

**PENERAPAN GRAFIK DATABASE UNTUK STANDARISASI  
FAIRING WORK PENYIMPANGAN BENTUK  
KOMPONEN KAPAL**

**TUGAS AKHIR**



RSPe  
623.820 0285  
Tri  
P-1  
1999

PERPUSTAKAAN ITS	
Tgl. Terima	28-6-2000
Terima oleh	H
No. Agenda	21-417

Oleh :

**TRI YULIANTO**  
NRP. 4193 100 006

**TEKNIK PERKAPALAN  
FAKULTAS TEKNOLOGI KELAUTAN  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
1999**



## LEMBAR PENGESAHAN

Tugas Akhir  
NA 1701

### **PENERAPAN GRAFIK DATABASE UNTUK STANDARISASI FAIRING WORK PENYIMPANGAN BENTUK KOMPONEN KAPAL**

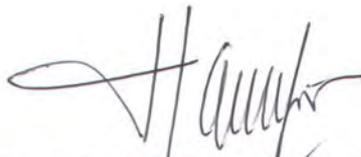
Disusun oleh :

**TRI YULIANTO**  
**NRP : 4193 100 006**

Telah diperiksa dan dinyatakan siap untuk diujikan  
pada tanggal Juli 1999

Surabaya, Juli 1999

Dosen Pembimbing



**Ir. R. SJARIEF WIDJAJA, Ph.D.**  
**NIP : 131782034**



**LEMBAR PENGESAHAN REVISI**

**Tugas Akhir  
NA 1701**

**PENERAPAN GRAFIK DATABASE UNTUK  
STANDARISASI FAIRING WORK PENYIMPANGAN  
BENTUK KOMPONEN KAPAL**

Disusun oleh :

**TRI YULIANTO**  
**NRP : 4193 100 006**

Telah direvisi dan diperiksa berdasarkan point verbal yang  
telah diujikan pada ujian Tugas Akhir  
pada tanggal 3 Agustus 1999

Surabaya, Agustus 1999

Dosen Pembimbing,



**Ir. R. SJARIEF WIDJAJA, Ph.D.**  
**NIP: 131782034**

**LEMBAR PENGESAHAN**

**PENERAPAN GRAFIK DATABASE UNTUK  
STANDARISASI FAIRING WORK PENYIMPANGAN  
BENTUK KOMPONEN KAPAL**

**TUGAS AKHIR (NA 1701)**

**Diajukan sebagai salah satu persyaratan  
untuk memperoleh gelar Sarjana Teknik pada  
Jurusan Teknik Perkapalan Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Mengetahui,  
Ketua Jurusan Teknik Perkapalan,**

**Ir. KOESTOWO S.W  
NIP: 130 687 430**

## ABSTRAK

Dalam proses produksi kapal seringkali ditemui adanya penyimpangan bentuk yang disebabkan oleh deformasi pada material pelat. Penyimpangan yang disebabkan oleh deformasi ini banyak dijumpai pada tahap *assembly*. Untuk melakukan pengawasan terhadap proses produksi sekaligus perbaikannya dapat dilakukan dengan bantuan komputer. Dengan *me-matching*-kan data desain dan data aktual kapal, maka dapat diketahui bahwa penyimpangan pada komponen tersebut memenuhi batas toleransi atau tidak. Jika penyimpangan melebihi batas toleransi maka harus dilakukan perbaikan. Dengan memilih database penyimpangan yang telah disediakan maka perbaikan komponen tersebut akan diketahui. Jenis perbaikannya dengan menggunakan cara pemanasan atau *fairing* dimana titik-titik yang akan dilakukan pemanasan dapat ditampilkan. Standar kualitas yang digunakan dalam pengawasan lambung kapal menggunakan Japan Shipbuilding Quality Standart (JSQS) yang selama ini digunakan oleh PT PAL INDONESIA.

## KATA PENGANTAR

Dengan mengucapkan puji syukur Alhamdulillah kehadiran Allah SWT atas segala rahmat dan hidayah-Nya akhirnya kami dapat menyelesaikan laporan tugas akhir ini dengan judul:

### **Penerapan grafik database untuk standarisasi fairing work penyimpangan bentuk komponen kapal**

Laporan tugas akhir ini disusun guna memenuhi persyaratan-persyaratan dalam menempuh studi tahap Sarjana Teknik Perkapalan. Didalam penyelesaian tugas akhir ini tidak lupa saya sampaikan banyak terima kasih kepada pihak-pihak yang telah membantu penulis menyelesaikan tugas akhir ini. Untuk itu pada kesempatan ini kami ingin mengucapkan terimakasih yang sebesar-besarnya kepada :

1. Bapak Ir Syarief Widjaya PhD, selaku dosen pembimbing
2. Bapak Ir P Eko Panunggal PhD, selaku dosen wali
3. Bapak Ir Koestowo S, selaku Ketua Jurusan Teknik Perkapalan
4. Bapak Ir Andjar Soeharto, selaku Sekretaris Jurusan Teknik Perkapalan
5. Bapak dan Ibu di Yogya yang telah dengan penuh kasih sayang memberi bimbingan serta doanya
6. Mas Budi sekeluarga, Mas Mugi dan adik-adikku Jito, Kuswanto, Hendi serta Simbah Putri yang senantiasa memberi dorongan, semangat dan doanya
7. Ririen yang dengan penuh kasih sayang dan kesetiaan memberikan semangat perjuangan hidup

8. Joddy, Lukman, Fasih yang telah banyak membantu dalam menyelesaikan tugas akhir ini
9. Rekan-rekan Djie Sam Soe yang ikut merasakan suka dan duka bersama penulis
10. Rekan-rekan Kapal '93 seperjuangan.

