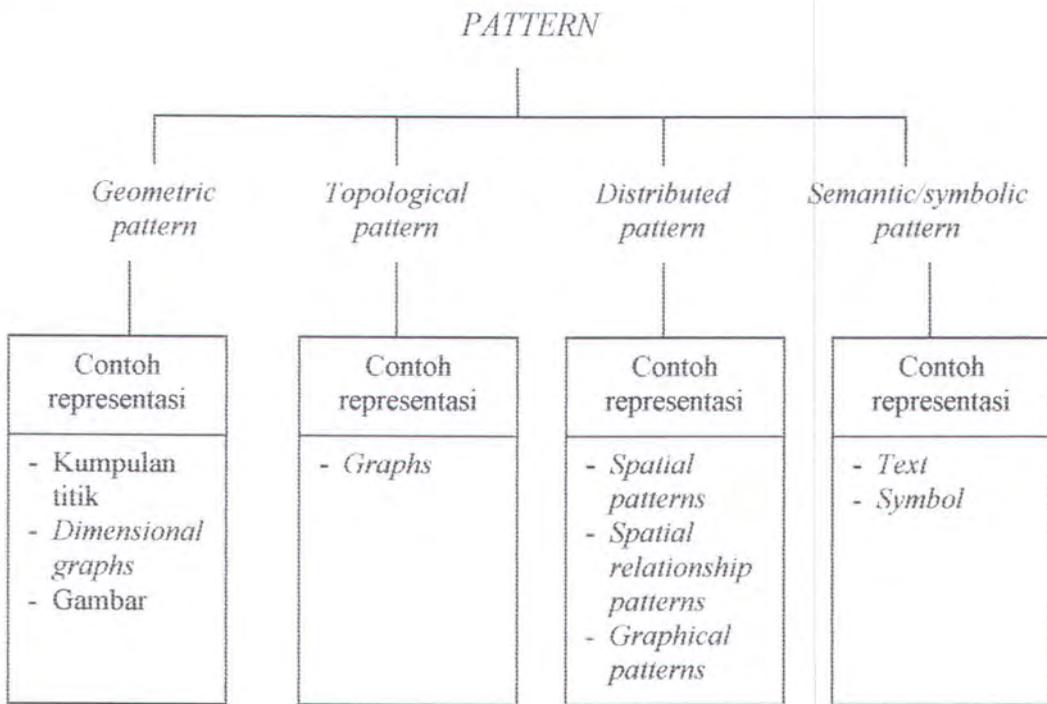
d). *Semantic/symbolic pattern*

Pada *semantic/symbolic pattern* dilakukan representasi atribut-atribut *semantic/symbolic* dari obyek yang meliputi nama, fungsi, ukuran, warna dan atribut lainnya. Atribut-atribut tersebut dapat direpresentasikan dalam bentuk teks atau simbol (seperti dalam diagram, *icon* dan lainnya). Pasangan nilai-nilai atribut dapat direpresentasikan dalam bentuk daftar (*list*).

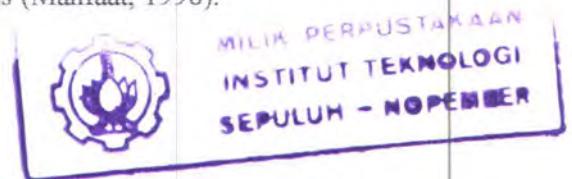
Sistem dari Dave, dkk. (1994), Giretti, dkk. (1994), dan Gross, dkk. (1994) yang berdasarkan pada CBR menggunakan *symbolic pattern* untuk melakukan *pattern matching* sebagai bagian dari proses penarikan desain (*recall*).

Sistem Giretti, dkk. menggunakan *symbolic pattern* untuk menunjang representasi *planar graph* yang merepresentasikan *spatial relationship* dari *spatial layout*. Sedangkan pada sistem Gross, dkk., *symbolic pattern* digunakan dalam bentuk diagram dan dihubungkan dengan program komputer untuk menginterpretasikan sketsa gambar tangan dan diagram. Sketsa ini dibentuk dari kumpulan primitif elemen yang disebut *glyphs*. Contoh dari *glyphs* adalah *lines*, *circles*, *boxes*, *triangles*, *arrows*, *numbers*, *letters* dan lain-lain.

Dalam Tugas Akhir ini penulis menggunakan teknik *topological relations* dan *numerical pattern matching* untuk melakukan proses penarikan (*recall*) desain lama. Teknik ini digunakan dengan alasan mudah dipahami oleh pengguna dan pemakaian angka-angka ukuran akan mempercepat proses *matching*.



Gambar 2.2. Tipe dasar dari *patterns* (Manfaat, 1998).



2.9.2. Metode *Matching*

Metode *matching* merupakan metode yang digunakan dalam proses *pattern matching* yang bertujuan untuk menentukan tingkat kesamaan antara pola dari input desainer (*user*) dengan desain-desain yang ada didalam *case base*. Ada tiga metode yang dipakai dalam *pattern matching*, yaitu: *neural networks*, *mathematical technique* dan *symbolic pattern matching*.

a). *Neural networks*

Metode ini merupakan model matematis dari struktur otak manusia (Coyne dan Newton, 1989), yang disebut juga *parallel distributed processing system* (Rumelhart, 1989). Metode ini terdiri dari sejumlah jaringan unit proses (*nodes*) yang berisi data dan dihubungkan dengan garis penghubung (*weighted*



arcs). Dalam metode ini input akan dimatching dengan sejumlah data yang sudah ada dan dengan proses iterasi akan diperoleh kesamaan antara input dengan data.

Metode *neural networks* ini mampu mengenali pola-pola yang tidak lengkap atau tidak teratur dan menghasilkan pola baru dengan pola yang lengkap. Kemampuan ini sangat mendukung dalam pengenalan desain, kontrol proses dan lainnya.

b). *Mathematical technique*

Metode *mathematical technique* ini banyak digunakan dalam permasalahan yang mengandung *geometric* dan *graph matching*. Pada intinya metode ini menggunakan teori matematika untuk menganalisa kesamaan antara pola-pola yang berbeda. Beberapa metode yang digunakan dalam matematika teknik yaitu : *coordinate distance measurement*, *hashing method* (Akutsu, 1995), *dynamic method* (Akutsu, 1995), *cross correlation* (Cooper, 1989) dan *graph theoretic algorithms* (Hopcroft dan Trajan, 1973).

Coordinate distance measurement

Metode ini digunakan untuk mengukur jarak antara titik, garis, kurva dan lainnya dalam sistem koordinat. Schneider, dkk. (1989), menggunakan metode ini untuk mengukur kesamaan bentuk antara dua gambar .

Hashing method

Metode ini memberikan akses langsung dan cepat pada data tertentu dengan cara memberi nilai tertentu dalam *data base*. Digunakan dalam proses *matching* bentuk tiga dimensi yang merupakan kombinasi antara *dynamic*



programming technique dengan *Kabsch's method* (1976). Tujuannya adalah meningkatkan efisiensi proses *pattern matching*.

Cross correlation

Berdasarkan pada *method template matching* yang mengukur kesamaan antara *template* dengan data/gambar .

Graph theoretic algorithms

Metode ini berdasarkan pada algoritma untuk mencari kesamaan antara *graph*. Ada beberapa tipe kesamaan antar *graph* (Harary, 1969 dan Berger, 1976) yaitu :

- 1). *Graph isomorphism* : mencari kesamaan antara *graph* dengan *graph* lainnya.
- 2). *Subgraph isomorphism* : mencari kesamaan antara *graph* dengan *subgraph* dari *graph* lainnya.
- 3). *Double subgraph isomorphism* : mencari kesamaan antara *subgraph* sebuah *graph* dengan *subgraph* dari *graph* lainnya.

Ballard dan Brown (1982), memberikan contoh dari ketiga tipe diatas yaitu : *matching metric*, *backtrack search* dan *associated graph matching* (pembentukan struktur data dari dua *graph* yang *dimatching*).

c). *Symbolic pattern matching*

Dipakai untuk proses *matching* dimana obyek mempunyai pola berupa simbol-simbol. Dalam prosesnya, pola-pola itu akan direpresentasikan sebagai simbol.

Dalam penulisan Tugas Akhir ini digunakan metode *mathematical technique* khususnya *coordinat distance measurement*.



2.9.3. Jenis Kesamaan

Penentuan kesamaan dilihat dari tinjauan terhadap suatu obyek (Rips, dkk., 1989). Tinjauan dapat berupa kategori obyek dan cara penyelesaiannya. Kategori dapat meliputi atribut-atribut tertentu yang dimiliki oleh obyek (*properties*), atribut tersebut dapat berupa fungsi, bentuk, ukuran atau hubungan lainnya.

Smith, dkk., (1989) mendefinisikan kesamaan kedalam dua tipe yang berbeda yaitu : *concrete-relation* dan *abstract-relation*. *Concrete-relation* adalah hubungan kesamaan antar atribut yang dimiliki oleh obyek. Sedangkan *abstract-relation* adalah hubungan yang mengatur kesamaan antara obyek. Pembagian klasifikasi *concrete-relation* ke dalam lima bagian : *resemblances*, *overall similarity*, *identity*, *part-similarity*, dan *part-identity*.

Resemblances

Klass ini merupakan hubungan yang meliputi seluruh hubungan kesamaan (Smith, 1989). *Resemblances* bersifat umum, contohnya adalah layout ruang akomodasi kapal dengan layout galangan kapal.

Overall Similarity

Dalam kesamaan ini semua hubungan yang terjadi dalam obyek dinyatakan dalam satu karakteristik. Contohnya adalah layout ruang akomodasi kapal tanker adalah *overall similarity* dengan layout ruang akomodasi pada kapal penumpang. Alasannya adalah keduanya merupakan layout ruang akomodasi di kapal.



Identity

Pada kesamaan ini semua hubungan yang terjadi dinyatakan sebagai satu karakteristik dan harus identik dengan input (*template*). Perbedaan dengan *overall similarity* adalah pada tingkat kesamaannya. Contohnya adalah kriteria dalam *chemical pattern matching*, dimana setiap *sub-graph* dari dua *graph* yang diuji harus identik.

Part-similarity

Pattern matching dalam kesamaan ini menggunakan sebagian dari atribut yang dimiliki oleh obyek. Contohnya adalah proses pencarian kesamaan dalam tugas akhir ini, dalam *sub-graph* setiap hubungan korespondensi *template* akan dipetakan ke dalam *data graph*. Jika salah satu hubungan memenuhi *part-similarity* maka akan diperoleh sebuah angka kesamaan.

Part-identity

Klas ini hampir sama dengan *part-similarity*, namun setiap atribut yang dimatching harus identik antara *template* dan data. Contoh *part-identity* dengan tinjauan dari segi fungsi adalah dua kapal tanker adalah identik.

2.10. Sistem SPIDA

Sebuah sistem komputer yang bernama SPIDA (*SPatial Inteligent Design Assistant*) yang mempunyai kemampuan melakukan *topological pattern matching* dan kombinasi *topological pattern* dan *geometric shape matching* untuk *spatial layout* secara umum telah dikembangkan oleh Manfaat (1998).



Topological pattern matching antar layout bertujuan untuk menentukan kesamaan *topological pattern* diantara layout-layout tersebut. Dalam proses *pattern matching* jenis ini, *association graph technique* (Ballard dan Brown, 1982) digunakan. Dengan ini SPIDA mampu menemukan jumlah maksimum ruangan yang fungsi-fungsinya bersesuaian diantara dua layout yang dibandingkan (*di-pattern matching*). Kemudian SPIDA mampu mengurutkan tingkat kesamaan antar layout yang dapat dipilih oleh desainer untuk digunakan sebagai desain awal dalam proses desain baru.

Seperti halnya *topological pattern matching* dalam kombinasi *topological pattern* dan *geometric shape matching*, SPIDA mampu menentukan tidak hanya kesamaan topologinya tetapi sekaligus juga kesamaan bentuk antar ruangan-ruangan yang fungsinya bersesuaian yang dihasilkan dari proses *topological pattern matching*. Selain *association graph technique* yang digunakan untuk mengukur kesamaan *topological pattern technique*, *planar shape matching* yang dikembangkan oleh Leu dan Huang (1988) digunakan untuk mengukur bentuk antar ruangan. Teknik yang terakhir ini dapat memberikan hasil yang akurat dimana pengukuran kesamaannya ditentukan berdasarkan *dissimilarity scores*, artinya tingkat perbedaan bentuk antar dua ruangan adalah yang diukur. Dengan demikian semakin besar *score* ini maka semakin berbeda kedua ruang tersebut. Atau dengan kata lain semakin kecil *score* tersebut maka tingkat kesamaan bentuk ruang tersebut semakin besar.



2.11. Pertimbangan Aspek Struktural dalam *Layout Design*

Dalam merencanakan suatu layout ruang akomodasi pada bangunan atas (*superstructure*), seorang desainer harus mempertimbangkan juga aspek struktural (konstruksi) disamping pertimbangan aspek-aspek yang lain menurut Smith dan Hatfield (1977) yang telah dijelaskan pada sub bab sebelumnya.

Aspek struktural dalam desain layout meliputi:

- 1). Penempatan dinding/sekat untuk ruangan pada bangunan atas.
- 2). Perencanaan bukaan-bukaan, seperti *engine casing*.
- 3). Sistem konstruksi yang digunakan pada ruang akomodasi.

Untuk lebih jelasnya berikut ini adalah sedikit uraian mengenai ketiga aspek struktural tersebut:

2.11.1. Penempatan Dinding/Sekat untuk Ruang pada Bangunan Atas

Bangunan atas yang meliputi ruang akomodasi diasumsikan sebagai konstruksi yang kaku dan tegar (*stiff*), menerus dari deck paling bawah (*lowest deck*) yaitu upper deck sampai dengan deck paling atas yaitu top deck. Konstruksinya dibuat menerus dengan tujuan untuk menghindari adanya konsentrasi tegangan. Karena konstruksinya yang menerus itulah maka bangunan atas harus dibuat setegar/sekokoh mungkin, baik pada sekat-sekatnya maupun pada sambungan-sambungannya.

Untuk memperoleh konstruksi yang kokoh maka pada sekat-sekat bangunan atas harus dipasang penegar-penegar (*stiffeners*) dan harus ditumpu





oleh gading-gading (*frames*). Selain itu peletakan dinding/sekat ruang akomodasi harus tepat pada gading untuk sekat melintangnya dan tepat pada pembujur untuk sekat memanjangnya.

2.11.2. Perencanaan Bukaannya

Seperti halnya dinding/sekat, perencanaan bukaan pada ruang akomodasi juga harus memperhatikan posisi gading maupun pembujur (*longitudinals*). Dimana posisi dinding/sekatnya harus sedemikian rupa, yakni tertumpu pada gading maupun pembujur.

Misalnya *engine casing*, sedapat mungkin konstruksi di bawah bukaannya diperkuat dengan memasang balok penegar, baik ke arah melintang maupun ke arah memanjang. Yang perlu diperhatikan adalah konstruksi manakah yang harus dikalahkan. Maksudnya ialah apakah balok pelintang yang dibuat menerus, sedangkan balok memanjangnya terpotong, ataukah sebaliknya. Hal ini harus diperhitungkan menurut peraturan-peraturan (*Rules*) yang digunakan oleh masing-masing desainer.

2.11.3. Sistem Konstruksi yang digunakan pada Ruang Akomodasi

Sistem Konstruksi yang digunakan pada ruang akomodasi mempunyai peranan yang sangat penting bagi terciptanya konstruksi yang kokoh. Tidak berbeda dengan sistem konstruksi yang diterapkan pada lambung kapal, sistem konstruksi pada bangunan atas, tepatnya pada ruang akomodasi bertujuan agar



pelat maupun penegarinya tidak mengalami *buckling* bila memperoleh gaya dari luar maupun dari dalam.

Untuk itu pemasangan penegar pada plat, ukuran penegar, gaya-gaya yang bekerja serta ketebalan plat harus terlebih dahulu diperhitungkan sesuai dengan aturan yang digunakan untuk menghindari *buckling*. Hal ini berlaku bagi lambung maupun ruang akomodasi pada bangunan atas tanpa terkecuali, termasuk juga layout ruang akomodasi yang dijadikan data di dalam *case base*.

2.11.4. Analisis Layout Ruang Akomodasi Kapal Berdasarkan Aspek Struktural

Dalam kasus ini, semua layout ruang akomodasi yang dimasukkan ke dalam *case base* adalah layout yang dianggap telah memenuhi semua syarat konstruksi di atas. Dalam hal ini penempatan dinding/sekat ruangan, perencanaan *engine casing* dan kaitannya dengan sistem konstruksi yang digunakan pada masing-masing kapal adalah maksimal. Hal itu berarti bahwa layout tersebut telah memperhitungkan aspek struktural dalam desainnya.

Sebagai contoh adalah berikut ini akan diberikan hasil analisis terhadap ruang akomodasi pada beberapa kapal di dalam *case base*, tepatnya pada Poop Deck:

- a. Ruang battery (battery room) dan Store kapal 1, terletak pada gading no. 15 ~ gading no. 18.
- b. Ruang ESB dan CO₂ room kapal 1, terletak pada gading no. 18 ~ gading no. 22.



- c. Engine casing kapal 1, terletak pada gading no. 7 ~ gading no. 22.
- d. Gangway kapal 1, terletak pada gading no. 22 ~ gading no. 24.
- e. Radio officer, Second officer, Second engineering dan steward kapal 1, terletak pada gading no. 24 ~ gading no. 29.



BAB III
IMPLEMENTASI TEKNIK

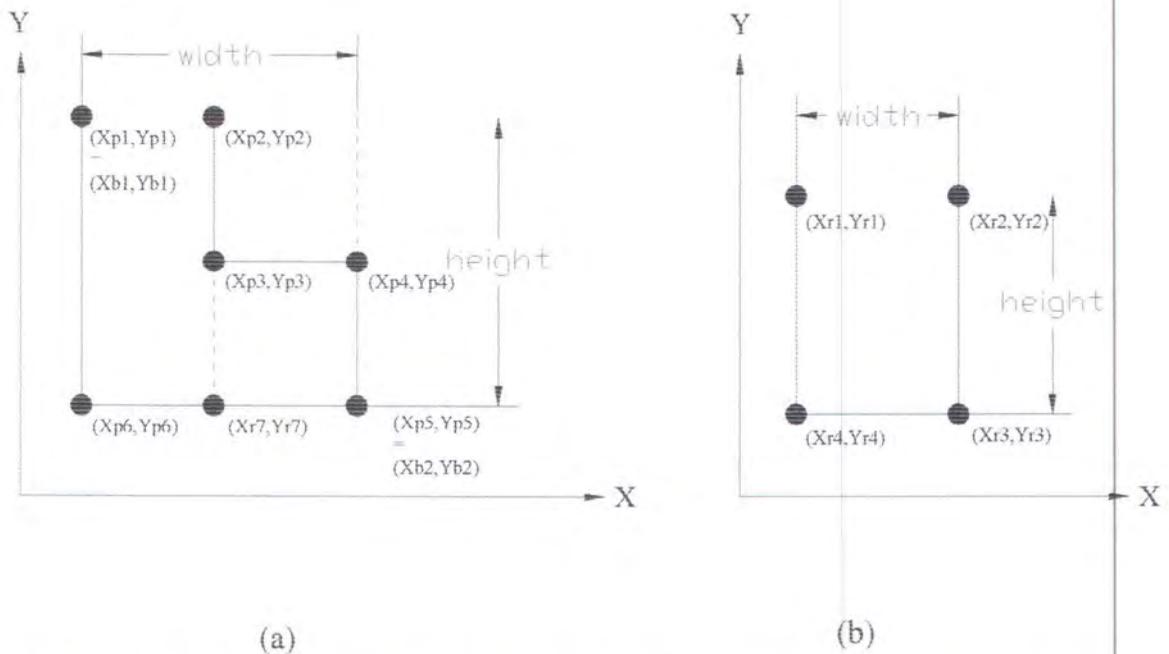


BAB III

IMPLEMENTASI TEKNIK

3.1. Representasi Ruang

Tugas akhir ini mendefinisikan ruang sebagai bangun dua dimensi yang berbentuk persegi panjang (*rectangle*) (termasuk bujur sangkar) atau poligon (*polygon*). Ruang tersebut direpresentasikan dalam suatu sistem koordinat



Gambar 3.1. Representasi ruang bentuk *polygon* (a), dan *rectangle* (b) (Manfaat, 1998).