Akhirnya dengan segala kerendahan hati, kami mohon maaf atas segala kekurangan dan kesalahan dalam penyajian laporan tugas akhir ini. Semoga laporan tugas akhir ini dapat bermanfaat bagi kami khususnya dan pembaca pada umumnya.

Surabaya, 20 Juli 1999

## DAFTAR ISI

	halaman
Halaman Judul . . . . .	i
Lembar Pengesahan . . . . .	ii
Lembar Pengesahan Revisi . . . . .	iii
Abstrak . . . . .	iv
Kata Pengantar . . . . .	v
Daftar Isi . . . . .	vii
Daftar Gambar . . . . .	x
Daftar Tabel . . . . .	xii
BAB I. PENDAHULUAN . . . . .	1
1.1 Latar Belakang . . . . .	1
1.2 Tujuan Penulisan . . . . .	2
1.3 Batasan Masalah . . . . .	3
1.4 Metodologi . . . . .	3
BAB II. TEKNOLOGI PEMBANGUNAN KAPAL . . . . .	5
II.1 Perkembangan Teknologi Pembangunan Kapal . . . . .	5
II.2 Faktor-Faktor Yang Mempengaruhi Proses Produksi . . . . .	9
II.3 Tahapan Proses Produksi . . . . .	12
II.3.1 Tahap Persiapan Produksi . . . . .	13
II.3.2 Tahap Fabrikasi . . . . .	13
II.3.3 Tahap Sub Assembly . . . . .	18

II.3.5 Tahap Erection .....	20
II.4 Pengawasan Proses Produksi Kapal .....	22
BAB III. PEMERIKSAAN PENYIMPANGAN BENTUK PADA PROSES PRODUKSI KAPAL .....	29
III.1 Tahapan Pemeriksaan Pada Proses Produksi Kapal .....	29
III.1.1 Pemeriksaan Pada Tahap Fabrikasi .....	30
III.2.2 Pemeriksaan Pada Tahap Sub Assembly .....	32
III.2.3 Pemeriksaan Pada Tahap Assembly .....	33
III.2.4 Pemeriksaan Pada Tahap Erection .....	35
III.2 Jenis-Jenis Penyimpangan Karena Bentuk Deformasi .....	36
III.2.1 Penyimpangan Bentuk Karena Deformasi Penyusutan .....	36
III.2.2 Penyimpangan Bentuk Karena Deformasi Sudut .....	40
III.2.3 Penyimpangan Bentuk Karena Deformasi Memanjang .....	46
III.2.4 Penyimpangan Bentuk Karena Deformasi Flexure dan Shrinkage .....	47
III.2.5 Penyimpangan Bentuk Karena Deformasi Buckling .....	48
III.3. Jenis Perbaikan Penyimpangan Bentuk Deformasi .....	50
III.3.1 Penanggulangan Penyimpangan Bentuk Deformasi .....	50
III.3.2 Perbaikan Penyimpangan Bentuk Deformasi .....	52
III.3.2.1 Perbaikan Deformasi Dengan Cara Mekanik .....	52
III.3.2.1 Perbaikan Deformasi Dengan Cara Pemanasan .....	53
III.4 Perbaikan Deformasi Pada Konstruksi Lambung Kapal .....	61
BAB IV. PENYAJIAN KOMPUTER GRAFIS KOMPONEN KAPAL .....	69

IV.1 Penyajian Grafis 3-D Komponen Kapal .....	69
IV.2 Sistem Koordinat Penyajian Grafis 3-D .....	70
IV.3 Pencarian Besarnya Penyimpangan Bentuk .....	72
BAB V. PENERAPAN DATABASE MANAGEMENT SYSTEM DALAM PENGOLAHAN DATA .....	76
V.1 Konsep Database Management System .....	76
V.2 Peranan Program Database .....	77
V.3 Data Abstraction ( Abstraksi Data) .....	80
V.4 Paket Bahasa Yang Digunakan Dalam Database Management System .....	83
V.5 Perancangan Database .....	85
BAB VI. PROTOTIPE PROGRAM STANDARISASI FAIRING WORK PENYIMPANGAN BENTUK KOMPONEN KAPAL .....	88
VI.1 Pemilihan Bahasa Pemrograman .....	88
VI.2 Flowchart Program .....	90
VI.3 Struktur Program .....	92
VI.4 Running Program .....	96
BAB VII. DISKUSI DAN REKOMENDASI .....	101
BAB VIII. KESIMPULAN .....	104
VIII.1 Kesimpulan. ....	104
VIII.2 Saran. ....	105
DAFTAR PUSTAKA	
LAMPIRAN	

## DAFTAR GAMBAR

Gambar 2.1	Komponen Product-Oriented Work Breakdown Structure. ....	7
Gambar 2.2	Proses Produksi Bengkel Fabrikasi .....	15
Gambar 3.1	Penyusutan Memanjang dan Melintang. ....	37
Gambar 3.2	Deformasi Sudut Sambungan Butt Joint dan Fillet. ....	41
Gambar 3.3	Grafik Hubungan Antara Perubahan Sudut Dengan Tebal Pelat .....	42
Gambar 3.4	Grafik Hubungan Antara Perubahan Sudut Dengan Kondisi Pengelasan .....	43
Gambar 3.5	Grafik Hubungan Antara Prosedur Pengelasan dan Perubahan Sudut. ....	45
Gambar 3.6	Deformasi Memanjang Karena Pengelasan .....	47
Gambar 3.7	Deformasi Felxure dan Deformasi Shrinkage .....	48
Gambar 3.8	Pembebanan Pada Proses Pengelasan .....	49
Gambar 3.9	Deformasi Buckling .....	50
Gambar 3.10	Line Heating Memanjang .....	54
Gambar 3.11	Line Heating Pendek-Pendek .....	54
Gambar 3.12	Pine Needle Heating .....	54
Gambar 3.13	Lattice Heating .....	55
Gambar 3.14	Ring Heating .....	55
Gambar 3.15	Model Spot Heating .....	56
Gambar 3.16	Grafik Hubungan Spot Heating Dengan Line Heating .....	57

Gambar 3.17	Jenis-Jenis Pemanasan Segitiga .....	57
Gambar 3.18	Batas Suhu Yang Dipakai Untuk Line Heating .....	58
Gambar 3.19	Jarak Nyala Api Brander Dengan Pelat .....	59
Gambar 3.20	Perbaikan Deformasi Pelat Antara 2 Gading Dengan Line Heating .....	61
Gambar 3.21	Perbaikan Deformasi Pelat Antara 2 Gading Dengan Line Heating .....	62
Gambar 3.22	Pemanasan Setempat (Spot Heating) .....	63
Gambar 3.23	Pemanasan Pada Sambungan Las Tumpul .....	64
Gambar 3.24	Pemanasan Pada Penguat Dalam .....	64
Gambar 3.25	Pemanasan Pada Bracket .....	65
Gambar 3.26	Pemanasan Deformasi Pada Face Plate Yang Naik .....	66
Gambar 3.27	Pemanasan Deformasi Pada Face Plate Yang Turun .....	66
Gambar 3.28	Urut-Urutan Pemanasan Pada Deck dan Wall Kedap .....	67
Gambar 3.29	Urut-Urutan Pemanasan .....	67
Gambar 3.30	Urut - Urutan Pemanasan Pada Deformasi Pelat Yang Berlubang .....	68
Gambar 4.1	Arah M dan N Pada Pelat Kulit .....	69
Gambar 4.2	Koordinat Cartesian Dalam Penggambaran Bentuk Komponen. .	71
Gambar 4.3	Langkah Pencarian Besar Penyimpangan .....	73
Gambar 5.1	Architecture DBMS .....	82

## DAFTAR TABEL

Tabel 3.1	Tahap Pemeriksaan .....	31
Tabel 3.2	Tahap Pemeriksaan Pada Fabrikasi .....	32
Tabel 3.3	Hubungan Tebal Pelat Dengan Jarak Nyala Api .....	59
Tabel 3.4	Hubungan Kecepatan Nozzle dan No Brander Serta Tebal Pelat. ....	59

# BAB I

## PENDAHULUAN

### I.1. LATAR BELAKANG

Dengan didukung oleh perkembangan ilmu pengetahuan dan teknologi terutama dibidang infrastruktur, teknologi pembangunan kapal mengalami perkembangan yang sangat pesat dari tahun ke tahun. Setelah Perang Dunia II dengan disponsori oleh *IHI-Shipyard*, Japan teknologi pembangunan kapal telah mengalami perubahan dengan mengembangkan metode *Integrated Hull Construction and Painting* (IOHP). Metode ini merupakan metode pembangunan kapal yang paling mutakhir dimana proses pembuatan badan kapal sudah dintegrasikan secara maksimal dengan pekerjaan *outfitting* dan pengecatan pada setiap *zone/area/stage*. Pada dasarnya metode pembangunan kapal ini adalah mengelompokkan bagian-bagian konstruksi berdasarkan atribut-atribut pembuatan dan rancang bangun. Proses produksi dilaksanakan dengan menyederhanakan proses pembuatan tiap komponen lambung menjadi beberapa operasi yang serupa.