Untuk merepresentasikannya, ruang tersebut harus mempunyai beberapa atribut, yaitu:

- *name*

Nama dari ruang sebagai identitas ruang tersebut, contoh *Galley_PD01s*.

Menunjukkan bahwa ruangan tersebut adalah ruang untuk dapur (*galley*), dimana



letaknya adalah pada *Poop Deck* untuk kapal (*ship*) ke-1 yang ditunjukkan oleh huruf *s*, untuk membedakan antara data pada *case base* dengan ruangan sebagai *template*. Sedangkan pada *template* digunakan huruf *t* pada akhir penamaan.

Data *template* adalah suatu data contoh yang direncanakan oleh desainer maupun permintaan *owner*.

- ***text***

Memberi keterangan label dari setiap ruangan dan menunjukkan identitas ruang tersebut pada saat ruangan tersebut ditampilkan dalam bentuk gambar layout. Contoh ***Galley***.

- ***space-function***

Menunjukkan fungsi dari ruangan, apakah ruangan tersebut digunakan untuk ruang pribadi, ruang istirahat, kantor, dapur dan sebagainya.

Contoh pada ruang yang mempunyai nama ***Galley_PD01s***, mempunyai fungsi *Galley* yaitu ruangan untuk memasak/dapur.

- ***Space-identifier***

Data ruangan tersebut terbagi menjadi dua kategori, yaitu satu sebagai data pada *case based*, sedangkan bagian yang lain sebagai data *template*. Untuk mengelompokkannya digunakan *identifier*. Ada dua macam *identifier* yaitu *Stored_Object* untuk menunjukkan data *case based* dan *Template_Object* untuk menunjukkan data *template*.

- ***geometric-shape***

Bentuk dari ruangan ditunjukkan pada atribut ini. Ada dua macam bentuk yaitu *rectangle* (persegi panjang) dan *base polygon* (poligon dasar). Untuk



rectangle disusun oleh koordinat pojok kiri atas dan lebar serta tinggi ruangan yaitu *outline-rect* ($Xr1 Yr1 w h$). Sedangkan untuk poligon disusun oleh koordinat pembentuk segiempat yaitu *vertices-coordinates* ($Xp1 Yp1 Xp2 Yp2 Xp3 Yp3 Xp4 Yp4 Xp5 Yp5 Xp6 Yp6$), *component-rect* ($(Xp1 Xr7 Yp1 Yr7) (Xp3 Xp5 Yp3 Yp5)$).

- ***bounding-box***

Segi empat pembatas adalah batas maksimum dari sebuah segiempat ataupun poligon. Untuk *rectangle* ($Xr1 Yr1 Xr3 Yr3$), sedangkan untuk poligon ($Xb1 Yb1 Xb2 Yb2$).

- ***area***

Luas dari ruangan tersebut ditunjukkan oleh atribut *area_ruang*.

- ***area-ratio***

Perbandingan luas area ruangan dengan luas bounding box yang mengelilinginya.

- ***width***

Lebar ruangan.

- ***height***

Tinggi ruangan.

- ***w-h-ratio***

Perbandingan antara lebar ruangan dengan tingginya.

- ***colour***

Menunjukkan atribut warna dari ruangan tersebut pada saat direpresentasikan dalam gambar.



3.2. Representasi Layout

Seperti pada ruang, layout mempunyai beberapa atribut yang menggambarkan *properties* dari layout tersebut.

Atribut tersebut yaitu:

- ***name***

Menunjukkan nama dari layout, dimana nama tersebut menunjukkan jenis, urutan dan letak layout tersebut.

Contoh *Ship_01_Deck_UD*, *Ship* menunjukkan jenis data, apakah termasuk data *case base* atau data *template*. Angka *01* setelah *Ship* menunjukkan urutan data dan Huruf *UD* setelah *Deck* menunjukkan letak layout tersebut di geladak *Upper-Deck*.

- ***comp-sp***

Layout terdiri dari beberapa ruang, *comp-sp* menunjukkan ruangan-ruangan yang termasuk dalam suatu layout.

- ***assignment***

Seperti yang ditunjukkan *name*, layout-layout tersebut dibedakan atas jenis data dan letaknya. Atas jenis data dibagi menjadi dua yaitu *case base* dan *template*, sedangkan berdasar letaknya dibagi menjadi 5 deck, sehingga ada 10 macam jenis layout. Jadi atribut ini menunjukkan termasuk jenis yang mana layout tersebut.

- ***sp-closeness***

Menunjukkan topologi (kedekatan antar ruang) dalam layout tersebut.



3.3. Proses *Pattern Matching*

Ketika proses pencarian contoh desain dilakukan, sistem melakukan *pattern matching* antara contoh – contoh desain dengan input yang ada. Proses ini melibatkan perbandingan nilai dari atribut yang sesuai antara contoh desain dengan *input*. *Pattern matching* jenis ini dapat dimasukkan kedalam teknik *topological and numerical pattern matching* dengan tujuan untuk mencari skor kesamaan tertinggi antara contoh desain dan *input* dalam aspek topologi dan numerik.

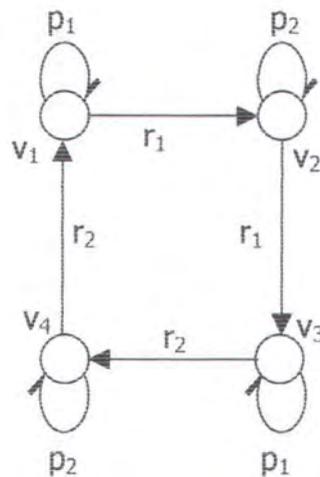
3.4. *Topological Pattern Matching*

Topologi dari suatu layout direpresentasikan dalam suatu *graph*. *Graph* tersebut terdiri atas beberapa kumpulan *node* V , kumpulan *property* P dari *node* tersebut dan kumpulan *relation* R yang mendefinisikan pasangan dari *node*. Suatu contoh dari suatu representasi *graph* diberikan dalam Gambar 3.2. *Node* v_1 dan v_3 mempunyai *property* p_1 , v_2 dan v_1 mempunyai *property* p_2 , dan egde antara v_1 dan v_2 dan antara v_2 dan v_3 masing-masing mempunyai *relation* r_1 , dan antara v_3 dan v_4 dan antara v_4 dan v_1 masing-masing mempunyai *relation* r_2 .

Diberikan dua *graph* didefinisikan dengan (V_1, P, R) dan (V_2, R, P) , untuk mencocokkan dua *graph* tersebut sebuah *association graph* digambar. Suatu *association graph* adalah suatu *graph* yang *node*-nya adalah pasangan *node* dari dua *graph*, yang mempunyai kemiripan *properties*, dan *egde* menghubungkan *node* dari *graph* tersebut yang menunjukkan keadaan yang *compatible*. Untuk menggambar G , gambar sebuah *node* dari G dengan label (v_1, v_2) untuk setiap v_1



dalam V_1 dan v_2 dalam V_2 dengan syarat bahwa v_1 dan v_2 harus mempunyai kesamaan *property* p dalam P . Selanjutnya, hubungkan dua *node* (v_1, v_2) dan (v'_1, v'_2) dari G yang menunjukkan keadaan yang *compatible* dengan R (contoh $r(v_1, v'_1) = r(v_2, v'_2)$).



Gambar 3.2. Representasi sebuah *graph* (Ballard dan Brown, 1982).

Sebagai contoh menggambar suatu “*association graph*”, yang mempertimbangkan *dua layout* yang pola topologinya dicocokkan seperti ditunjukkan dalam Gambar 3.3. dan Gambar 3.4. berikut:

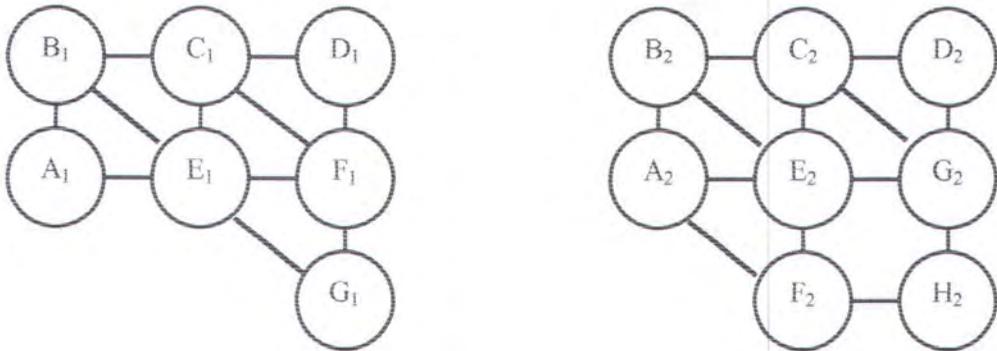
Layout input (I)

B ₁	C ₁	D ₁
A ₁	E ₁	F ₁
		G ₁

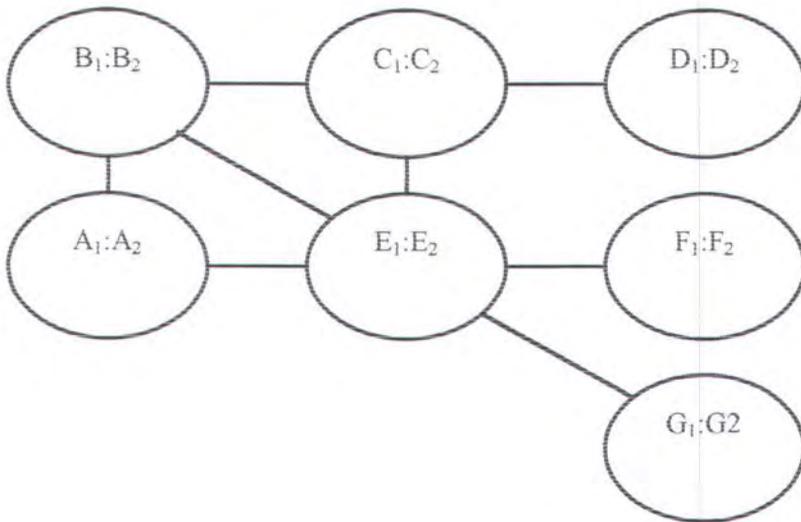
Layout di dalam case base (II)

B ₂	C ₂	D ₂
A ₂	E ₂	G ₂
	F ₂	H ₂

Gambar 3.3. Dua buah *layout* yang akan dicocokkan topologinya.



Gambar 3.4. Representasi layout dalam bentuk *graph*.



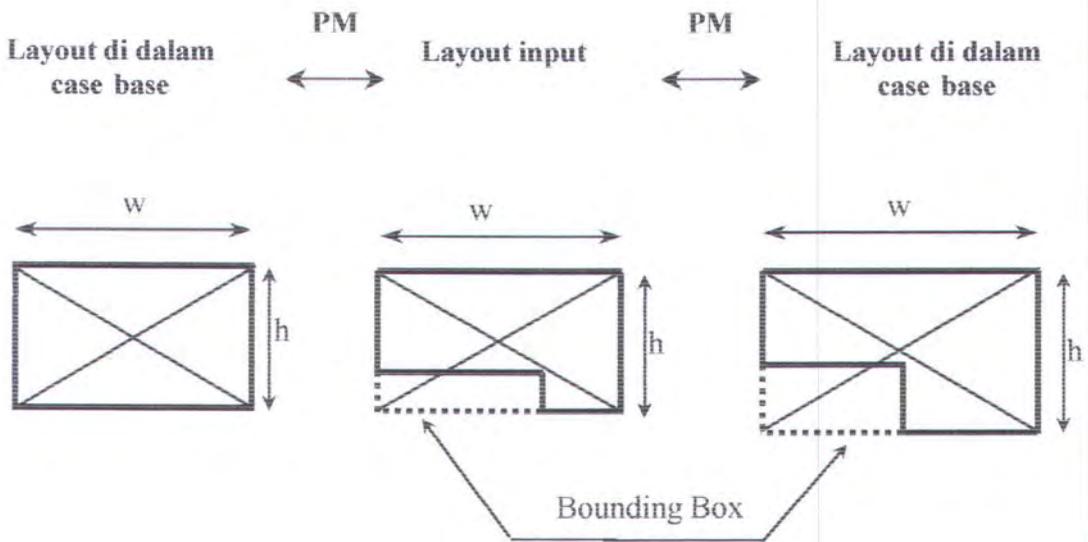
Gambar 3.5. Suatu *association graph* yang dihasilkan dari *matching* dua layout dalam Gambar 3.3. dan 3.4.

3.5. Numerical Pattern Matching

Setelah diperoleh topologi hasil *pattern matcing* dan diperoleh *score* untuk tahap tersebut, kemudian dilakukan *Numerical pattern matching* untuk memperoleh *score* dari kesamaan data-data perhitungan yang berhubungan dengan nilai-nilai dan angka-angka numerik. Dimana ukuran numerik yang meliputi width (w), height (h), luas (Area), rasio w/h maupun rasio luas



poligon/luas bounding box dari desain input (I) dibandingkan dengan data numerik dari desain di dalam case base. Setelah itu dicari score-nya untuk masing-masing perbandingan. Hal ini ditunjukkan dalam Gambar 3.6. Hasil akhirnya adalah berupa total score dari tahap 1 dan tahap 2.



Atribut numerik meliputi :

- ✓ Width (L)
- ✓ Lebar (B)
- ✓ Luas (A)
- ✓ Rasio L/B
- ✓ Rasio Luas Polygon/Luas Bounding Box

Gambar 3.6. Model kesamaan numerik antara layout input dengan layout di dalam case base.



3.5.1. Pengukuran Kesamaan Antara Contoh Desain dan Masukan (*Input*)

Teknik untuk mendapatkan ukuran pertidaksamaan antara obyek atau pola yang dikemukakan oleh Jain dan Dubes (1988) dalam bidang *pattern recognition* diadopsi untuk mendefinisikan tingkat perbedaan atau kedekatan antar obyek. Teori yang mendasari teknik ini adalah *pattern matrix*, *proximity matrix*, dan *proximity index* dimana teknik tersebut banyak digunakan untuk menentukan ukuran kesamaan.

- *Pattern matrix*

Pattern matrix adalah sebuah format yang merepresentasikan hubungan antar pola yang merepresentasikan obyek dan ciri – cirinya (atau atribut atau skor). Bila suatu obyek direpresentasikan oleh sebuah kumpulan ciri – ciri d , maka dapat dikatakan bahwa obyek tersebut direpresentasikan oleh sebuah pola, atau vektor d -kolom (Jain dan Dubes, 1988). Jika ada obyek sebanyak n , kumpulan obyek ini dipandang sebagai sebuah *pattern matrix* $n \times d$, tiap baris dari matrik ini merepresentasikan sebuah pola, sedangkan tiap kolom menunjukkan sebuah ciri atau ukuran. Angka yang menghubungkan baris dengan kolom menunjukkan nilai dari ciri sebuah pola. Contoh dari *pattern matrix* dapat dilihat dalam Gambar 3.7. berikut:



pola	Width (ciri 1)	Height (ciri 2)	Area (ciri 3)	---	W/H (ciri d)
Space1_UD	15	18	270	---	0.833
Space2_UD	14	12	168	---	1.167
Space3_UD	12.5	9	112.5	---	1.389
---	---	---	---	---	---
Spacen_UD	11	13.4	147.4	---	0.821

Gambar 3.7. Contoh dari *pattern matrix*.

- **Proximity matrix**

Indek kedekatan antar pola dapat dibuat. Indek ini dapat dihitung dari *pattern matrix* atau dapat dibentuk dari data mentah (Jain dan Dubes, 1988). Suatu *proximity matrix* $[d(i, j)]$ disusun dari indek pasangan kedekatan antar pola. Dalam matrik ini, tiap baris dan kolom merepresentasikan sebuah pola. Sebagai contoh, dengan mempertimbangkan spesifikasi desain Space1_UD sampai n sebagai pola-pola yang berbeda yang akan dibandingkan (Gambar 3.8.), maka *proximity matrix* akan menjadi:



	Space1_UD	Space2_UD	Space3_UD	---	Spacen_UD
Space1_UD	0	10	12	---	15
Space2_UD	---	0	13	---	11
Space3_UD	---	---	0	---	17
---	---	---	---	---	---
Spacen_UD	---	---	---	---	0

Gambar 3.8. Contoh dari *proximity matrix*.

Proximity index dapat berupa kesamaan (*similarity*) atau pertidaksamaan (*dissimilarity*) (Jain dan Dubes, 1988). Semakin obyek/pola ke-*i* dan ke-*j* menyerupai satu dengan lainnya, maka indek kesamaan akan semakin besar dan indek pertidaksamaan akan semakin kecil.

- **Proximity index**

Menurut Jain dan Dubes (1988), *proximity index* antara pola ke-*i* dan ke-*k* dinotasikan dalam $d(i,k)$ dan harus memenuhi tiga sifat yaitu :

- 1) (a) untuk pertidaksamaan : $d(i,i) = 0$, untuk semua *i*.
(b) untuk kesamaan : $d(i,i) \geq \max d(i,k)$, untuk semua *i*.
- 2) $d(i,k) = d(k,i)$, untuk semua (*i,k*).
- 3) $d(i,k) \geq 0$, untuk semua (*i,k*).

dalam penelitian yang telah dilakukan, teknik untuk menentukan kesamaan antara spesifikasi desain lama dan permintaan pemesan dalam bentuk *attribute – value* telah ditentukan. Tipe dari nilai dari atribut-atribut ini dapat dikategorikan sebagai



nilai nominal (seperti: nama dan letak ruangan) atau kontinyu (seperti: width, height, area dan w/h). Karena ciri pada penelitian ini bertipe kontinyu, maka proximity index yang paling banyak digunakan, yaitu metrik **Minkowski**, diadopsi untuk mengukur pertidaksamaan (Jain dan Dubes, 1988).

Dalam sebuah *pattern matrix*, pola ke- i dari matrik, ditunjukkan dengan vektor kolom x_i .

$$x_i = (x_{i1}, x_{i2}, \dots, x_{id})^T, i = 1, 2, \dots, n$$

dimana d adalah jumlah ciri, n jumlah pola dan T menunjukkan vektor transpos.

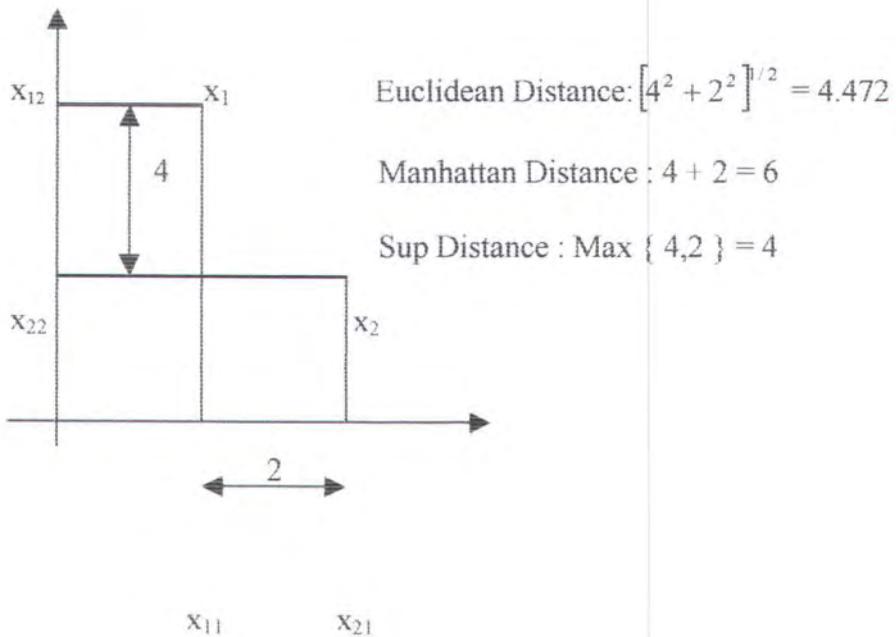
Metrik Minkowski didefinisikan sebagai berikut :

$$d(i, k) = \left(\sum_{j=1}^d |x_{ij} - x_{kj}|^r \right)^{1/r}, \text{ dimana } r \geq 1$$

Metrik ini juga harus memenuhi sifat – sifat metrik tambahan (Jain dan Dubes, 1988):

- 4) $d(i, k) = 0$, hanya jika $x_i = x_k$
- 5) $d(i, k) \leq d(i, m) + d(m, k)$, untuk semua (i, k, m)

Tiga metrik Minkowski yang paling umum ditentukan diilustrasikan dalam Gambar 3.9. Dalam gambar ini, dua pola x_1 dan x_2 , masing–masing dengan dua ciri: x_{11}, x_{12}, x_{21} dan x_{22} .



Gambar 3.9. Metrik-metrik Minkowski (Jain dan Dubes, 1988).

Berdasarkan ilustrasi dalam gambar 3.9., bentuk umum dari tiga metrik Minkowski dapat ditentukan sebagai berikut (Jain dan Dubes, 1988):

1. $r = 2$ (Euclidean distance)

$$d(i, k) = \left[\sum_{j=1}^d (x_{ij} - x_{kj})^2 \right]^{1/2} = \left[(x_i - x_k)^T (x_i - x_k) \right]^{1/2}$$

2. $r = 1$ (Manhattan, atau taxicab, atau city block distance)

$$d(i, k) = \sum_{j=1}^d |x_{ij} - x_{kj}|$$

3. $r \rightarrow \infty$ (“ sup “ distance)

$$d(i, k) = \max |x_{ij} - x_{kj}|$$

$$1 \leq j \leq d$$



dalam hal ini *Euclidean distance* digunakan, karena ini merupakan metrik Minkowski yang paling umum digunakan terutama dalam bidang teknik. Disamping itu, *Euclidean distance* merupakan metrik satu-satunya yang dalam teori geometri tidak terpengaruh oleh adanya translasi dan rotasi dari ruang pola.

3.5.2. Normalisasi

Ketika menggunakan metrik *Euclidean distance* untuk mengukur kedekatan antara nilai-nilai dari atribut suatu ruangan, maka pertimbangan harus secara hati-hati diarahkan pada skala perbedaan antar ciri. *Euclidean distance* secara implisit lebih mementingkan ciri-ciri dengan subjangkauan (*range*) yang besar dari pada *range* yang lebih kecil. Artinya, menskalakan satu ciri dalam satuan m^2 dan ciri kedua dalam m membuat ciri kedua secara numerik memberikan pengaruh (*impact*) yang lebih besar dari pada ciri pertama.

Contoh yang lain adalah perbedaan skala antara nilai-nilai dari ciri-ciri ruang. Nilai dari *width* semua ruang pada upper deck misalnya adalah *width* antara 10 – 30 m, sementara nilai dari *area* ruang besar dapat sampai ratusan m^2 . Jadi perbedaan skala ini dilibatkan dalam perhitungan *Euclidean distance*, maka penambahan dari nilai *width* ruang misalnya dari 15 menjadi 19 m, tidak banyak berpengaruh terhadap perubahan *distance* yang dihasilkan. Sebaliknya, perubahan pada *area* ruang, misalnya dari 250 menjadi 280 m^2 , akan berpengaruh terhadap *distance* yang dihasilkan. Untuk mengatasi masalah ini, normalisasi (*normalisation*) diperlukan.



Dalam *pattern matrix*, pola ke- i dapat ditunjukkan dengan vektor (kolom) x_i^* dan nilai ciri ke- j untuk pola ke- i dapat ditunjukkan oleh x_{ij}^* . Tanda bintang menunjukkan “raw” atau data yang belum dinormalisasi. Jika n adalah jumlah pola dan d adalah jumlah ciri, *pattern matrix*nya adalah $n \times d$ matrik A^* (Jain dan Dubes, 1988):

$$A^* = [x_1^* \ x_2^* \ \dots \ x_n^*]^T = \begin{pmatrix} x_{11}^* & x_{12}^* & \dots & x_{1d}^* \\ x_{21}^* & x_{22}^* & \dots & x_{2d}^* \\ \dots & \dots & \dots & \dots \\ x_{n1}^* & x_{n2}^* & \dots & x_{nd}^* \end{pmatrix}$$

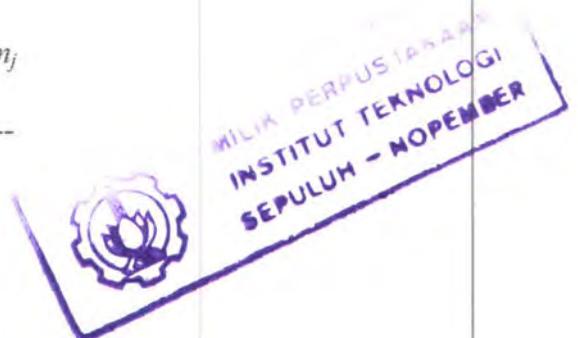
Tiap baris dari A^* adalah sebuah pola. Rata-rata ciri ke- j , yaitu m_j , dan varian ciri ke- j , yaitu s_j^2 , didefinisikan sebagai nilai tengah (rata-rata) dan varian ciri ke- j .

$$m_j = (1/n) \sum_{i=1}^n x_{ij}^*$$

$$s_j^2 = (1/n) \sum_{i=1}^n (x_{ij}^* - m_j)^2$$

Satu tipe dari normalisasi adalah dengan menggeser dan menskalakan sumbu-sumbu sehingga semua ciri mempunyai nilai tengah (rata-rata) nol dan varian satuan:

$$x_{ij} = \frac{x_{ij}^* - m_j}{s_j}$$





Jadi, x_{ij} adalah nilai dari ciri ke- j untuk pola ke- i yang telah dinormalisasi. Nilai ciri ini kemudian digunakan dalam perhitungan *Euclidean distance* antara ciri-ciri dari pola-pola. Setelah menentukan teknik-teknik *case retrieval* yang diperlukan, sebuah algoritma pengukuran kesamaan antara spesifikasi desain kapal lama (*design cases*) dan permintaan pemesan (input) dapat didefinisikan dan diuraikan dibawah ini.

3.5.3. Algoritma untuk Mendapatkan Ukuran Kesamaan Pola

1). Langkah 1:

Setelah melakukan pencarian contoh-contoh desain dalam *case base* dan menemukan contoh-contoh yang relevan dalam suatu kelompok, jika diberikan input berupa data ruang, maka input ini digabungkan dengan kelompok tersebut dan mereka kemudian diletakkan dalam kelompok sementara yang lain.

2). Langkah 2:

Lakukan normalisasi nilai-nilai dari ciri-ciri dari semua pola dalam kelompok sementara tersebut. Ini dilakukan pertama dengan perhitungan nilai tengah, m_j , untuk tiap ciri (ciri ke- j) dari semua pola. Varian, s_j^2 , untuk ciri ke- j dari semua pola juga dihitung. Semua nilai ciri-ciri yang dinormalisasi, x_{ij} , kemudian ditentukan.

3). Langkah 3:

Hitung *Euclidean distance* antara nilai-nilai dari semua ciri yang dinormalisasi dari pola input dan pola kelompok sementara. Hasil dari perhitungan ini



memberikan ukuran pertidaksamaan (*dissimilarity*) antara input dan pola contoh desain.

4). Langkah 4:

Urutkan hasil dari tahap 3 untuk mendapatkan ranking pertidaksamaan antara input dan pola contoh desain, dimana pola contoh desain dengan skor terendah diletakkan pada urutan teratas sedangkan skor tertinggi diletakkan pada urutan paling bawah. Ini berarti bahwa semakin tinggi skor, ketidaksamaan (perbedaan) antara pola input dengan pola contoh desain semakin besar, atau dengan kata lain semakin rendah skor, maka semakin kecil ketidaksamaan (perbedaan) antara keduanya atau mempunyai kesamaan yang lebih tinggi.

3.6. Penyusunan Data

Data Tugas Akhir ini data yang digunakan adalah data ruang akomodasi kapal penumpang. Masing-masing kapal diambil lima geladak yang dianggap memiliki karakteristik yang sama, sehingga data dikelompokkan menjadi lima berdasarkan geladak tersebut. Jadi data tersebut terbagi menjadi lima kelompok layout, yaitu *Upper_Deck*, *Poop_Deck*, *Boat_Deck*, *Bridge_Deck* dan *Navigation_Deck*. Masing-masing kelompok data memiliki jumlah data sebanyak jumlah data kapal. Data dari tiap geladak merupakan data yang berdiri sendiri dan tidak ada hubungan satu sama lain dalam proses *matching* untuk geladak yang berbeda. Dan masing-masing layout mempunyai beberapa ruangan dimana masing-masing ruang mempunyai atribut sendiri-sendiri.



3.7. Prosedur Pengolahan Data

Data diolah dengan prosedur sebagai berikut:

1) Data didefinisikan dalam suatu kelompok data (*class*)

Tujuan dari proses pendefinisian ini adalah untuk mengelompokkan data dalam satu bagian yang dapat diakses oleh program pengakses. Dalam *class* data-data input dibagi kedalam tiga kategori, yaitu *space name*, *space function* dan *space identifier*.

Ketiga kategori ini harus dimasukkan sebagai input untuk setiap ruangan. *Space name* merupakan nama dari masing-masing ruangan. *Space function* merupakan kegunaan dari setiap ruang. Sedangkan *space identifier* merupakan keterangan dari input data yang ditentukan oleh desiner, apakah data tersebut masuk dalam kategori *user input* atau data yang dimasukkan dalam *case base*. Seluruh data harus memiliki kesamaan nama, fungsi dan identifikasi untuk ruangan dengan kategori yang sama dari kapal yang berbeda. Hal tersebut dapat dilihat pada contoh berikut:

```
(defrooms      :name 'Laundry_UD01s
                :txt "Laundry"
                :space-func 'Laundry
                :sp-ident 'Stored_Object
                :gs (make-instance 'Rectangle
                                   :outline-rect '(50 50 1.8 3.5))
                :bb '(50 50 86 120)
                :area 2520
                :ar 1
                :w 1.8
                :h 3.5
                :wh-rat 0.514
                :col ':bisque)
```

Data yang digunakan dibagi menjadi empat *class* yang merupakan representasi dari lima geladak yaitu *Upper_Deck*, *Poop_Deck*, *Boat_Deck*,



Bridge_Deck dan Navigation_Deck. Faktor yang harus diperhatikan adalah setiap kelas tidak saling berhubungan dalam *graph*, sehingga proses *matching* terjadi hanya dalam masing-masing *class*.

2) Klasifikasi data dalam *case base*

Dalam proses *matching* ada dua macam input data yang dikenal dan digunakan, yaitu input eksternal dan input internal. Input eksternal biasa dikenal sebagai *user input*, dimana data input diberikan oleh perancang. Sedangkan input internal merupakan data yang sudah tersimpan dalam *case base*. *User input* dapat disimpan dalam internal input, dengan cara membagi data dalam *template* dan *stored-object*.

Stored-object adalah data yang tersimpan dalam *case base* dan berfungsi sebagai data pembanding dari *user input*. Data-data adalah kumpulan dari empat *class* yang sudah ditentukan sebelumnya, yaitu dari empat geladak yang ada.

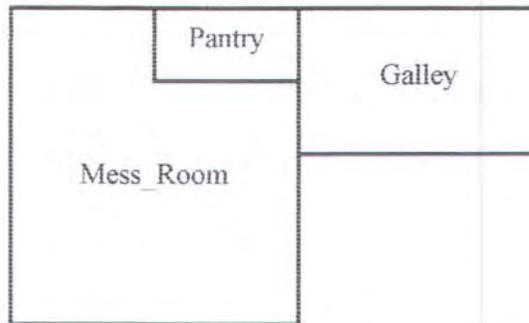
Template adalah *stored-object* yang dimodelkan sebagai *user input* dan digunakan sebagai dasar dari pengenalan pola yang berarti input pola ke dalam *pattern matching process* (Hall dan Matias, 1993).

Cara pembagiannya adalah dengan menentukan pilihan pada *space identifier* yang dapat berupa *template* atau *stored-object*. Jika *space identifier* dinyatakan sebagai *template* maka data akan dikategorikan sebagai *user input* dan jika *space identifier* berupa *stored-object* maka data akan dikategorikan sebagai data pembanding yang dimasukkan dalam *case base*.



3) Kedekatan antar ruang (*space closeness*)

Berdasarkan teknik yang dipakai, yang dimaksud dengan kedekatan antar ruangan adalah bila dua ruangan berdampingan dan dipisahkan oleh pembatas. Dalam graph kedekatan dilambangkan dengan garis yang menghubungkan antara dua ruang yang berdampingan. Jadi pada proses penyamaan (*matching*) dua layout maka harus memperhatikan hubungan ruangan yang berdekatan langsung. Sebagai contoh untuk tiga ruang seperti dalam Gambar 3.10. berikut:



Gambar 3.10. Gambar tiga ruang yang didefinisikan kedekatannya.

Maka kedekatannya didefinisikan sebagai berikut:

```
(deflayout :name 'Ship_02_Deck_UD
:comp-sp '(Mess_Room_UD02s Pantry_UD02s
Galley_UD02s)
:asg 'Ship_Upper_Deck
:sp-clos 'NIL)
```

4) Data matching

Data direpresentasikan dalam *graph* sebagai satu kesatuan yang terdiri dari *sub-graph* dan garis korespondensi. Setiap ruangan akan dinyatakan sebagai *sub-graph*. *Sub-graph* dinyatakan dengan *node* dan setiap ruangan yang



berbatasan akan dihubungkan dengan sebuah garis korespondensi yang melambangkan kedekatan antar ruangan. Gabungan dari beberapa *sub-graph* dan garis korespondensi menjadi suatu *graph*. *Graph* inilah yang nantinya digunakan dalam proses *topological pattern matching*. Dalam *sub-graph* tersebut terdapat data yang merepresentasikan suatu *binary tree* yang nantinya digunakan untuk proses *shape matching*.

3.8. Implementasi Pendekatan *Case Based Reasoning in Design*

Sesuai dengan batasan masalah, pada tugas akhir ini pendekatan CBR *in Design* diimplementasikan sebatas penarikan (*recalling*) *case* pada *case base*, yaitu dengan menggunakan teknik *topological and numerical pattern matching*. Dimana teknik-teknik tersebut telah dijelaskan pada sub-bab- sub-bab sebelumnya.

Sebagai data dalam *case base* adalah layout ruang akomodasi ABK dari kapal-kapal Tanker dengan *range* 1500 sampai dengan 6000 DWT. Data tersebut disimpan dalam *case base* dalam bentuk koordinat dari ruangan-ruangan penyusun layout beserta *properties* yang menunjukkan ciri dari ruangan tersebut, data kedekatan yang menunjukkan topologi dan data ruangan tersebut yang disusun dalam layout.

Proses penarikan dilakukan dengan mengacu pada kesamaan topologi dan bentuk dengan teknik *association graph matching* untuk memperoleh kesamaan topologi dan teknik *planar shape matching* untuk memperoleh



kesamaan geometrik. Dengan input adalah suatu sketsa layout., dimana sebagai data input sama dengan data yang digunakan sebagai data cases.

Hasil yang akan diperoleh adalah data layout dengan urutan data yang mempunyai kesamaan tertinggi dengan sketsa layout sampai data yang mempunyai tingkat kesamaan terendah. Hasil inilah yang nantinya dijadikan desain awal oleh desainer, dimana desainer bebas memilih layout mana yang diambil. Jadi implementasi teknik-teknik diatas mempunyai manfaat bagi desainer menunjukkan acuan atau memilihkan desain yang sesuai dengan keinginannya. Hal ini karena semakin layout tersebut mempunyai tingkat kemiripan yang tinggi, maka layout tersebut semakin mendekati keinginan desainer.

3.9. Algoritma dari *Topological and Numerical Pattern Matching*

Untuk menentukan urutan layout pada *case base* yang memiliki tingkat kesamaan tertinggi terhadap layout input, maka algoritma yang digunakan adalah sebagai berikut:

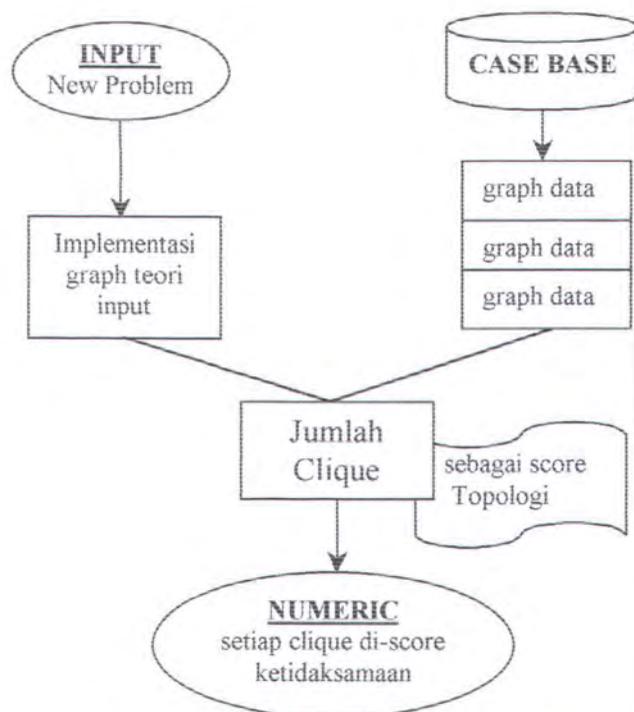
- 1). Apabila terdapat layout input yang akan dicari kesamaannya terhadap layout *case base* maka dilakukan *scoring* terhadap kedekatan antar ruangan secara topologi. Baik layout input maupun layout *case base* diimplementasikan dalam *graphs* seperti dalam Gambar 3.4. dan diperoleh hasil *matching* keduanya juga dalam *graphs* seperti dalam Gambar 3.5.
- 2). Hasilnya berupa "*Clique*" yaitu suatu set banyaknya pasangan ruangan yang bersesuaian hubungan fungsionalnya antara layout input dengan layout *case base* yang dihasilkan dari *topological pattern matching*.



- 3). Langkah selanjutnya ialah melakukan *scoring* terhadap perbedaan masing-masing pasangan dalam *clique* dengan cara *numerical pattern matching* yaitu menggunakan atribut numerik seperti dalam Gambar 3.6 . Hal ini dilakukan untuk memperoleh jumlah nilai numerik dari yang tertinggi sampai yang terendah.
- 4). Menampilkan urutan layout dari yang nilainya paling besar ke layout yang nilainya paling kecil. Hal ini dimaksudkan agar *owner* dapat menentukan sendiri layout mana yang akan dipilih dengan pertimbangan topologi maupun numerik.

Langkah-langkah algoritma diatas dapat dijelaskan dalam Gambar 3.11.

berikut ini.



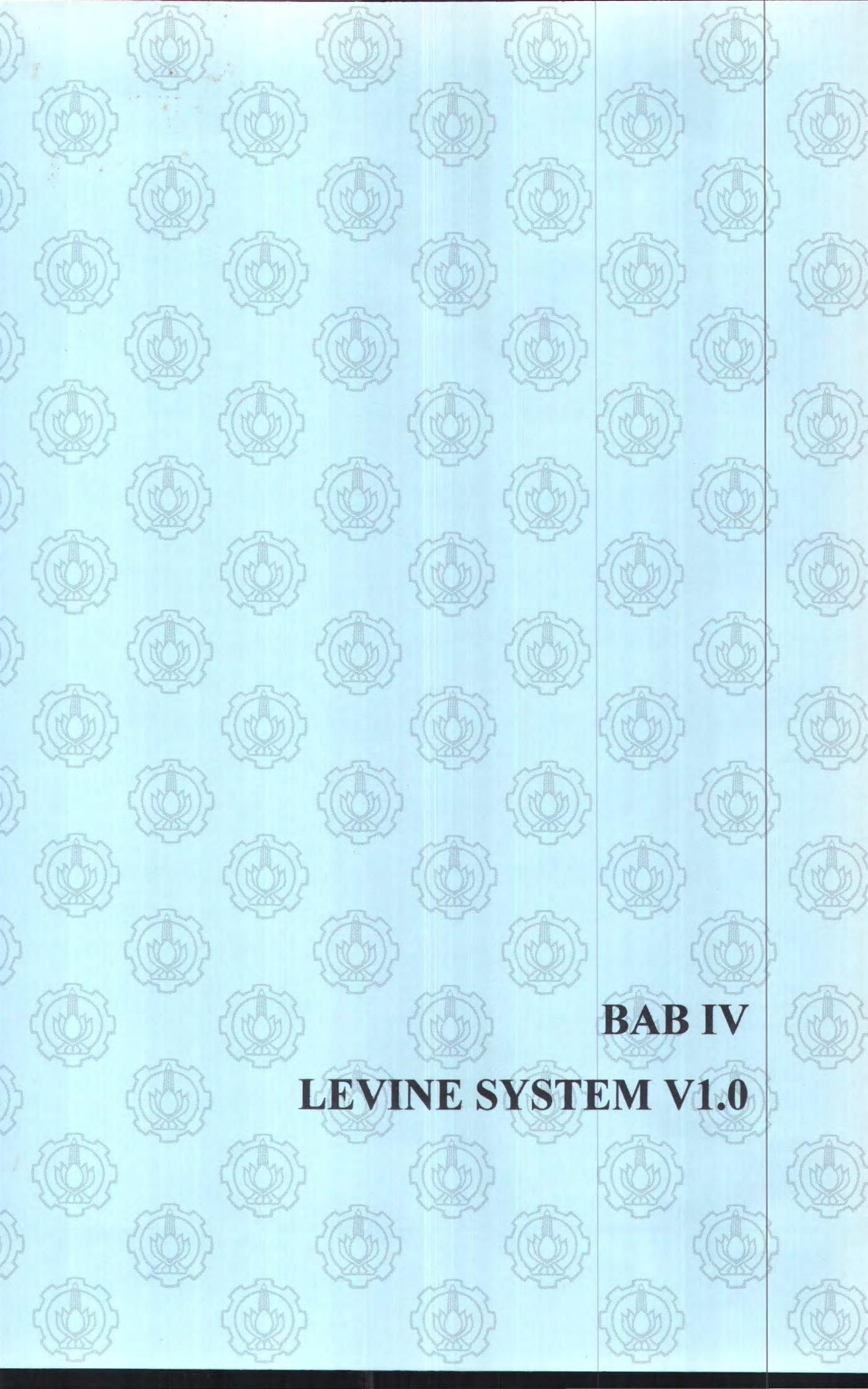
Gambar 3.11. Algoritma dari *topological* dan *numerical pattern matching*.