Teknologi pembangunan kapal dengan metode ini harus didukung dengan peningkatan potensi sumber daya galangan secara menyeluruh, seimbang dan terintegrasi, disertai dengan semakin sempurnanya standar-standar kerja dan konsistensi ketepatan proses produksi yang sangat tinggi. *Sistem Accuracy Control/Quality Control* mutlak diperlukan untuk pengendalian ketepatan ukuran pada setiap tahapan produksi agar kesalahan dapat diketahui sedini mungkin, dan dengan demikian dapat mengurangi pekerjaan ulang (*rework*) yang akan

menambah biaya produksi. Pelaksanaan pekerjaan pengawasan/pengendalian mutu bermula dari suatu usaha pencatatan dari setiap pekerjaan yang pernah dilakukan. Catatan ini antara lain meliputi standar unit ukuran yang ditetapkan dan penyimpangan pada ukuran yang diperbolehkan. Untuk meningkatkan efektivitas kerja dari *Accuracy Control/Quality Control* maka diperlukan suatu pengawasan yang didukung sistem komputasi. Ketidaktepatan ukuran dan penyimpangan bentuk komponen kapal akibat dari proses pengerjaan panas ataupun dingin, proses perbaikannya perlu dilakukan standarisasi agar perbaikannya dapat dilakukan dengan cepat dan tepat dengan mutu yang baik. Dengan didukung sistem komputasi pengidentifikasian jenis penyimpangan dapat diketahui dengan cepat dan perbaikannya dapat dilakukan sesuai dengan standar perbaikan.

## 1.2. TUJUAN PENULISAN

Penulisan skripsi ini mempunyai tujuan untuk membuat prototipe program standarisasi *fairing* (perbaikan) penyimpangan bentuk komponen kapal yang ditampilkan secara grafis tiga dimensi dengan memakai sistem koordinat  $x, y, z$ . Prototipe program tersebut mampu mengidentifikasi sebuah komponen kapal yang mengalami penyimpangan bentuk masih dalam batas toleransi atau tidak. Apabila penyimpangannya melebihi batas toleransi maka akan ditampilkan sebuah instruksi perbaikan menggunakan sistem pemanasan (*line heating*). Banyaknya *line* (garis) dan posisi dari *line* untuk proses *line heating* dapat ditampilkan.

### I.3. BATASAN MASALAH

Untuk mencapai tujuan yang diinginkan maka ruang lingkup pembahasan dibatasi dengan :

- Penyimpangan bentuk hanya meliputi macam-macam deformasi.
- Penyimpangan bentuk diasumsikan ke arah satu sumbu koordinat saja.
- Perbaikan penyimpangan dilakukan dengan metode pemanasan.
- Penyimpangan bentuk yang diamati, yang sering terjadi pada konstruksi lambung kapal.
- Standar penyimpangan dan perbaikan berdasarkan *Japan Shipbuilding Quality Standard* (JSQS)

### I.4. METODOLOGI

#### a. Studi Literatur

Mempelajari teori dari buku-buku yang ada relevansinya dengan pembahasan skripsi.

#### b. Studi Lapangan

Melakukan pengamatan di lapangan tentang penyimpangan-penyimpangan bentuk komponen kapal yang sering terjadi pada setiap tahap proses produksi beserta perbaikannya.

#### c. Pembuatan prototipe program

Dalam proses produksi kapal dengan sistem yang sudah terintegrasi dibutuhkan suatu ketepatan ukuran. Penyimpangan-penyimpangan bentuk komponen kapal

pada setiap tahap produksi harus dilakukan pengawasan. Untuk mempermudah pengawasan terhadap penyimpangan bentuk dan perintah perbaikannya maka perlu dilakukan standarisasi.

Di dalam prototipe program ini akan dibuat *database* penyimpangan-penyimpangan bentuk komponen kapal beserta perbaikannya yang mempunyai fungsi untuk melakukan standarisasi perbaikan (*fairing work*).

Input data dari suatu komponen kapal akan ditampilkan dalam bentuk grafis kemudian di-*match*-kan ke *database* penyimpangan bentuk, kemudian akan muncul perintah perbaikan terhadap komponen kapal tersebut apabila komponen kapal tersebut mengalami penyimpangan bentuk. Perintah perbaikan (*fairing work*) ditampilkan secara grafis sehingga akan memudahkan pekerja untuk melakukan perbaikan.

## **BAB II**

### **TEKNOLOGI PEMBANGUNAN KAPAL**

Teknologi pembangunan kapal mengalami perkembangan yang sangat pesat seiring dengan semakin majunya teknologi dibidang infrastuktur dan komputer. Pembangunan kapal dengan sistem terintegrasi peran komputer sangat dibutuhkan untuk mendukung proses desain dan produksi.

Desain dan produksi merupakan integral yang tidak dapat dipisahkan karena kelebihan-kelebihan pada tahap desain berkurang apabila tidak dilanjutkan dengan pemikiran proses produksi dengan penggunaan komputer. Sebagai contoh sistem *CAD* dengan kemampuan *bill of material* bisa mengidentifikasi dengan lebih persis dan menjadwal lebih akurat mengenai material yang dibutuhkan. Dan data-data ukuran pelat yang harus dipotong, disimpan dalam bentuk data-data numerik dapat ditransfer langsung ke *CNC machine* dimana sebagai pengontrol mesin melakukan pemotongan.