Untuk lebih memperjelas algoritma di atas maka akan penulis berikan contoh dalam Tabel 3.1. yaitu tampilan *score* layout input terhadap layout *case base* setelah dilakukan *topological* dan *numerical pattern matching*.

Tabel 3.1. Contoh urutan layout dari hasil *topological* dan *numerical pattern matching*.

Input	Case Base	Topologi Score/Clique	Numerik Score	Rank
Layout1	Case1	5 pairs	0.01	1
	Case2	5 pairs	0.55	2
	Case3	4 pairs	0.34	3
	Case4	3 pairs	0.26	4
	Case5	3 pairs	0.76	5
	Case6	1 pairs	0.92	6



BAB IV
LEVINE SYSTEM V1.0

**BAB IV****LEVINE SYSTEM V1.0**MILIK PERPUSTAKAAN
INSTITUT TEKNOLOGI
SEPULUH - NOPEMBER**4.1. LEVINE System V1.0**

LEVINE System V1.0 adalah suatu sistem yang mengimplementasikan pendekatan *Case Based Reasoning* dengan memakai teknik *topological and numerical pattern matching*. Sistem ini memanfaatkan desain-desain lama untuk proses penyelesaian desain baru dengan cara penarikan (*recall*) desain layout yang memiliki kesamaan topologi (kedekatan ruangan) dan kesamaan numerik.

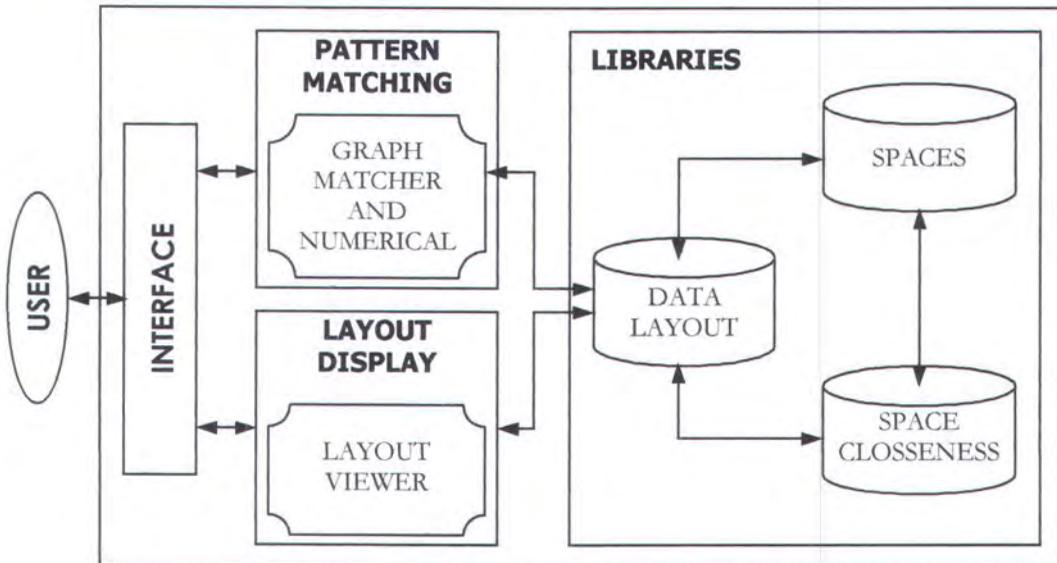
Sebagai salah satu alat dalam *pattern matching*, LEVINE System V1.0 mampu melakukan penarikan (*recall*) layout dari case base hingga diperoleh desain layout yang hampir sama (*most similar*) dengan desain input. Hal ini dilakukan dengan cara apabila pada sistem diberikan input berupa layout *template*, maka sistem akan mengadakan proses pencarian data layout akomodasi yang telah tersimpan di dalam *case base*. Dengan mencari kesesuaian antara input dengan data dalam *case base* secara topologi, akan diperoleh set pasangan ruangan yang memiliki angka kesamaan tertinggi (*clique*). Kemudian setiap pasangan ruangan tersebut akan di-scoring secara numerik sesuai dengan nilai atribut yang telah dimilikinya. Hasilnya berupa urutan layout yang memiliki angka kesamaan tertinggi sampai terendah dan dapat ditampilkan desain yang paling sesuai (*most similar*) dengan layout input.

Dalam bab ini akan dijelaskan dukungan LEVINE System V1.0 terhadap pemanfaatan desain lama (*design reuse*) untuk proses *pattern matching*.



4.2. Pemodelan Sistem

Pemodelan LEVINE System V1.0 terdiri dari kumpulan data (*libraries*), pola penyamaan (*pattern matching*), *layout viewer* dan *interface*. Pemodelan tersebut ditunjukkan dalam Gambar 4.1. di bawah ini.



Gambar 4.1. *Architecture* LEVINE System V1.0.

Sistem ini merupakan implementasi dari Program *Harlequin Lispworks* (*Common Lisp*) (*The Harlequin Group Limited* 1994) dan digunakan pada komputer *Pentium(r) II Processor Intel MMX (TM) Technology*, 64.0 MB RAM, *Memory* 1.99 GB, berbasis *Windows*.

4.3. Libraries

Bertujuan untuk menampilkan desain layout tertentu, penarikan data layout setelah diberikan input yang berupa *layout template*. *Libraries* terdiri dari data layout, data ruang (*spaces*) dan data kedekatan antar ruang (*space closeness*).



Data layout dihubungkan secara langsung dengan komponen lain dalam sistem, kecuali *interface*. *Libraries* adalah komponen dalam layout *cases* yang menghubungkan data layout dengan data ruang dan data kedekatan ruangan. *Libraries* dapat diperbarui dan dikembangkan dengan memasukkan data-data baru.

Berikut ini merupakan penjelasan mengenai komponen-komponen penyusun *libraries*:

Data Layout

Isi dari data layout berupa kumpulan desain layout akomodasi kapal tanker yang dipresentasikan dengan beberapa *attribut-value* seperti *name*, *component space closeness*, *assignment* dan *space closeness*.

Data Ruang (Spaces)

Berisi kumpulan ruangan pada akomodasi yang merupakan komponen dari data layout. Setiap ruangan tersebut memiliki *unique identifier* yang berbeda satu sama lain, tergantung pada *deck* dan layout dimana ruang itu berada.

Data Kedekatan Ruang (Space Closeness)

Berisi kumpulan komponen kedekatan antar ruang dalam data layout, dimana hubungan kedekatan antar ruang adalah *unique*.

4.4. *Layout Viewer*

Merupakan komponen dari LEVINE System V1.0 yang digunakan untuk menampilkan gambar desain layout (*display*), baik layout dalam *cases* maupun layout *template* sesuai dengan kebutuhan *user* pada layar komputer.



4.5. *Pattern Matching*

Adalah komponen yang digunakan untuk penarikan desain layout dalam *cases* setelah diberikan layout input dari desainer. *Graph matcher* digunakan untuk mencari kesesuaian antara input dan data dalam *case base* berdasarkan pada *topological* dan *numerical pattern matching*. Dalam *pattern matching* jenis ini bentuk penyamaan (*matcher*) akan berjalan jika hubungan *pattern matching* dari layout telah lengkap.

4.6. *Sistem Interface*

Sistem *interface* memberikan fasilitas bagi desainer maupun *user* untuk dapat mengoperasikan LEVINE System V1.0 melalui tampilan *windows* yang sederhana dan mudah dipahami.

Tampilan awal dari LEVINE System V1.0 adalah *interface* utama (*main menu*) seperti terlihat dalam Gambar 4.2.

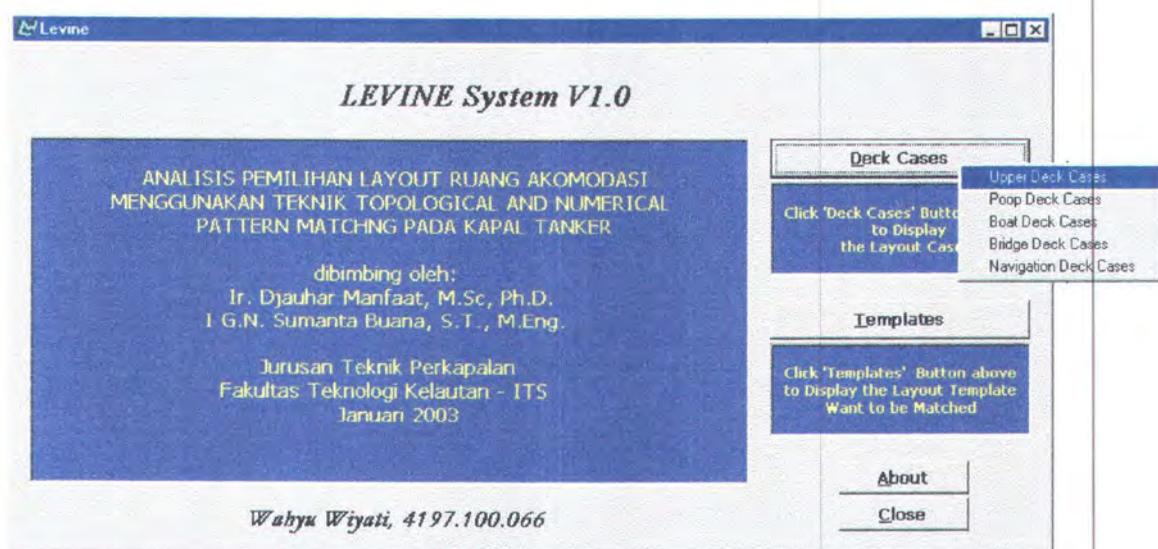


Gambar 4.2. Tampilan *interface* utama (*main menu*).



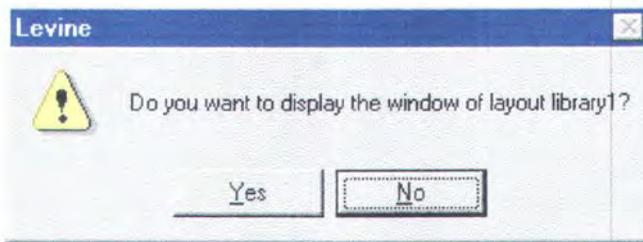
Dimana didalamnya terdapat 4 buah tombol (*button*), yaitu:

- 1). Tombol **Deck Cases**, untuk menampilkan data-data (*layout cases*) yang ada pada sistem ini, dan akan muncul lima kelompok data sesuai dengan lokasi (pada geladak mana layout itu berada) dari layout. Jika tombol ini ditekan, akan menampilkan menu pilihan kelompok layout data seperti gambar berikut:



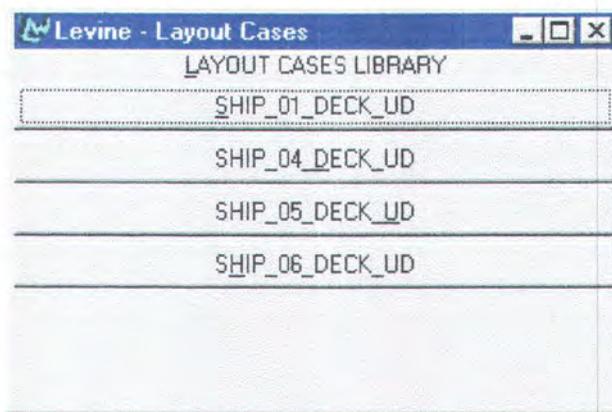
Gambar 4.3. Menu pilihan dari tombol **Deck Cases**.

Terdapat empat pilihan yaitu **Upper Deck Cases**, **Poop Deck Cases**, **Boat Deck Cases**, **Bridge Deck Cases** dan **Navigation Deck Cases**, yang masing masing terdiri dari beberapa layout dengan karakteristik yang sama. Misalnya dipilih (*Click*) **Upper Deck Cases**, akan tampil pilihan layout untuk ditampilkan dilayar. Yang sebelumnya akan tampil dialog seperti ditunjukkan dalam Gambar 4.4. berikut:



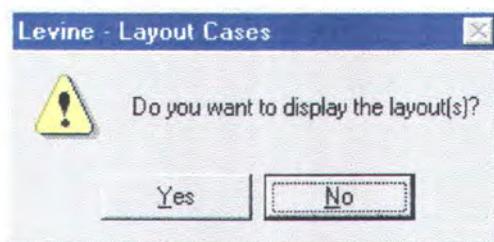
Gambar 4.4. Dialog untuk menampilkan pilihan layout cases.

Jika dipilih tombol **Yes** maka akan tampil pilihan layout seperti ditunjukkan dalam Gambar 4.5. berikut:



Gambar 4.5. Pilihan layout untuk semua data layout pada geladak Upper Deck.

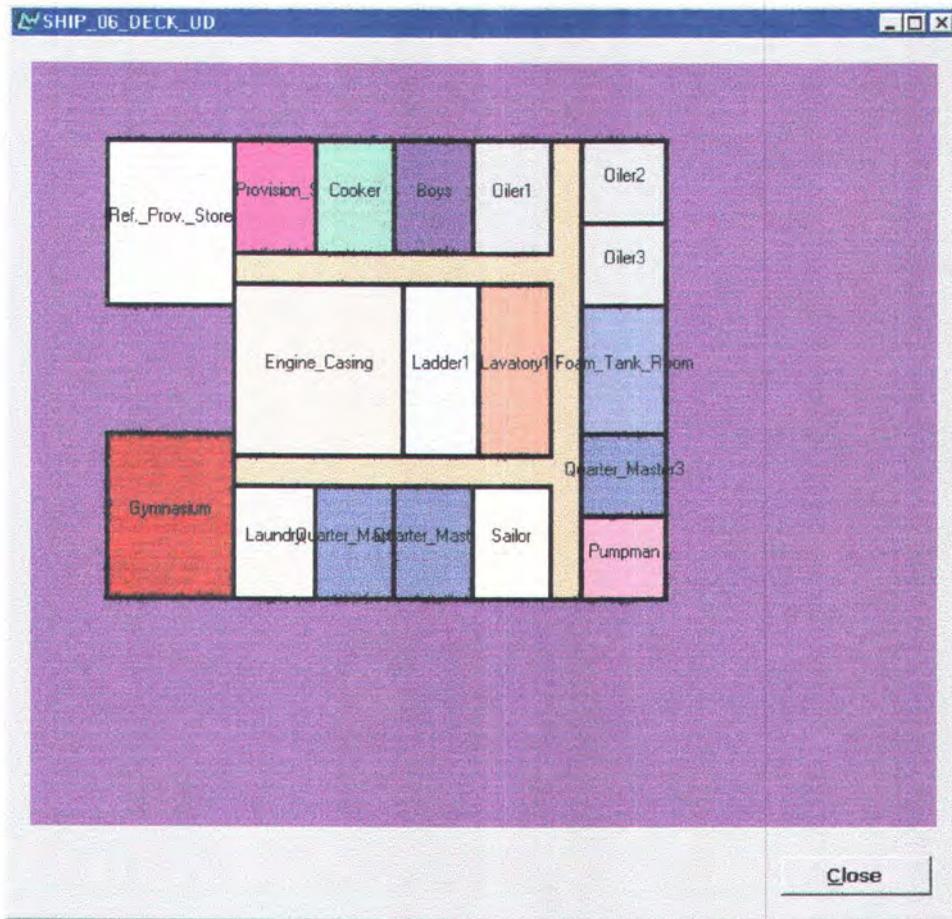
Ditunjukkan ada empat buah data layout pada Upper Deck. Dengan memilih salah satu akan ditampilkan gambar layoutnya, dan sebelumnya akan muncul dialog lagi seperti dalam Gambar 4.6. berikut:



Gambar 4.6. Dialog untuk menampilkan gambar layout.

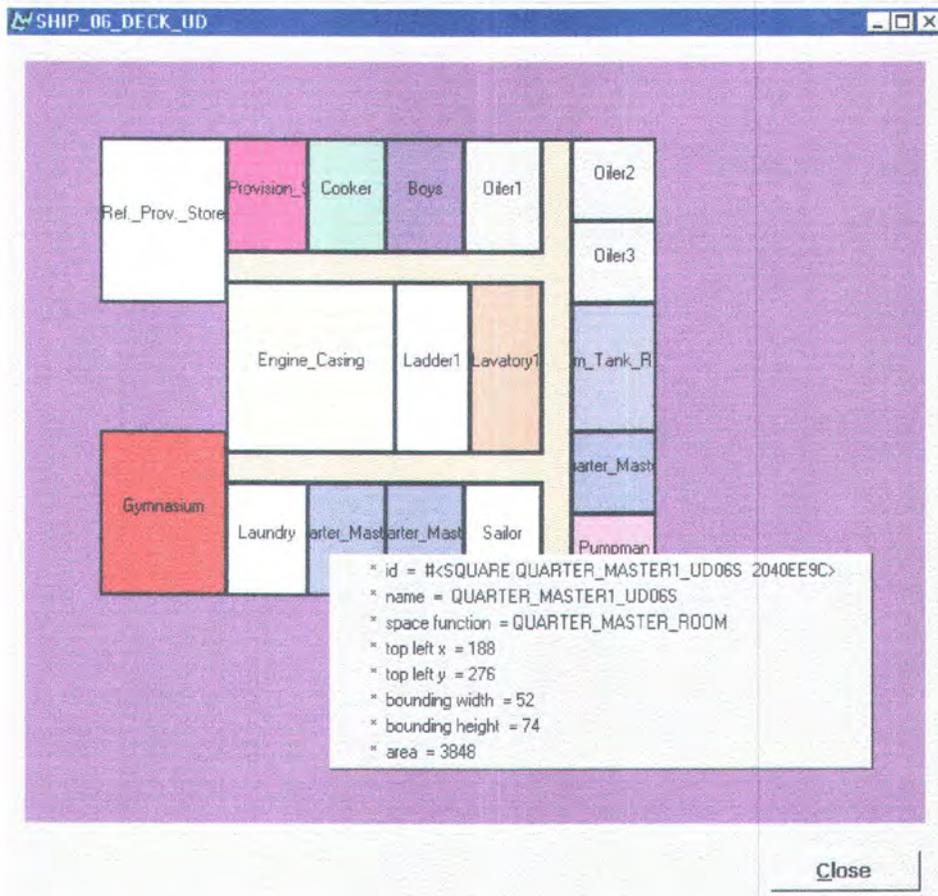


Selanjutnya akan muncul gambar dari layout yang dipilih seperti dalam Gambar 4.7. berikut:

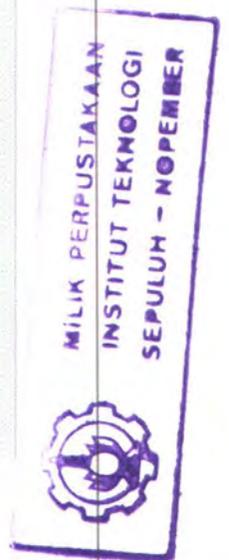


Gambar 4.7. Gambar layout yang telah dipilih.