#### **II.1. PERKEMBANGAN TEKNOLOGI PEMBANGUNAN KAPAL**

Menurut Chirillo, perkembangan teknologi produksi kapal dibagi menjadi empat tahapan, berdasarkan teknologi yang digunakan pada proses produksinya.

##### □ *Conventional Construction and Outfitting*

Tahapan pertama ini merupakan teknologi produksi kapal, yang berorientasi pada *sistem* atau *fungsi* yang ada pada kapal dan volume pekerjaan

hampir seluruhnya dilaksanakan di *building berth*. Metode ini dimulai peletakan lunas, kemudian pemasangan gading, kulit dan seterusnya sampai bangunan atas dan terakhir pekerjaan *oufitting* (O/F). Pekerjaan O/F dilakukan sistem per sistem, antara lain : pemasangan ventilasi, perpipaan, permesinan, pelistrikan dan lain-lain.

Pembangunan kapal dengan metode ini sering digunakan di galangan-galangan kecil dimana fasilitas galangan yang dimiliki masih sangat sederhana. Metode pembangunan kapal ini merupakan teknologi paling konvensional dan tingkat produktivitasnya sangat rendah, karena semua lingkup pekerjaan memiliki ketergantungan yang tinggi antara yang satu dengan yang lainnya, sehingga membutuhkan waktu yang sangat lama. Selain itu, mutu hasil pekerjaan sangat rendah karena hampir seluruh pekerjaan dilakukan secara manual dan lingkungan pekerjaan yang tidak mendukung (tidak nyaman, posisi kerja yang sulit).

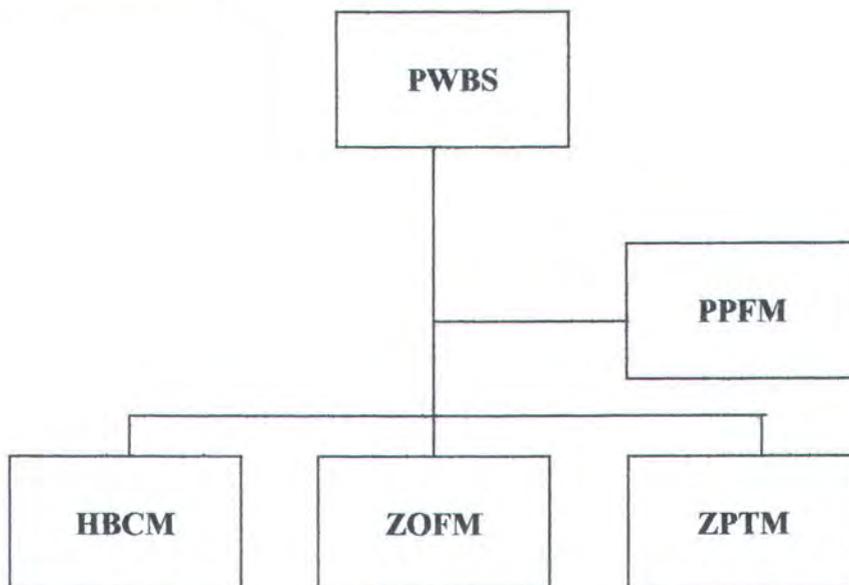
#### □ *Hull Block Construction Method & Pre-Outfitting*

Tahapan ini mulai dikembangkan sejak dikenalkannya teknologi pengelasan pada pembangunan kapal, dimana kapal sudah dibuat dalam bentuk seksi-seksi dan blok-blok, kemudian disambung satu sama lain di *building berth*. Beberapa pekerjaan O/F sudah mulai dilakukan pada blok atau badan kapal yang sudah jadi. Dengan metode ini, *steel troughput* meningkat dan mutu hasil pekerjaan lebih baik, karena volume pekerjaan di *building berth* menjadi berkurang dan pekerjaan pengelasan banyak dilakukan di bengkel dengan kondisi lingkungan kerja yang lebih nyaman. Pekerjaan pengelasan sudah dilakukan

dengan mesin las semi/otomatis dengan posisi *down-hand*. Blok-blok dapat diputar untuk menghindari dari pengelasan dengan posisi *overhead*. Pada tahapan ini, kontrol dimensi dan bentuk blok sudah menjadi penting terutama pada daerah sambungan blok, sehingga peran *Accuracy Control/Quality Control* sudah dibutuhkan untuk melakukan pengawasan pada setiap proses produksi.

□ *Process Lane Construction and Zone Outfitting*

Perkembangan teknologi pembangunan kapal modern telah dimulai pada tahap ini, dimana konsep *Group Technology* sudah diterapkan dalam proses produksi badan kapal dan pekerjaan O/F. Dengan konsep ini proses pembangunan kapal sudah berorientasi pada produk atau dikenal dengan *Product-Oriented Work Breakdown Structure (PWBS)*, dengan pengelompokan lingkup pekerjaan seperti terlihat pada gambar 2.1.



Gambar 2.1. Komponen *Product-Oriented Work Breakdown Structure*

*HBCM* : Hull Block Construction Method  
*ZOFM* : Zone Outfitting Method  
*ZPTM* : Zone Painting Method  
*PPFM* : Pipe Piece Family Manufacturing

Metode ini secara sistematis mengklasifikasikan produk antara (*inter-products*) menurut kelompok yang memiliki kesamaan dalam proses produksinya, misalnya process lane untuk : (i) perakitan bentuk dasar, (ii) perakitan bentuk lengkung beraturan, dan (iii) perakitan bentuk lengkung tidak beraturan dan kompleks. Hal ini sudah menuntut keteraturan dalam penataan sumber daya produksi, misalnya : peralatan/mesin, tenaga kerja, dan material. Demikian halnya dengan pekerjaan O/F, dimana sudah dilakukan secara paralel berdasarkan *region/zone*, dan tidak lagi berdasarkan sistem fungsionalnya. Pekerjaan O/F, dengan konsep *Advance Outfitting*, sudah dibagi dalam 3 tahap yaitu : *on-unit*, *on-block* dan *on-board*.

Pada tahapan ini, proses produksi sudah terkonsentrasi pada bengkel-bengkel dan volume pekerjaan di *building berth* semakin kecil. Penggunaan teknologi ini sudah mensyaratkan diterapkannya Sistem *Accuracy Control*, di setiap proses produksi untuk menjamin ketepatan dimensi dan bentuk blok (pelat, profil, perpipaan dan *inner-parts* lainnya yang ada pada daerah sambungan blok).

□ *Integrated Hull Construction, Outfitting, and Planning (IOHP)*

Metode pembangunan kapal ini merupakan teknologi yang paling mutakhir, dimana proses pembuatan badan kapal sudah diintegrasikan secara maksimal dengan pekerjaan *outfitting* dan pengecatan pada setiap *zone/area/stage*. Teknologi ini merupakan pengembangan dari teknologi produksi tahapan ketiga, melalui peningkatan potensi sumber daya galangan secara menyeluruh, seimbang dan terintegrasi, disertai dengan semakin sempurnanya standar-standar kerja dan konsistensi ketepatan proses produksi yang sangat tinggi.

Teknologi IOHP mensyaratkan suatu *Build Strategy* yang matang dan kemampuan *planning and scheduling* yang sangat tinggi dan rasional, berdasarkan kondisi potensi sumberdaya galangan yang ada. Pada tahap ini, Sistem *Accuracy Control* bukan hanya mutlak dilaksanakan secara sempurna dan menyeluruh, tetapi juga menuntut kesempurnaan *design engineering* dan *standar-standar* kerjanya. Dengan demikian, penyambungan seluruh bagian konstruksi antar blok yang satu dengan yang lainnya mencapai tingkat ketepatan yang tinggi atau penyimpangan-penyimpangan yang terjadi masih berada pada batas toleransi yang telah ditentukan, sehingga volume pekerjaan pada *building berth* menjadi sangat kecil.

## II.2. FAKTOR-FAKTOR YANG MEMPENGARUHI PROSES PRODUKSI

Proses pembangunan kapal merupakan pekerjaan yang mempunyai *kompleksitas* yang sangat tinggi. Sehingga pada proses pembangunan kapal sering terjadi kesalahan-kesalahan yang dapat berpengaruh pada proses produksi. Secara garis besar ada beberapa faktor utama yang sangat mempengaruhi proses produksi antara lain :

### □ **Material**

Pada pembangunan kapal material yang dibutuhkan sangat mempengaruhi jalannya proses produksi. Untuk material logam baja diperlukan proses pengerjaan meliputi pemotongan, *forming* (pembentukan), penggabungan pelat, dan profil. Berdasarkan proses pengerjaan masing-masing material yang berbeda maka peralatan yang dibutuhkan, kualifikasi tenaga kerja, tata letak

bengkel dan prosedur kerja yang digunakan sangat berbeda. Sehingga ada beberapa hal yang harus diperhatikan di dalam pemakaian material yaitu :

- Spesifik Teknik Material

Meliputi : sifat-sifat mekanik, seperti kekuatan tarik, tekan, kemuluran dan beberapa sifat kandungan kimia kandungan logamnya.

- Ukuran/dimensi

Meliputi : ketebalan, panjang, lebar dan kelurusannya.

- Pabrik pembuat/produsen

Meliputi : kualifikasi dari pabrik pembuat dilihat dari catatan mutu produk yang dihasilkan

- Lingkungan/cuaca/temperatur/kelembaban

Yaitu merupakan media dimana material tersebut dikerjakan atau diproses. Hal ini tergantung dari tempat proses pekerjaan dilakukan, mialnya untuk di Indonesia maka kondisi lingkungannya akan sangat berbeda dengan Eropa.

□ **Tenaga Kerja**

Kualifikasi tenaga kerja pada setiap proses produksi akan sangat bebeda, tergantung dari jenis proses pengerjaan juga jenis material yang akan dikerjakan. Sedangkan untuk pemakaian tenaga kerja dapat diklasifikasikan dari segi :

- Pendidikan formal/non formal

Meliputi latar belakang pendidikan seperti STM, Politeknik dan kualifikasi khusus seperti *Welder Qualification grade G3* dan lain-lain.