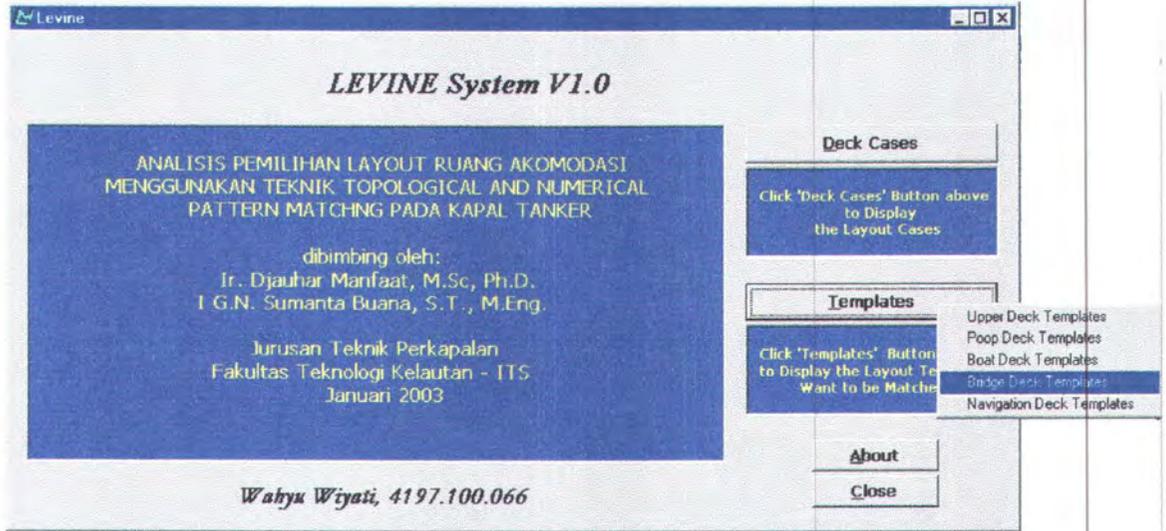
Dalam tampilan *window* gambar layout di atas terdapat tombol **C**lose digunakan untuk menutup *window* tersebut. Selain atribut dari masing-masing ruangan dalam layout tersebut dapat ditampilkan dengan menekan tombol kanan *mouse* pada posisi *pointer* di atas ruang yang ingin dilihat atributnya, seperti ditunjukkan dalam Gambar 4.8.



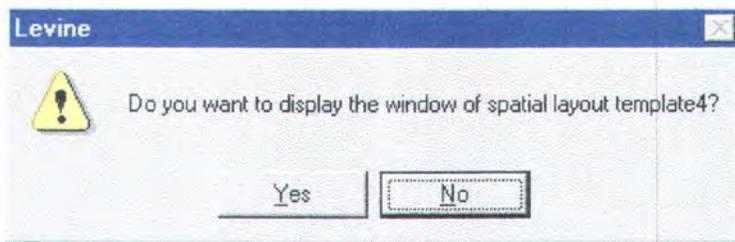
Gambar 4.8. Tampilan atribut untuk **Quarter_Master_UD06s**.



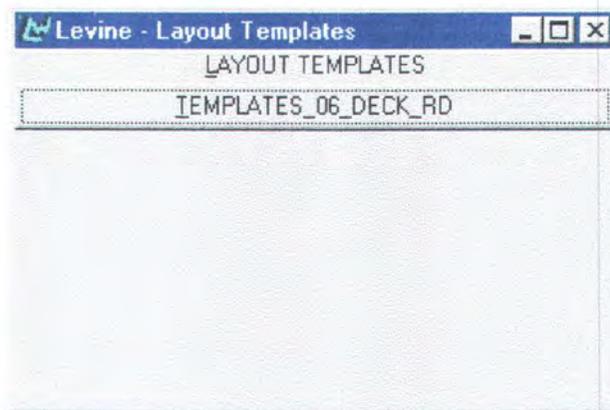
- 2). Tombol **Template**, akan menampilkan *layout input* untuk proses *pattern matching* yang dinamakan data *template* (Hall dan Matias, 1993). Tidak jauh berbeda dengan tombol **Deck Cases**. Hanya saja jika ditelusuri sampai tampilan *window layout*, pada tampilan *window layout*nya terdapat tombol **Match**, yang berfungsi untuk mengaktifkan *pattern matcher* untuk melakukan tugasnya mencari layout yang mempunyai tingkat kemiripan tertinggi dengan layout yang ditampilkan pada data layout yang ditampilkan oleh tombol **Deck Cases**. Hal ini akan ditunjukkan dalam Gambar 4.9., Gambar 4.10., Gambar 4.11., Gambar 4.12. dan Gambar 4.13. berikut ini:



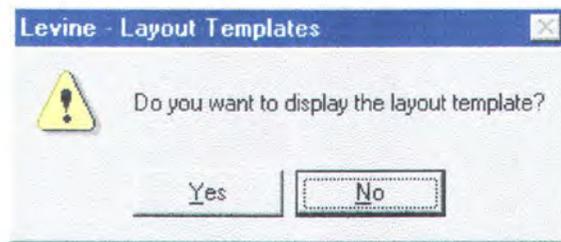
Gambar 4.9. Menu pilihan dari tombol Templates.



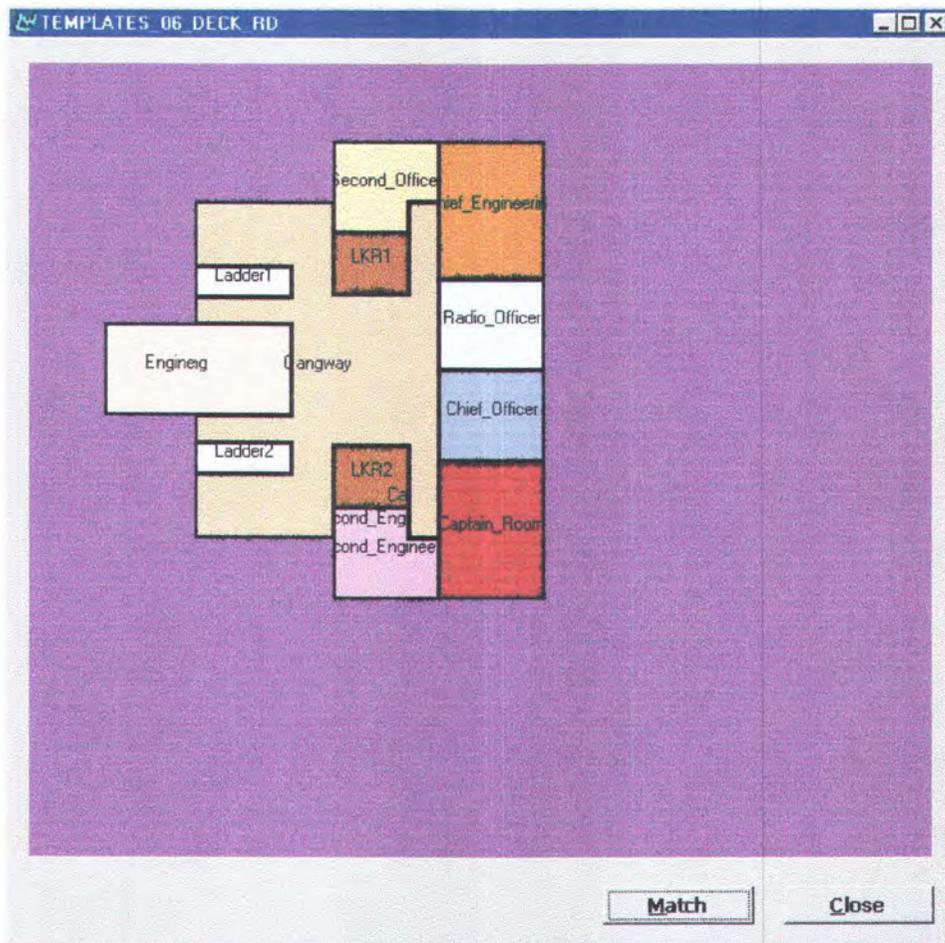
Gambar 4.10. Dialog untuk menampilkan pilihan layout *template*.



Gambar 4.11. Pilihan layout untuk layout *template* pada Bridge Deck.



Gambar 4.12. Dialog untuk menampilkan gambar layout *template*.

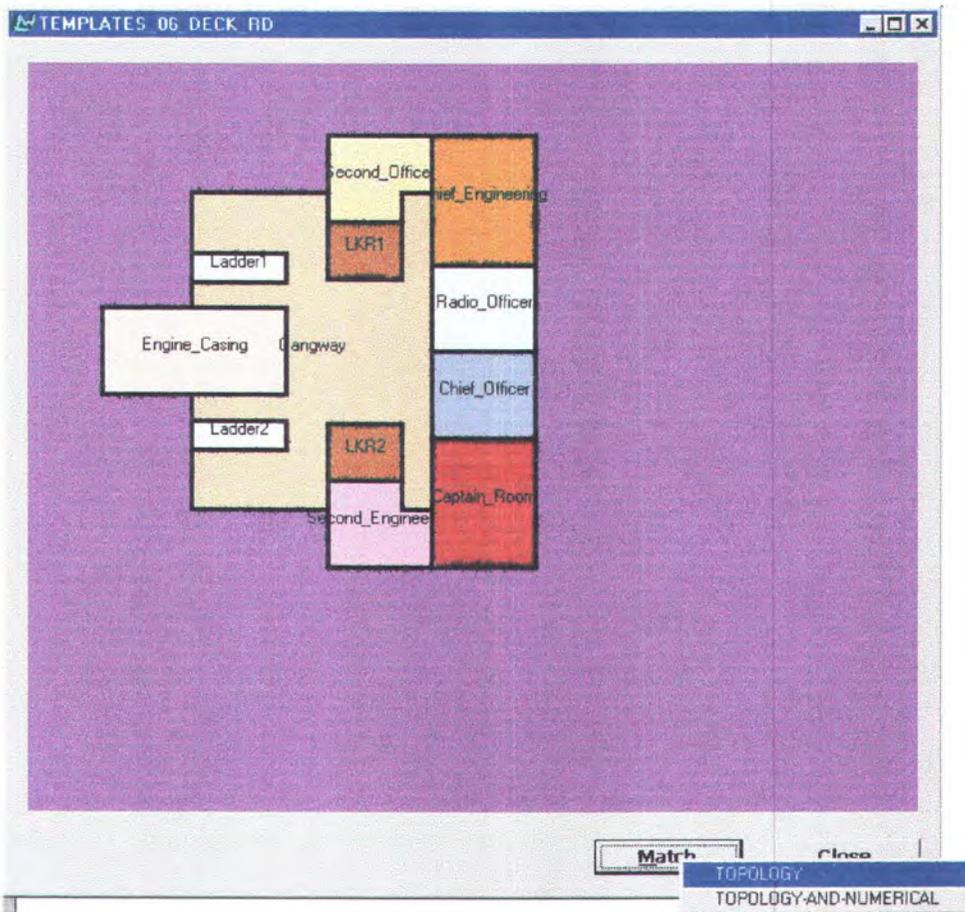


Gambar 4.13. Gambar layout *template* yang telah dipilih, siap untuk diproses.

Jika ditekan tombol **Match** akan muncul dua pilihan yaitu **TOPOLOGY** dan **TOPOLOGY-AND-NUMERICAL** yang merupakan pilihan untuk proses *pattern matching*. Jika seperti dipilih **TOPOLOGY** maka proses *matching* akan



menggunakan metode *topological pattern matching* yaitu menggunakan teknik *association graph matching* untuk penyamaan antar dua layout. Sedangkan bila TOPOLOGY-AND-NUMERICAL dipilih maka proses *matching* akan menggunakan metode *topological and numerical pattern matching* yang menggunakan teknik kombinasi antara *association graph matching* untuk penyamaan antar dua layout dan *numerical scoring* untuk memperoleh nilai dari pasangan ruangan sehingga dapat dilihat ketidaksamaan (*disimilarity*) antar pasangan ruangan tersebut. Pilihan tersebut ditunjukkan dalam Gambar 4.14. berikut:

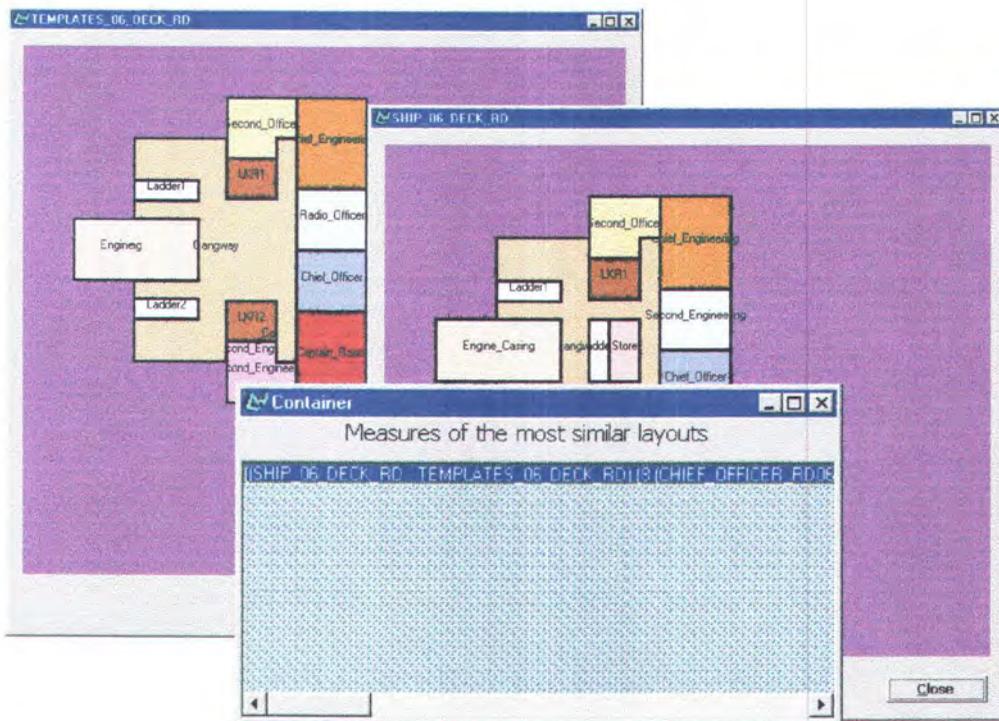


Gambar 4.14. Pilihan proses *matching* pada suatu *template*.



a. Pilihan **TOPOLOGY**

Seperti diuraikan diatas pilihan ini akan menghasilkan layout dengan metode *topological pattern matching*. User akan diberi pilihan apakah ingin mendapatkan hasil layout dengan tingkat kesamaan yang paling tinggi atau menarik layout dengan jumlah tertentu. Jika memilih dengan jumlah tertentu maka user akan memperoleh layout dengan urutan tingkat kesamaan yang paling tinggi sampai dengan tingkat kesamaan terendah sejumlah yang ia inginkan tadi. Untuk pilihan dengan layout yang mempunyai kesamaan tertinggi hasilnya ditunjukkan dalam Gambar 4.15. berikut:

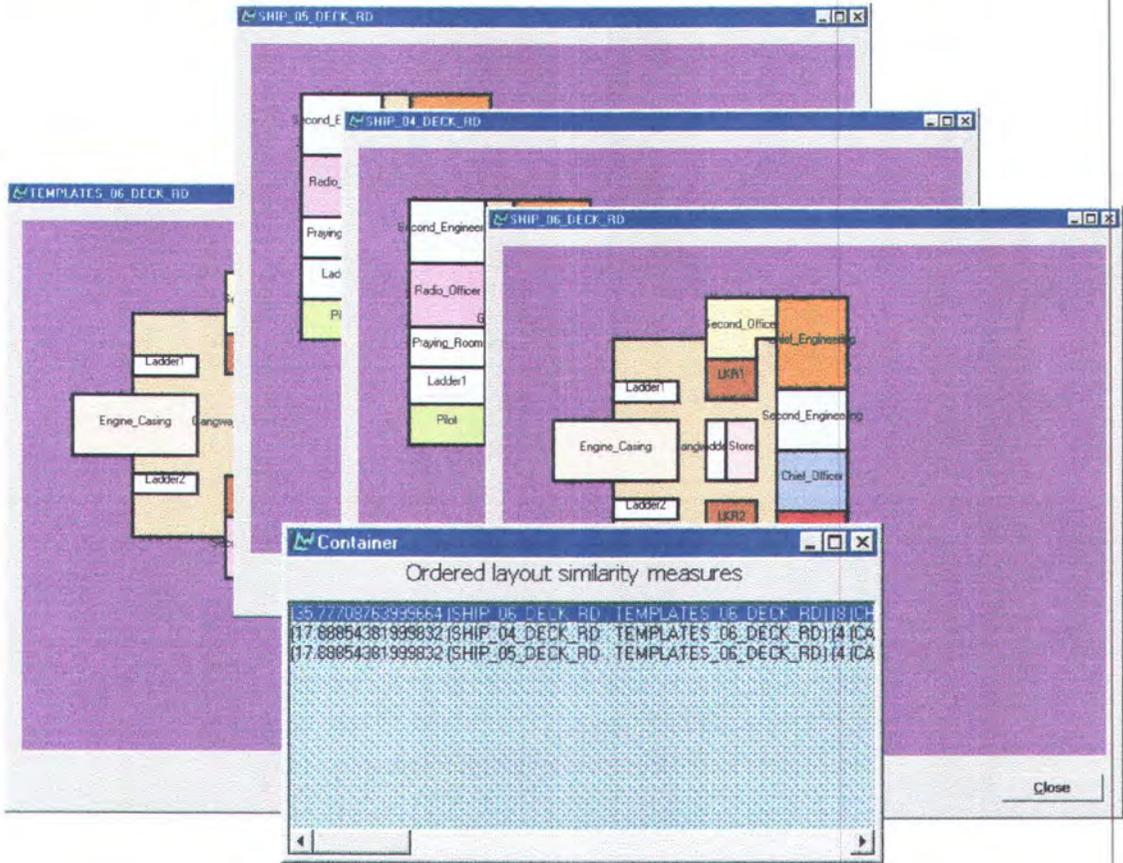


Gambar 4.15. Contoh hasil proses pilihan **TOPOLOGY** dengan hasil layout yang mempunyai tingkat kesamaan tertinggi secara topologi.

**b. Pilihan TOPOLOGY-AND-NUMERICAL**

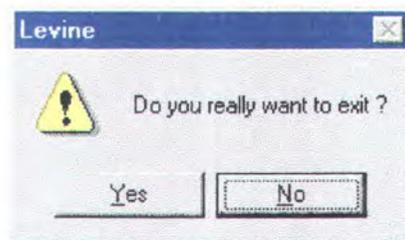
Pilihan ini adalah inti dari LEVINE System V1.0. Seperti halnya dengan pilihan **TOPOLOGY**, pilihan **TOPOLOGY-AND-NUMERICAL** ini akan memberikan hasil layout yang mempunyai tingkat ketidaksamaan (*dissimilarity*) tertinggi atau beberapa layout dengan *user* input dan hasilnya adalah urutan dari layout yang mempunyai tingkat ketidaksamaan tertinggi sampai terendah sejumlah yang diinput oleh *user*. Perbedaan yang akan muncul adalah pada *list* yang dihasilkan *window* bernama *container*. Perbedaan itu adalah adanya *score* untuk hasil proses *topological and numerical pattern matching*. Gambar 4.16. berikut mengacu pada contoh hasil *running* jika *user* input adalah untuk tiga layout yang mempunyai tingkat ketidaksamaan tertinggi.

Dalam *window container* terlihat hasil penarikan layout dari *case base*. Disana ditunjukkan urutan hasil layout, dari layout yang mempunyai kesamaan tertinggi sampai terendah. *Score*, jumlah *clique* nama layout dan nama-nama ruang yang berkorespondensi dapat dilihat pada *window* tersebut.



Gambar 4.16. Contoh hasil proses pilihan **TOPOLOGY-AND-NUMERICAL** dengan hasil tiga layout yang mempunyai tingkat kesamaan tertinggi secara geometrik dan topologi.

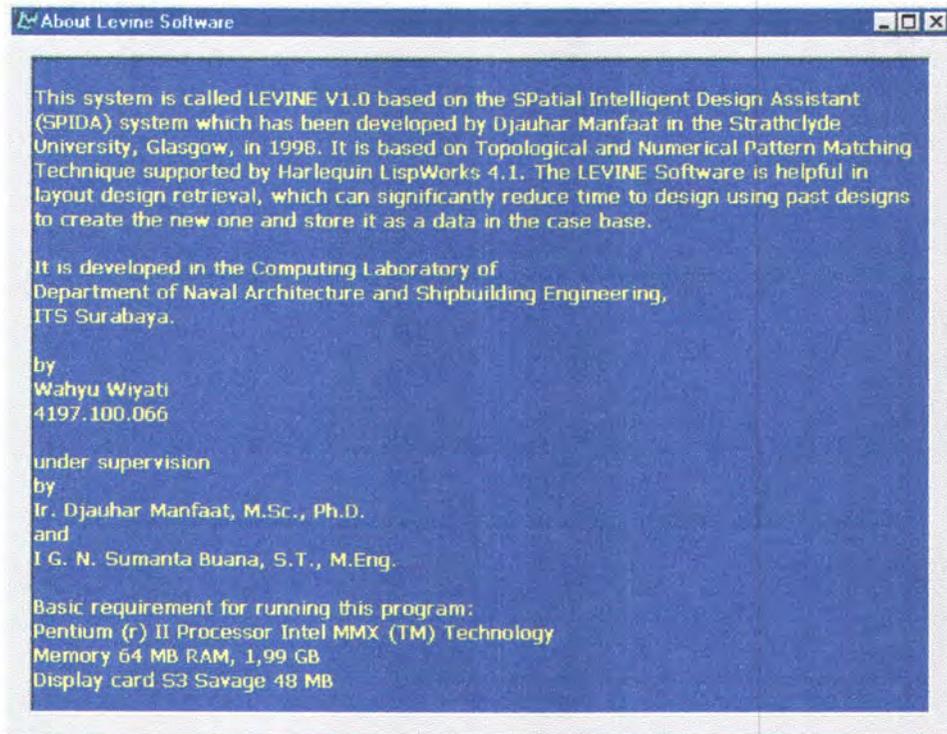
- 3). Tombol **C**lose, berfungsi untuk menutup program. Bila diklik akan keluar tampilan dialog seperti ditunjukkan dalam Gambar 4.17. berikut:



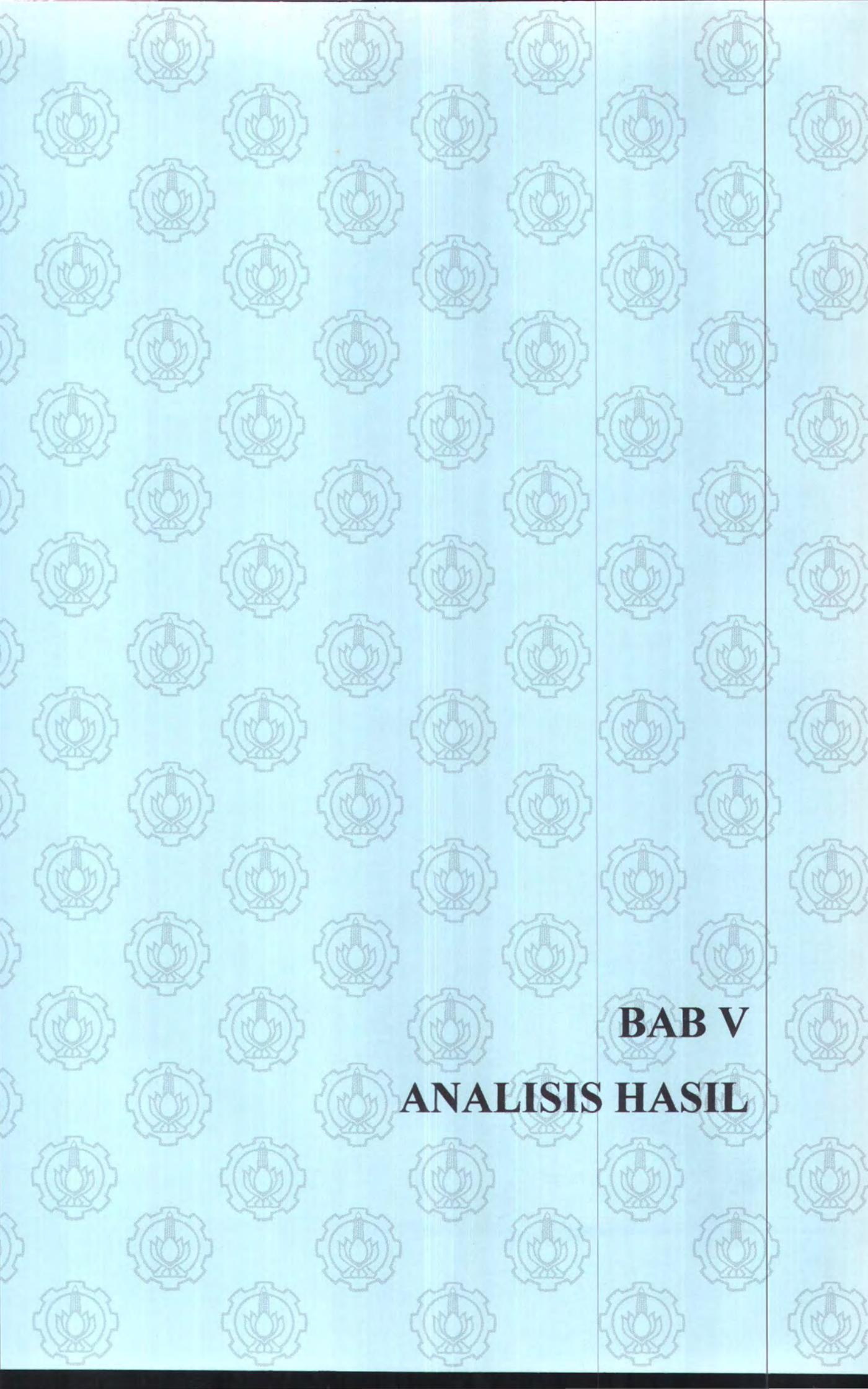
Gambar 4.17. Tampilan dialog untuk menutup program.



- 4). Tombol **About**, untuk menampilkan penjelasan tentang LEVINE System V1.0, seperti yang ditunjukkan dalam Gambar 4.18. berikut ini:



Gambar 4.18. Isi dari About berupa penjelasan program.



BAB V
ANALISIS HASIL



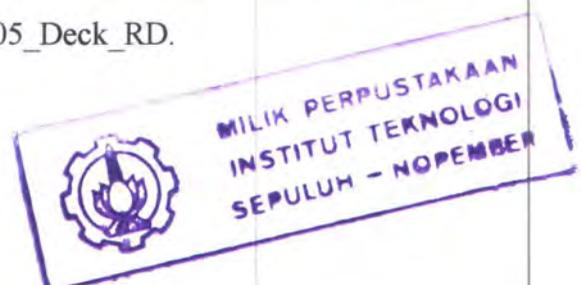
BAB V

ANALISIS HASIL

Untuk keperluan analisis, digunakan data *template* yang secara *visual* mempunyai tingkat kesesuaian yang berbeda-beda sehingga dapat diketahui apakah hasil yang dicapai nantinya sesuai dengan hipotesa awal atau tidak. Setelah dilakukan *running* program untuk *matching* semua data *template*, hasil yang diperoleh ditabulasi sehingga mempermudah analisis. Tabel tersebut merupakan tabel kesesuaian yang berisi:

- urutan data layout kapal berdasarkan angka kesamaan hasil proses *matching* yang disusun berurutan dari *clique* tertinggi hingga terendah,
- angka kesamaan (*clique*) antara *layout template* dengan masing-masing data layout kapal,
- nama ruangan (*space*) hasil proses *matching* yang memiliki korespondensi (topologi), dan
- angka kesamaan (*score*) total semua *clique* antara *layout template* dengan masing-masing data layout kapal.

Dari proses *running* program, diperoleh hasil seperti pada tabel 5.1. masing-masing untuk hasil *matching* antara layout pada *Template_06_Deck_RD* dengan *Ship_06_Deck_RD*, *Template_06_Deck_RD* dengan *Ship_04_Deck_RD* dan *Template_06_Deck_RD* dengan *Ship_05_Deck_RD*.





Tabel 5.1. Hasil *matching* antara data layout Template_06_Deck_RD dengan Ship_06_Deck_RD, Ship_04_Deck_RD dan Ship_05_Deck_RD.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Numerical Score
Temp_06_Deck_RD	Ship_06_Deck_RD	8	Chief Officer Room	35.77708764
			Loker 2	
			Ladder 3	
			Loker 1	
			Ladder 2	
			Ladder 1	
			Engine Casing	
Temp_06_Deck_RD	Ship_04_Deck_RD	4	Captain Room	17.88854382
			Second Engineering Room	
			Chief Engineering Room	
			Ladder 1	
Temp_06_Deck_RD	Ship_05_Deck_RD	4	Captain Room	17.88854382
			Second Engineering Room	
			Chief Engineering Room	
			Ladder 1	

Bila kita tinjau tabel 5.1 disana ditunjukkan hasil *matching* antara layout Temp_06_Deck_RD dengan cases layout geladak kedua. *Template* ini sengaja dibuat mirip secara visual dengan Ship_06_Deck_RD, dan hanya ada sedikit perbedaan ukuran ruang. Secara topologi jelas menunjukkan hubungan yang sama. Secara logika, jika dilakukan *matching* maka *case* yang akan menempati urutan pertama kemiripannya adalah Ship_06_Deck_RD. Dan setelah dilakukan



running program menunjukkan hasil sesuai dengan logika tadi dengan waktu yang amat singkat dan *score* tertinggi serta jumlah *clique* sebanyak ruangan. Hanya dalam beberapa detik semua *case* telah ditampilkan sesuai dengan urutan kemiripan (*similarity*).

Pada tabel 5.1 untuk *matching* antara Temp_01_Deck_RD dengan *cases* pada Bridge Deck (RD) dimana memiliki tiga *cases* yang memiliki tingkat kesamaan topologi yang berbeda (pertama bernilai 8 sedangkan kedua dan ketiga bernilai 4 *clique*). Menunjukkan bahwa jika hasil *matching* menunjukkan jumlah *clique* yang sama (secara topologi mempunyai tingkat kesamaan yang sama) maka tingkat kesamaan ditunjukkan oleh *score* dari kesamaan bentuk. Dimana *score* yang terbesar adalah yang memiliki tingkat kesamaan tertinggi.

Ada kasus yang menarik pada tabel 5.1 yang hanya menunjukkan perbedaan pada bentuk. Yaitu pada hasil *matching* antara Temp_06_Deck_RD dengan *cases* pada Bridge Deck (RD) dihasilkan dua layout yang mempunyai tingkat kesamaan topologi yang sama dengan jumlah *clique* empat, dimana masing-masing *clique* dari kedua layout hasil tersebut pada ruangan-ruangan yang mempunyai korespondensi yang sama (punya fungsi yang sama) kemudian hasil dari topologi tersebut dihitung nilai dari *numerical score* masing-masing. Hal ini menunjukkan performa dari LEVINE System V1.0 dalam menangani penyamaan bentuk (*shape matching*) sangat baik.

Dari uraian diatas menunjukkan teknik *topological and numerical pattern matching* sangat baik dalam memberikan hasil dari proses *recalling* suatu *case* dari *case base*. Hal ini ditunjukkan dengan hasil urutan yang kemungkinan



menduanya kecil, yang sangat penting sekali dalam pengambilan keputusan terhadap desain layout hasil *pattern matching*. Dengan menggunakan komputer akan memberikan kecepatan yang tinggi dan pengurutan yang baik dari suatu proses *recalling*. Sehingga dapat dikatakan bahwa LEVINE System V1.0 teruji dengan baik dalam melakukan proses *matching* dan mempunyai manfaat yang besar dalam proses desain layout.

Selanjutnya dalam hal ini desainer/*user* bebas memilih apakah yang dipakai adalah layout pada kapal yang mempunyai kesamaan tertinggi kemudian dilakukan modifikasi (*adaptation*) dari desain tersebut atau desainer mengambil yang mempunyai urutan rangking yang kedua atau desain yang lain. Atau desainer akan mengambil desain yang mempunyai nilai kesamaan yang tertinggi lalu akan mengambil bentuk-bentuk kesamaan-kesamaan dari desain-desain yang lain dan sebagainya. Dalam pemilihan desain tersebut desainer mempunyai pertimbangan-pertimbangan lain serta batasan-batasan tertentu yang dapat membantu desainer dalam mengambil keputusan.

Untuk memberikan analisis yang lebih akurat mengenai hasil akhir LEVINE System V1.0 ini maka akan ditampilkan beberapa hasil *topological and numerical pattern matching* untuk beberapa deck dalam bentuk tabel, sebagai berikut:



Tabel 5.2. Hasil *matching* antara data layout Template_06_Deck_PD dengan Ship_06_Deck_PD, Ship_05_Deck_PD, Ship_04_Deck_PD, Ship_03_Deck_PD, Ship_02_Deck_PD, dan Ship_01_Deck_PD.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Numerical Score
Temp_06_Deck_PD	Ship_06_Deck_PD	6	Crew Messroom Lavatory Spare Hospital Engine Casing Officer Messroom	26.83282
Temp_06_Deck_PD	Ship_05_Deck_PD	4	CO2 Room Hospital Ladder1 Galley	17.88854
Temp_06_Deck_PD	Ship_04_Deck_PD	4	Spare Hospital Ladder Galley	17.88854
Temp_06_Deck_PD	Ship_03_Deck_PD	4	Lavatory Spare Ladder1 Galley	17.88854
Temp_06_Deck_PD	Ship_02_Deck_PD	4	Lavatory Spare Ladder1 Galley	17.88854
Temp_06_Deck_PD	Ship_01_Deck_PD	2	Spare Ladder1	8.944272



Tabel 5.3. Hasil *matching* antara data layout Template_03_Deck_BD dengan Ship_03_Deck_BD, Ship_02_Deck_BD, Ship_01_Deck_BD, Ship_06_Deck_BD, Ship_05_Deck_BD dan Ship_04_Deck_BD.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Numerical Score
Template_03_Deck_BD	Ship_03_Deck_BD	5	Steward Room Quarter Master Room Motor Room Ladder1 Battery Room	22.36068
Template_03_Deck_BD	Ship_02_Deck_BD	3	Radio Officer Room Chief Engineering room Ladder	13.41641
Template_03_Deck_BD	Ship_01_Deck_BD	3	Captain Room Chief Engineeringg Room Ladder1	13.41641
Template_03_Deck_BD	Ship_06_Deck_BD	1	Gangway	4.472136
Template_03_Deck_BD	Ship_05_Deck_BD	1	Ladder1	4.472136
Template_03_Deck_BD	Ship_04_Deck_BD	1	Ladder1	4.472136



Tabel 5.4. Hasil *matching* antara data layout Template_06_Deck_BD dengan Ship_06_Deck_BD, Ship_05_Deck_BD, Ship_04_Deck_BD, Ship_03_Deck_BD, Ship_02_Deck_BD dan Ship_01_Deck_BD.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Numerical Score
Template_06_Deck_BD	Ship_06_Deck_BD	5	Second Officer Room Boatswain Room Electrician Room AC Room Engine Casing	22.36068
Template_06_Deck_BD	Ship_05_Deck_BD	2	Second Officer Room Third Officer Room	8.944272
Template_06_Deck_BD	Ship_04_Deck_BD	2	Second Officer Room Third Officer Room	8.944272
Template_06_Deck_BD	Ship_02_Deck_BD	2	Second Officer Room Engine Casing	8.944272
Template_06_Deck_BD	Ship_03_Deck_BD	1	Gangway	4.472136
Template_06_Deck_BD	Ship_01_Deck_BD	1	Gangway	4.472136



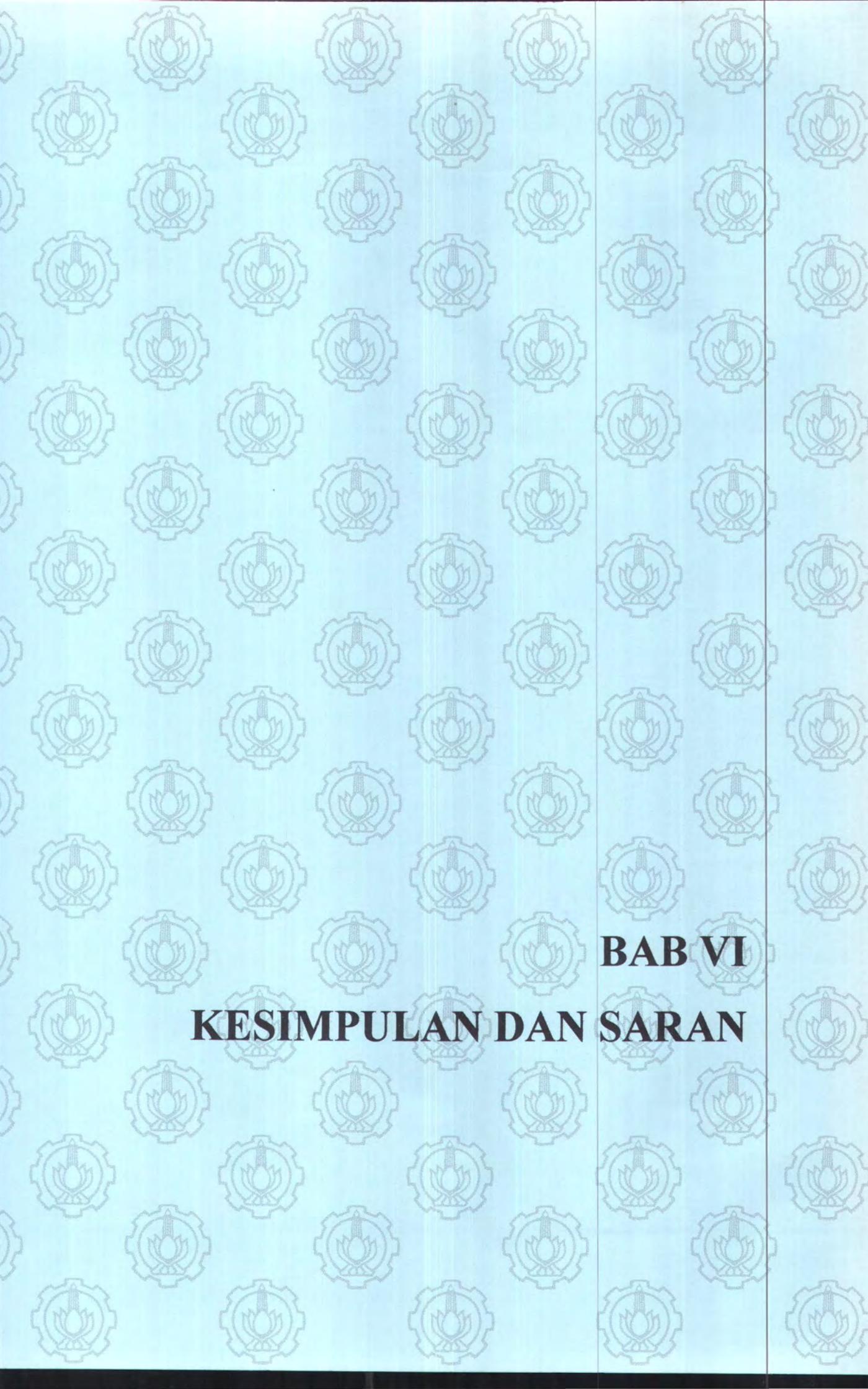
Tabel 5.5. Hasil *matching* antara data layout Template_03_Deck_ND dengan Ship_06_Deck_ND, Ship_05_Deck_ND, Ship_04_Deck_ND, Ship_03_Deck_ND, Ship_02_Deck_ND dan Ship_01_Deck_ND.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Numerical Score
Template_03_Deck_ND	Ship_06_Deck_ND	1	Ladder1	4.472135
Template_03_Deck_ND	Ship_05_Deck_ND	1	Ladder1	4.472135
Template_03_Deck_ND	Ship_04_Deck_ND	1	Ladder1	4.472135
Template_03_Deck_ND	Ship_03_Deck_ND	1	Ladder1	4.472135
Template_03_Deck_ND	Ship_02_Deck_ND	1	Ladder1	4.472135
Template_03_Deck_ND	Ship_01_Deck_ND	1	Ladder1	4.472135



Tabel 5.6. Hasil *matching* antara data layout Template_03_Deck_ND dengan Ship_06_Deck_ND, Ship_05_Deck_ND, Ship_04_Deck_ND, Ship_03_Deck_ND, Ship_02_Deck_ND dan Ship_01_Deck_ND.