Hal ini sangat berpengaruh terhadap pemilihan tenaga kerja.

- **Pengalaman dan masa kerja**

Seorang tenaga kerja yang telah mempunyai pengalaman dibidang yang sesuai akan sangat membantu pada proses pelaksanaan pekerjaan.

- **Ketrampilan/Skill**

Ketrampilan khusus yang dimiliki oleh tenaga kerja akan sangat membantu didalam proses pekerjaan, misalnya pada teknologi pengelasan logam tipis, membending pelat dan lain sebagainya.

- **Sikap/karakter**

Sikap dan karakter setiap tenaga kerja akan sangat membantu pada pelaksanaan proses produksi sehingga menciptakan iklim kerja sesuai dengan yang diinginkan.

- **Metode Produksi**

Hal-hal yang harus diperhatikan pada penetapan metode produksi adalah yang berkaitan langsung dengan tugas masing-masing tenaga kerja, diantaranya :

- Standar dan Prosedur kerja (*Operating Procedure Agreement*)
- Urutan Pekerjaan (*Operating Instruction*)
- Peralatan/perlengkapan keselamatan kerja

- **Peralatan Produksi**

Pada penerapan peralatan kerja maka kita harus memperhatikan hal-hal sebagai berikut :

- Spesifikasi Teknik
- Kondisi Operasional
- Kalibrasi dan Sistem Pemeliharaan

### II.3. TAHAPAN PROSES PRODUKSI

Pada pembangunan kapal baru selalu melalui proses produksi. Proses pembangunan kapal itu sendiri dimulai sejak material datang sampai dengan penyerahan (delivery) kapal kepada pihak pemesan. Penerapan proses produksi kapal akan meliputi beberapa tahap diantaranya yaitu : *Persiapan Produksi* (meliputi perancangan dan persiapan gambar kerja, penyimpanan dan pemeriksaan material, pemeriksaan tengah kerja dan material), *Fabrikasi* (meliputi pemotongan dan pembentukan pelat), *Sub-Assembly* (meliputi penggabungan beberapa komponen dasar), *Assembly* (meliputi penggabungan beberapa komponen dasar menjadi blok utama), *Erection* (meliputi penggabungan blok-blok utama menjadi kapal secara utuh).

Pada masing-masing tahap proses produksi tersebut selama pelaksanaannya sering terjadi penyimpangan proses produksi sehingga hal ini akan menghambat jalannya proses produksi karena waktu yang diperlukan untuk pembangunan kapal akan bertambah karena terjadinya pekerjaan ulang (*rework*), hal ini tentu akan menambah biaya produksi.

### II.3.1. TAHAP PERSIAPAN PRODUKSI

Pada galangan yang akan melaksanakan pembangunan kapal, tahap pertama yang harus dilaksanakan adalah tahap persiapan produksi. Dalam tahap ini mempunyai tujuan mengatur kegiatan persiapan produksi sehingga pada waktu yang telah ditentukan pekerjaan pembangunan kapal dapat dilaksanakan dan ditetapkan. Pada tahap ini ruang lingkup yang dikerjakan adalah :

- Dokumen produksi (umum), yang meliputi : gambar dan daftar material, perkiraan kebutuhan tenaga kerja, perkiraan kebutuhan material.
- Tenaga kerja yang terlibat dalam kaitannya dengan kualifikasi dan jumlah tenaga kerja dan pekerjaan yang lain.
- Material yang perlu dipersiapkan dengan mempertimbangkan keadaan dan stock gudang, pemakaian material untuk pekerjaan sekarang, pemesanan/pembelian material dari luar (jumlah dan waktu pemeberian).
- Fasilitas dan sarana produksi meliputi : kemampuan bengkel produksi, kapasitas mesin-mesin, alat-alat angkat yang tersedia (jumlah, kapasitas, macam dan tempat).

### II.3.2. TAHAP FABRIKASI

Pada proses fabrikasi ini, yang penulis amati adalah proses fabrikasi yang terjadi pada Departemen Fabrikasi lambung dalam Divisi Kapal Niaga di PT. PAL INDONESIA. Departemen ini bertugas untuk membuat komponen kapal dari

yang kecil hingga yang terbesar. Jadi bengkel ini merupakan bengkel awal dari pembuatan kapal secara fisik. Pada departemen ini, gambar-gambar dan rambu-rambu yang diperlukan sebagai berikut :

□ **Marking List**

Untuk mengetahui bentuk dari komponen-komponen yang akan dikerjakan dalam satu blok. *Marking List* ini memuat antara lain : nomor kapal, nomor blok, serta ukuran-ukuran dan tanda-tandanya.

□ **Material List**

Untuk mengetahui jumlah dari komponen-komponen yang akan dikerjakan dalam satu blok, berat blok tersebut, dan tempat komponen-komponen tersebut dikerjakan. Selain itu juga untuk mengecek komponen-komponen yang belum dikerjakan.