Nama Template Layout	Nama Ship Layout	Jumlah Clique	Ruangan Hasil Matching	Numerical Score
Template_06_Deck_ND	Ship_06_Deck_ND	1	Ladder1	4.472135
Template_06_Deck_ND	Ship_05_Deck_ND	1	Ladder1	4.472135
Template_06_Deck_ND	Ship_04_Deck_ND	1	Ladder1	4.472135
Template_06_Deck_ND	Ship_03_Deck_ND	1	Ladder1	4.472135
Template_06_Deck_ND	Ship_02_Deck_ND	1	Ladder1	4.472135
Template_06_Deck_ND	Ship_01_Deck_ND	1	Ladder1	4.472135



BAB VI
KESIMPULAN DAN SARAN



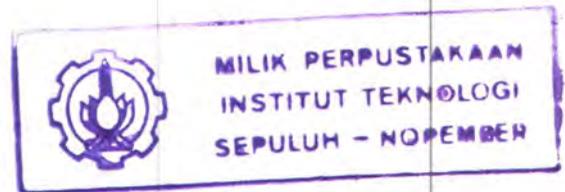
BAB VI

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Dari hasil perencanaan tataletak ruang akomodasi dengan LEVINE System dapat ditarik kesimpulan sebagai berikut:

1. Pemanfaatan desain lama (*design reuse*) untuk membuat desain baru dapat membawa keuntungan bagi *desainer* dan *owner* (pemesan) karena desainer tidak perlu merencanakan desain dari awal (*from scratch*). Sehingga desain baru dapat dihasilkan dalam waktu yang lebih cepat dan lebih mendekati permintaan *owner*.
2. Hasil *matching* yang didapat dari LEVINE System mempunyai kecepatan yang tinggi karena menggunakan komputer sebagai *tools*. Dengan demikian *desainer* dapat mendapatkan hasil desain yang sesuai dengan keinginan *owner* dengan cepat.



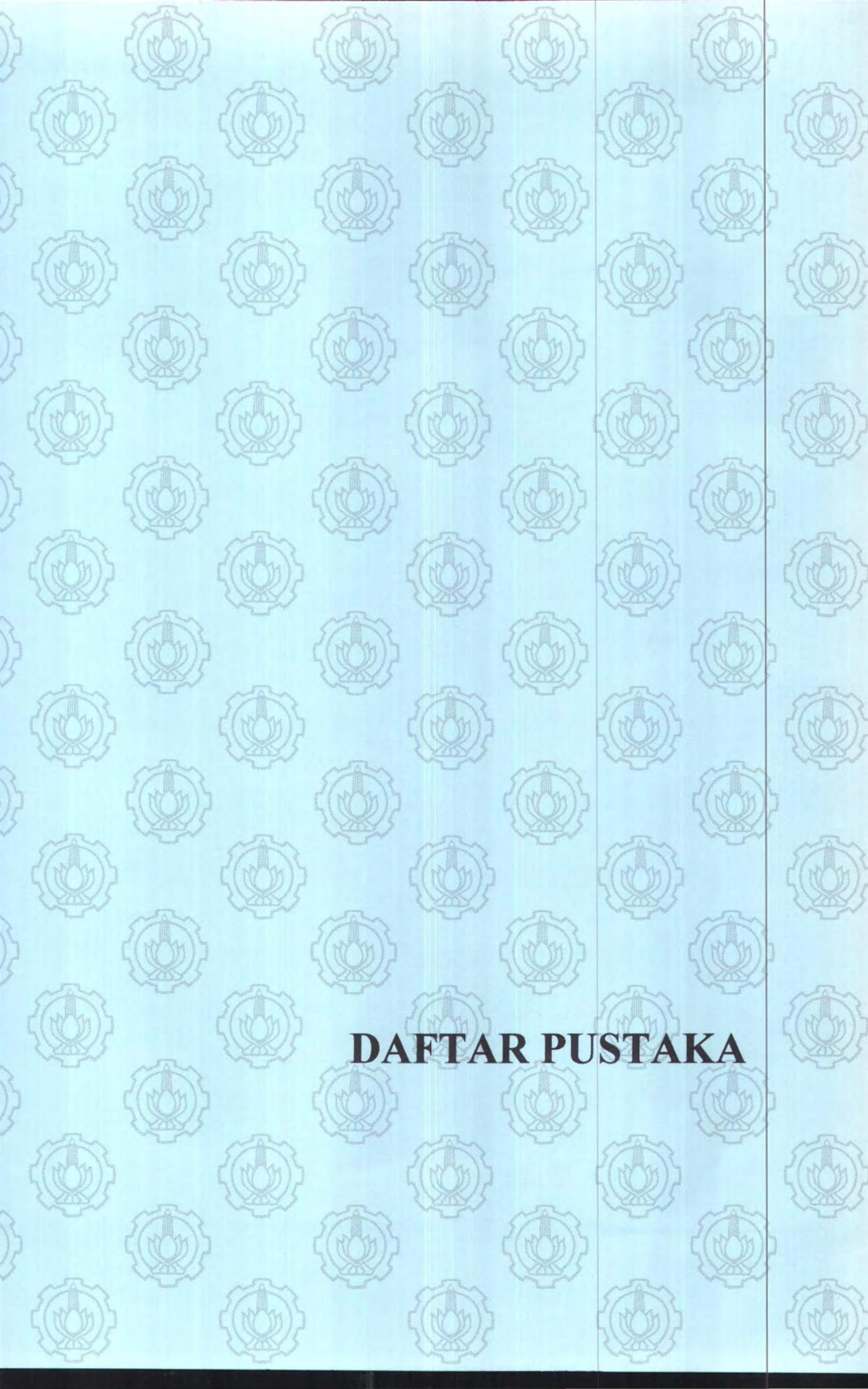
6.2. Saran

Untuk memaksimalkan LEVINE System perlu diperhatikan jumlah data yang digunakan. Semakin banyak data yang digunakan akan menghasilkan pilihan desain yang lebih variatif dan lebih mendekati keinginan *owner*.

Sistem ini juga bisa dikembangkan untuk membantu merancang layout selain ruang akomodasi ABK kapal penumpang, dengan cara



penambahan *option* jenis-jenis layout berdasarkan tipe kapal dan data layout kapal yang dimaksud dalam *case base*.



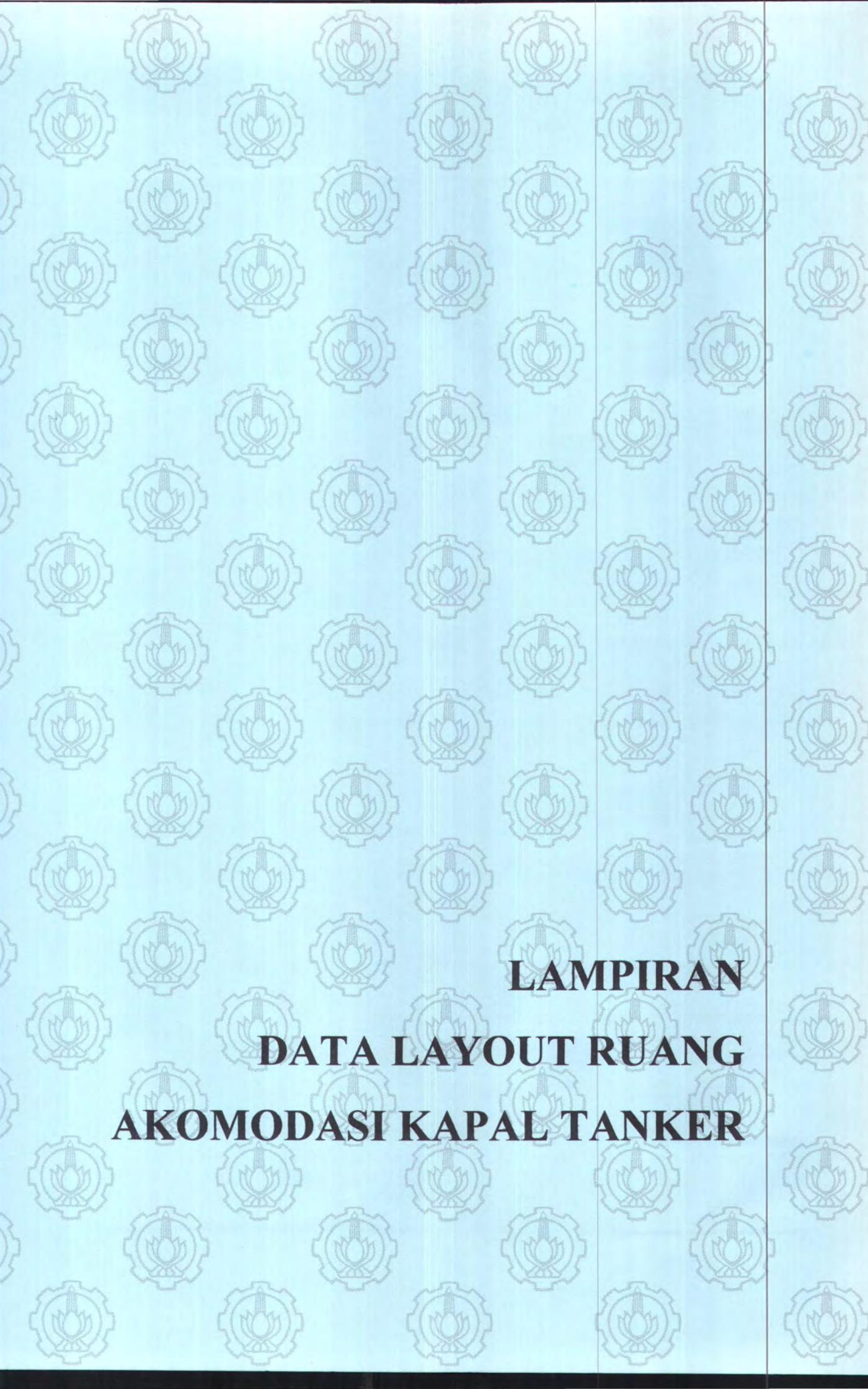
DAFTAR PUSTAKA

DAFTAR PUSTAKA

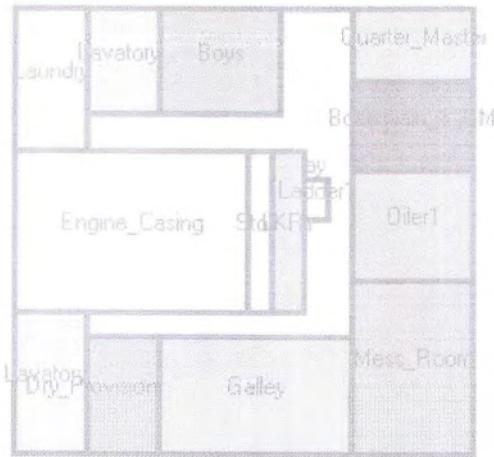
- Aamodt, A. and Plaza, E. (1994). *Case Base Reasoning: Foundation Issues, Methodological and System Approach*. AICom – Artificial Intelligent Communication Press vol. 7:4, 39-59.
- Akin, O. (1978). How do architects design? In J.L. Latombe (Eds.), *Artificial Intelligent and Pattern Recognition in Computer Aided Design*, pp. 65-103. Nort-Holland.
- Apple, James M. (1990). **Tata Letak Pabrik dan Pemindahan Bahan**, Edisi Ketiga, Penerbit ITB Bandung.
- Ballard, D. H. and C. M. Brown. (1982). *Computer Vision*. Englewood Cliffs, New Jersey: Prentice-Hall, Inc.
- Dave, B., G. Schmitt, B. Faltings, and I. Smith. (1994). Case Based Design in Architecture. In J. S. Gero and F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, The Netherlands, pp. 145-162. Kluwer Academic Publishers.
- Omeshek, E. and J. Kolodner. (1992). A Case-Based Design Aids for Architecture. In J. S. Gero and F. Sudweeks (Eds.), *Artificial Intelligence in Design '94*, Dordrecht, pp. 497-516. Kluwer Academic Publishers.
- Faltings, B. (1997). **Case Reuse by Model-Based Interpretation**. In M. Maher and P. Pu (Eds.), *Issues and Applications of Case-Based Reasoning in Design*. Hillsdale, N.J.: Lawrence Erlbaum Associates.

- Flemming, U., Z. Aygen, R. Coyne, and J. Snyder. (1997). **Case-Based Design in a Software Environment that Support the Early Phases**. In M. L. Maher and P. Pu (Eds.), *Issues and Applications of Case-Based Reasoning in Design*. Hillsdale, N.J.: Lawrence Erlbaum Associates.
- Giretti, A., Spalazzi, L. and Lemma, M. (1994). A. S. A. *An Interactive Assistant to Architectural Design*. **Artificial Intelligence in Design**, 93-108.
- Hua, K., B. Faltings, and I. Smith. (1996). CADRE: *Case-Based Geometric Design*. *Artificial Intelligence in Engineering* 10(2), 171-183.
- Hall, G. and Matias, A. (1993). **Rotation, Scale and Translation Invariant Template Matching on a Transputer Network**. *Microprocessors and Microsystems* 17(6), 333-340.
- Jain, A. K. and Dubes, R. C. (1988). *Algorithms for Clustering Data*. New Jersey: Prentice Hall.
- Kolodner, J. L. (1993). *Case-Based Reasoning*. New York: Morgan Kaufmann.
- Maher, M. L., M. B. Balachandran, and D. M. Zhang. (1995). *Case-Based Reasoning in Design*. Mahwah, New Jersey: Lawrence Erlbaum Associates, Publisher.
- Manfaat, D. (1998). **Computer-Based Approach to Effective Utilisation of Spatial Layout Design Experience**, Thesis for the Degree of Doctor, University of Strathclyde, Glasgow, Scotland, UK.
- Manfaat, D., Duffy, A. H. B. and Lee, B. S. (1996). *Review of Pattern Matching Approaches*. *The Knowledge Engineering Review* 11(2), 161-189.

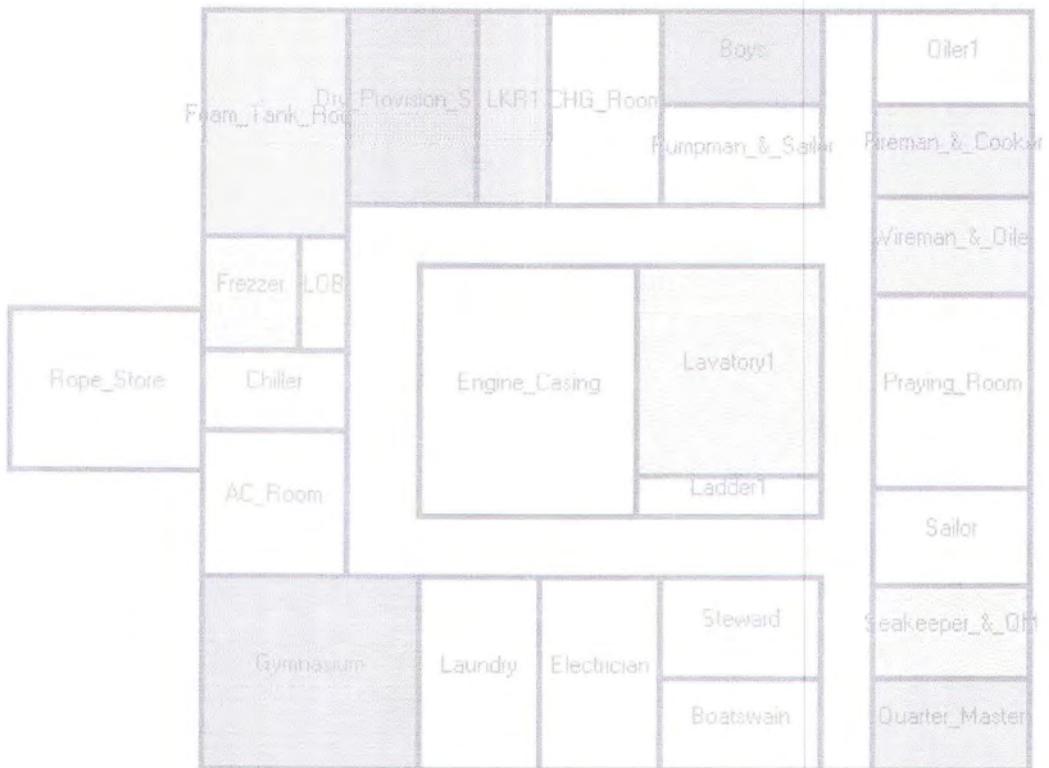
- Manfaat, D., Duffy, A. H. B. and Lee, B. S. (1998). *Topological Pattern and Geometric Shape Matching for Layout Case Retrieval*. Artificial Intelligent in Design, 327-344.
- Manfaat, D., Buana, I. G. N. S. dan Rohman, H. (2002). Penggunaan Teknik *Numerical and Symbolic Pattern Matching* Untuk Case Retrieval Desain Struktur Kapal. Prosiding Seminar Nasional Teori dan Aplikasi Teknologi Kelautan 2002, I-69 – I-83.
- Nugraha, K. A. A. (2001). **Pemilihan Desain Ruang Akomodasi Kapal Tanker Kelas 10000 DWT dengan Teknik Topological Pattern (Graph) Matching**. Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan – ITS.
- Pangastuti, R. (2001). **Penerapan Teknik Spatial Relation Ship Pattern Matching dalam Pemanfaatan Desain Tataletak Ruang Akomodasi Anak Buah Kapal Penumpang 1000 Orang**. Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan – ITS.
- Purnatha, I G. (2000). **Analisis Pengambilan Keputusan Tataletak Ruang Akomodasi KM Caraka Jaya Niaga III**, Jurusan Teknik Perkapalan, Fakultas Teknologi Kelautan – ITS.
- Pusat Pembinaan dan Pengembangan Bahasa. (1990). **Kamus Besar Bahasa Indonesia**. Departemen Pendidikan dan Kebudayaan, Balai Pustaka, Jakarta.
- Sussman, Gerald J., Abelson, H. and Perlis, Alan J. (1996). *Structure and Interpretation of Computer Programs*, Second Edition, The MIT Press, Cambridge, Massachusetts.



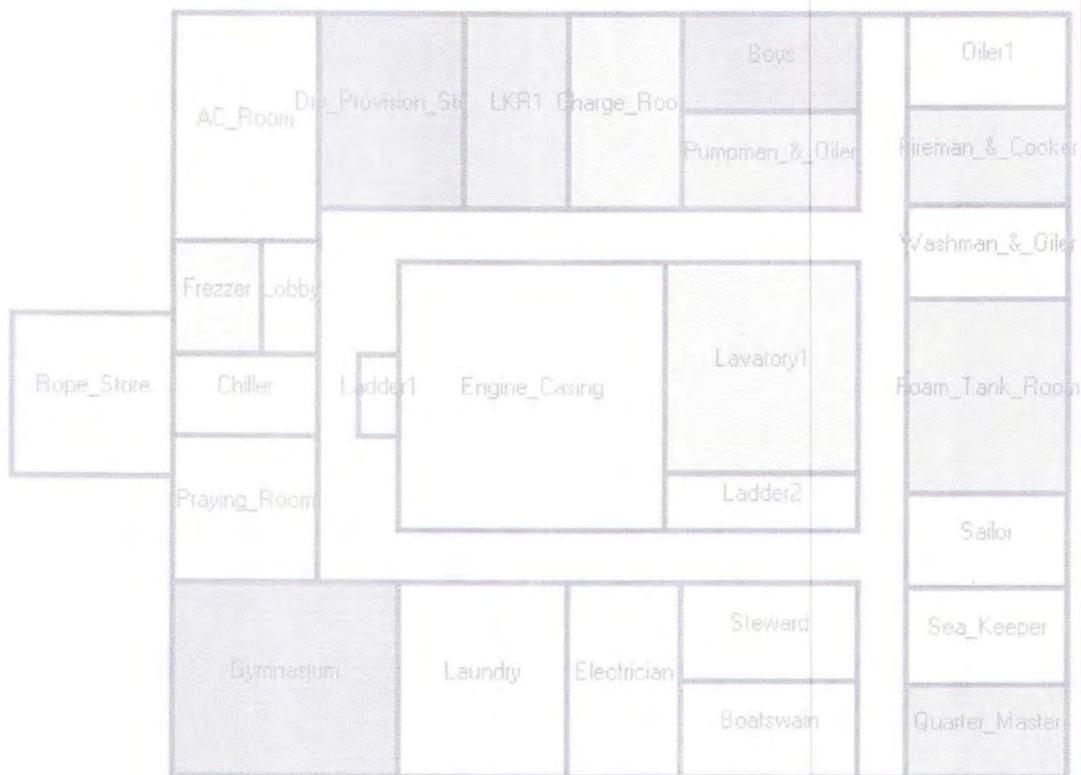
LAMPIRAN
DATA LAYOUT RUANG
AKOMODASI KAPAL TANKER



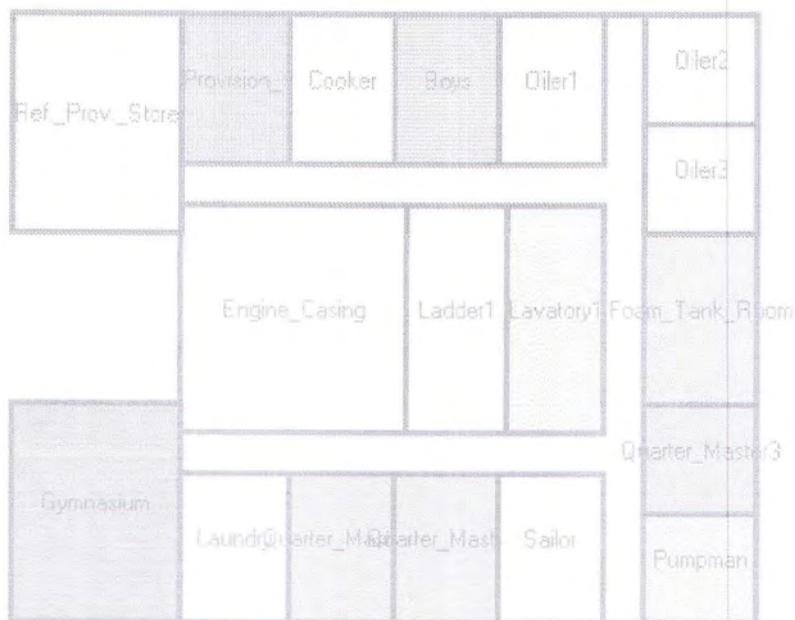
Ship_01_Deck_UD



Ship_04_Deck_UD



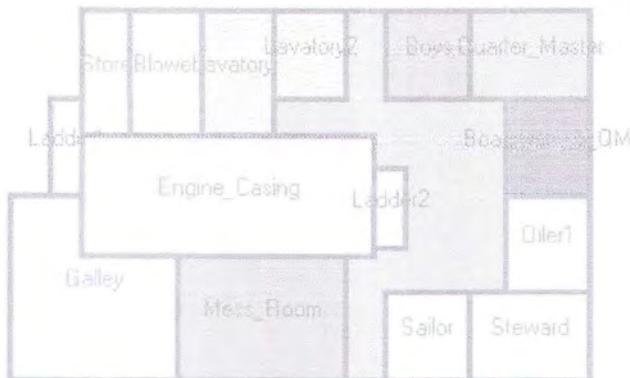
Ship_05_Deck_UD



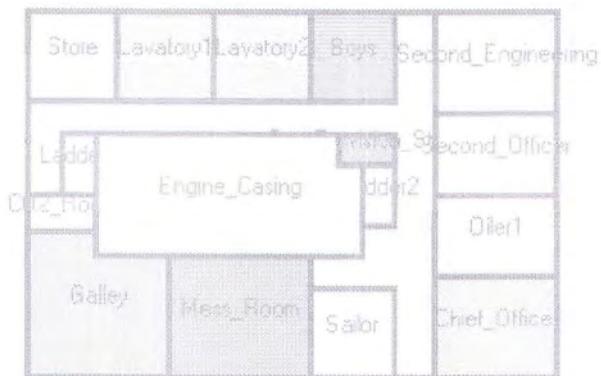
Ship_06_Deck_UD



Ship_01_Deck_PD



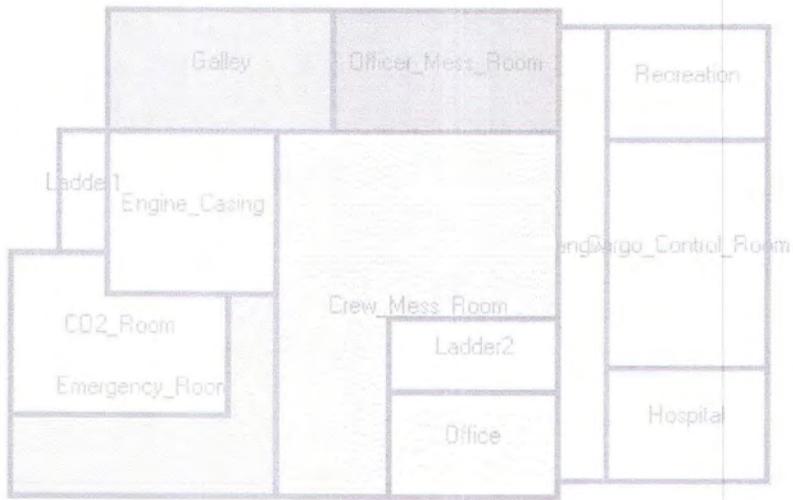
Ship_02_Deck_PD



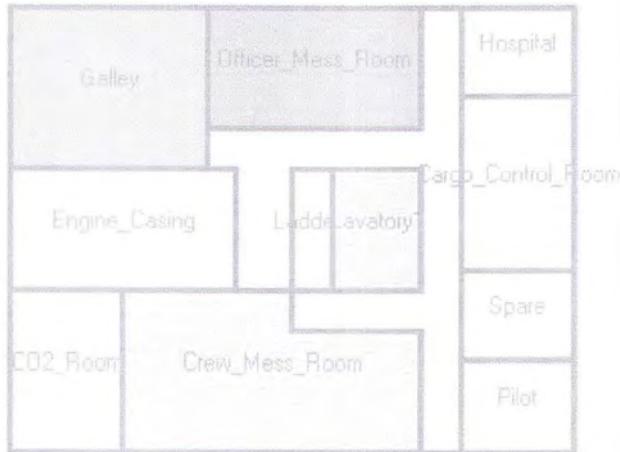
Ship_03_Deck_PD



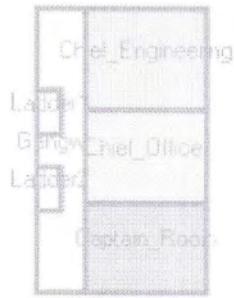
Ship_04_Deck_PD



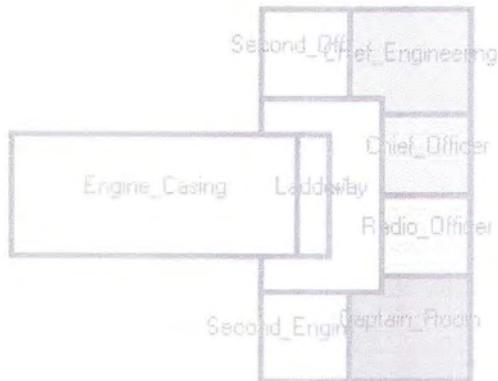
Ship_05_Deck_PD



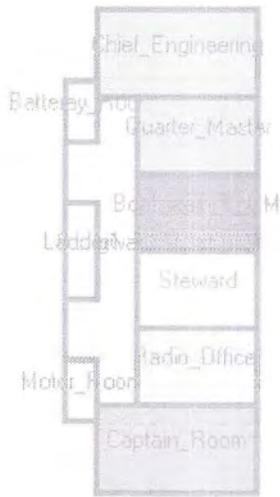
Ship_06_Deck_PD



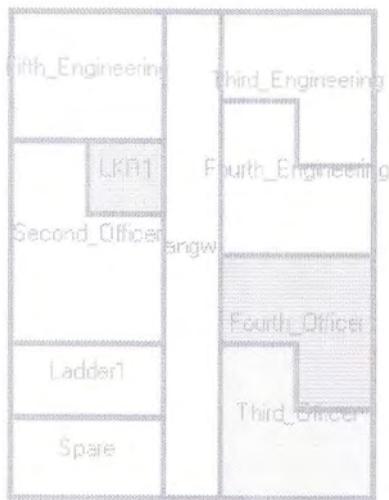
Ship_01_Deck_BD



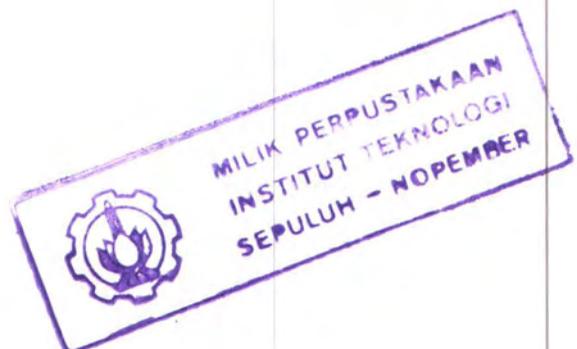
Ship_02_Deck_BD

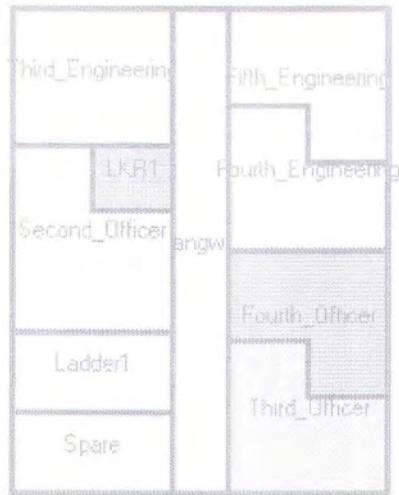


Ship_03_Deck_BD

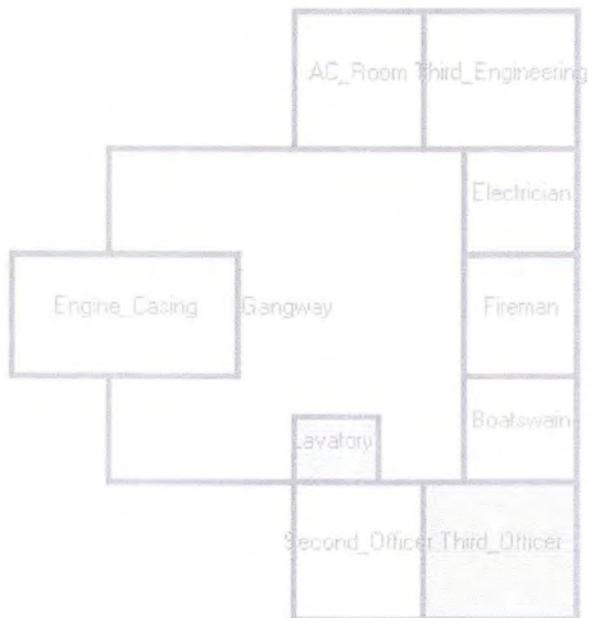


Ship_04_Deck_BD

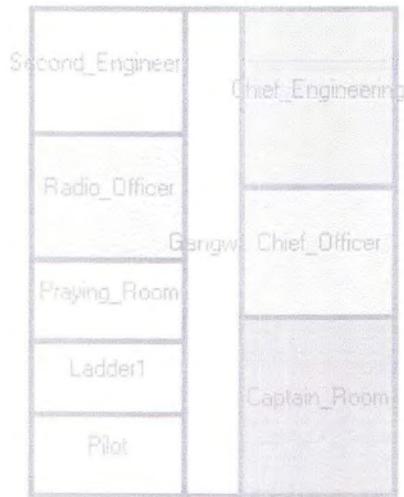




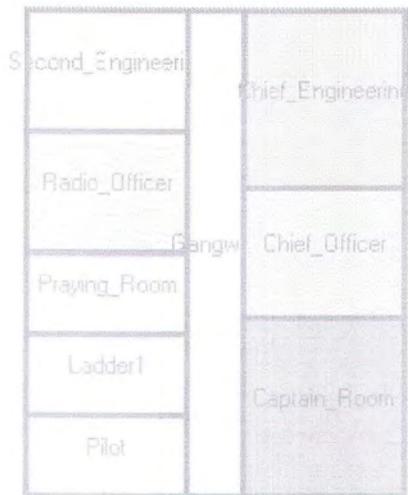
Ship_05_Deck_BD



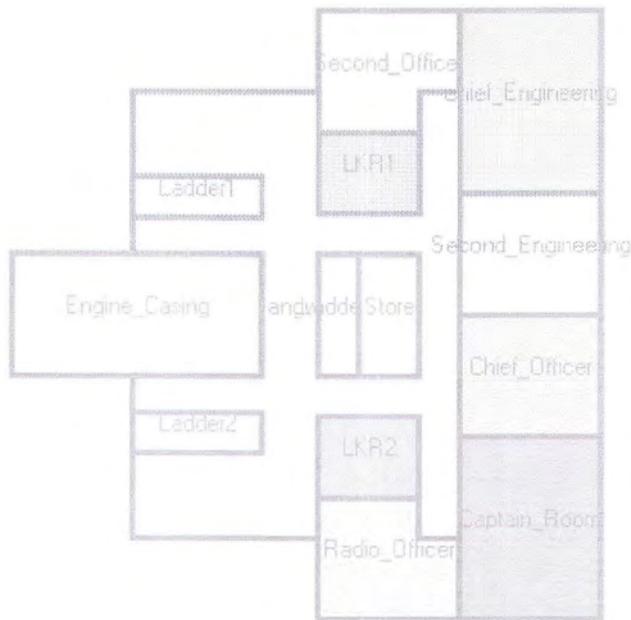
Ship_06_Deck_BD



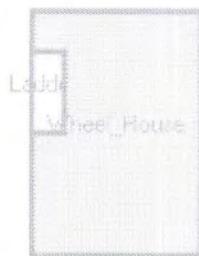
Ship_04_Deck_RD



Ship_05_Deck_RD



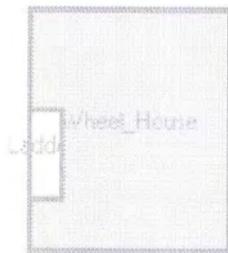
Ship_06_Deck_RD



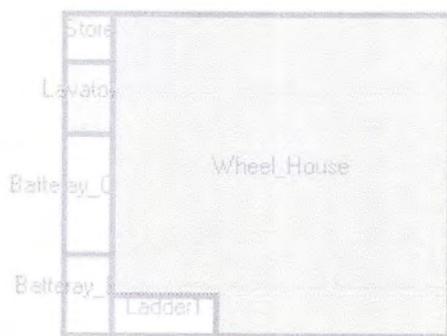
Ship_01_Deck_ND



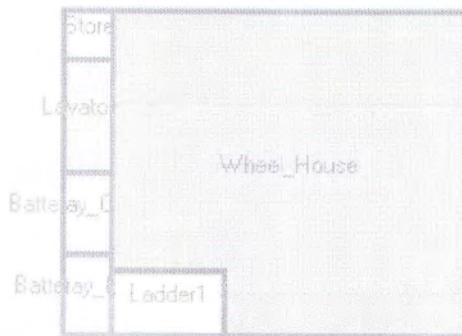
Ship_02_Deck_ND



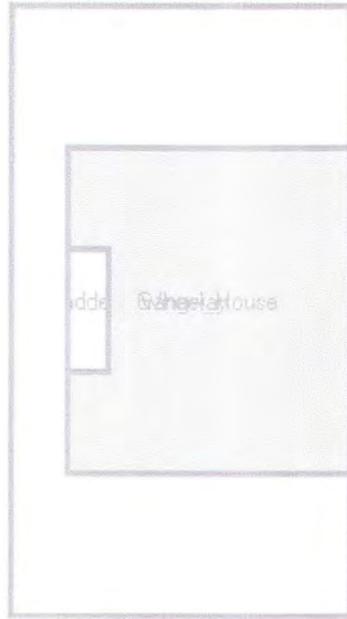
Ship_03_Deck_ND



Ship_04_Deck_ND



Ship_05_Deck_ND





DEPARTEMEN PENDIDIKAN NASIONAL
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
FAKULTAS TEKNOLOGI KELAUTAN
JURUSAN TEKNIK PERKAPALAN

Kampus ITS -Sukoilo, Surabaya 60111 Telp. 5947254, 5994251-5 Pes. 1173 - 1176 5947254

SURAT KEPUTUSAN TUGAS AKHIR

No. : 51 a / K03.4.2/PP/2002

Nama Mahasiswa : Wahyu Wiyati
Nomor Pokok : 4197100066
Tanggal diberi tugas : 12 Pebruari 2002
Tanggal selesai tugas : 28 Juni 2002
Dosen Pembimbing : 1. I Gusti Ngurah Sumanta Buana, ST, M.Eng
2. Ir. Djauhar Manfaat, MSc., Ph.D.

Uraian / judul tugas akhir yang diberikan :

*#ANALISIS PEMILIHAN LAYOUT RUANG AKOMODASI MENGGUNAKAN
TEKNIK TOPOLOGICAL AND NUMERICAL PATTERN MATCHING PADA
KAPAL TANKER#*

Surabaya, 12 Pebruari 2002

Jurusan Teknik Perkapalan

Ketua,

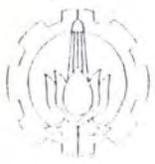


Ir. Djauhar Manfaat, MSc., Ph.D.

NIP. 131 651 444.

Tembusan :

1. Yth. Dekan FTK-ITS
2. Yth. Dosen Pembimbing
3. Arsip



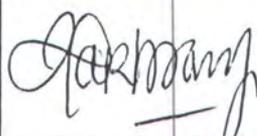
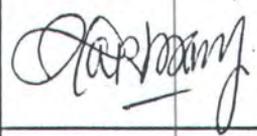
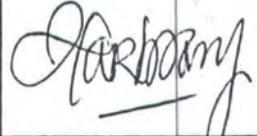
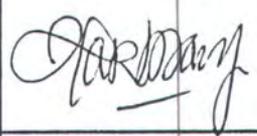
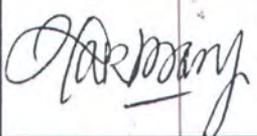
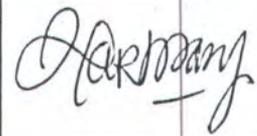
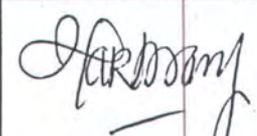
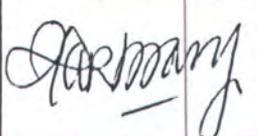
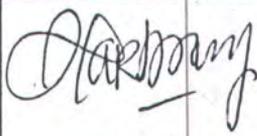
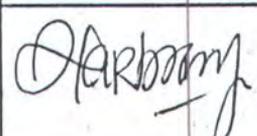
FAKULTAS TEKNOLOGI KELAUTAN ITS
JURUSAN TEKNIK PERKAPALAN

DAFTAR KEMAJUAN TUGAS AKHIR (NA 1701)

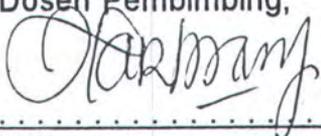
nama mahasiswa : Wahyu Wiyati
N.P. : 4197100066
Tugas diberikan : Semester Genap 2001 . . / 2002 .
Tanggal mulai tugas : 12 Februari 2002
Tanggal selesai tugas : 28 Juni 2002
Pembimbing : 1. Ir. Gusti. Ngurah Sumanta Buana, S.Tp. M.Eng.
2. Ir. Djauhar Manfaat, MSc, Ph.D.

Tanggal	Uraian Kemajuan Tugas	Tanda Tangan
5/02	Melengkapi data RU kapal Tanker	
13/02	Bab I. Pendahuluan	
10/02	Revisi Bab I + Bab II. Dasar Teori	
10/02	Revisi Bab II. lanjutan	
16/02	Sebagian Bab III. Teknik Implementasi	
13/02	Pembuatan Program (masukkan data)	
14/02	Demo program	
11/02	Demo program	

lihat halaman berikutnya

Tanggal	Uraian Kemajuan Tugas	Tanda Tangan
2/11/02	Bab IV dan demo program	
10/11/02	Bab IV	
17/11/02	Revisi bab IV & diskusi	
1/12/02	Bab IV selesai	
1/12/02	Distribusi & demo program	
6/12/02	Revisi program	
13/12/02	Demo program	
20/12/02	Pembuatan interface	
27/1/03	Diskusi & bab V.	
3/1/03	Demo program	
20/1/03	Pengecekan laporan bab V dan VI	

Catatan :
 Formulir ini harus dibawa pada saat konsultasi
 Konsultasi dilaksanakan minimal seminggu
 sekali.
 Formulir ini harus dikumpulkan kembali pada
 saat mengumpulkan laporan tugas akhir.

Surabaya, 23 - 1 - 2002 ..
 Dosen Pembimbing,

 NIP.