□ **Cutting Plan**

Untuk mengetahui jumlah dari material-material yang dibutuhkan dalam satu blok baik berupa pelat, profil maupun material yang lain. Pada *cutting plan* ini tercantum pula gambar perencanaan posisi penggunaan komponen-komponen yang akan dipotong.

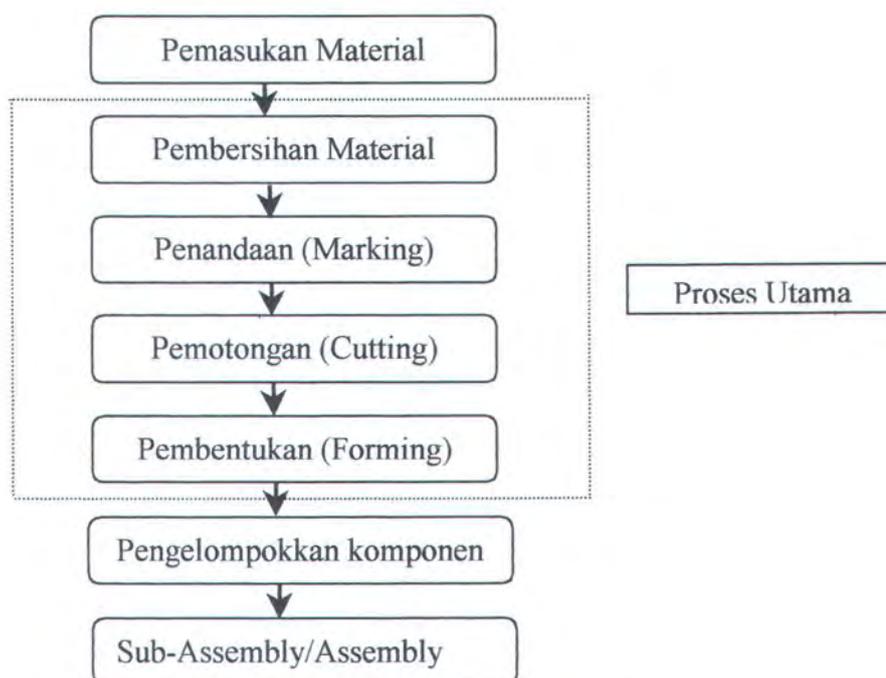
□ **Working Drawing**

Adalah gambar bentuk blok-blok yang sebenarnya. Untuk bengkel *Fabrikasi* hanya digunakan untuk mengecek kemungkinan terjadinya penyimpangan-penyimpangan setelah komponen selesai dibuat dan telah dikirim ke bengkel *Assembly*

#### □ Lift and Scaffold Piece

Digunakan untuk membuat cincin-cincin (kupingan-kupingan) yang digunakan untuk mengangkat blok-blok yang sudah selesai diassembly untuk diturunkan di dok. Adapun bentuk dan ukurannya tergantung pada berat blok yang diangkat.

Secara prinsip proses produksi pada bengkel *Fabrikasi* dapat digambarkan sebagai berikut :



Gambar 2.2 Proses Produksi Bengkel Fabrikasi

Keterangan gambar :

#### □ Proses pembersihan

Sebelum proses pembersihan ini dilakukan maka terlebih dahulu diadakan identifikasi material yang merupakan suatu tindakan pemeriksaan material yang akan dipakai meliputi :

- *Charge Number Material*
- Klasifikasi Material
- Dimensi Material
- Kondisi Permukaan Material

Selain itu hasil pemeriksaan material dicatat dalam suatu laporan pemeriksaan (Check Sheet) sekaligus perbaikannya apabila material tersebut mengalami cacat. Pada pelat yang bergelombang harus diperbaiki dengan pelurusan, dengan menggunakan *roll machine*. Pelat tersebut dilewatkan pada suatu susunan selinder baja. Setelah itu baru dimasukkan dalam *short blasting* dan *primary machine*. Alat ini berguna untuk menghilangkan karat dan kotoran yang melekat pada material dengan menggunakan butiran-butiran baja berdiameter  $\pm 0,5 - 0,8$  mm yang disemprotkan dengan tekanan tinggi kemudian dicat dengan menggunakan cat dasar.

#### □ Proses Marking

Proses marking yaitu proses penandaan pada permukaan material yang akan mengalami pekerjaan sesuai dengan ketentuan tanda kerjanya. Pemandahan data *mould loft* serta pengukuran-pengukurannya harus dilakukan seakurat mungkin, karena kesalahan pada proses marking ini tidak hanya menyebabkan komponen-komponen rusak dan penyimpangan bentuk pada material tetapi juga akan menaikkan jumlah material yang terbuang dan pada akhirnya akan mengurangi nilai produktivitas galangan. Langkah-langkah pekerjaan *marking* :

- Pelat diletakkan di atas lantai yang rata, dan dicek apakah material tersebut sesuai dengan yang ada di dalam daftar.

- Rencana pemotongan (*cutting plan*), dan pada gambar rencana pemotongan ini diusahakan sisa material sekecil mungkin.
- Pembersihan material, sebelum di-*marking* pelat dibersihkan terlebih dahulu agar kapur dapat melekat betul.
- Persiapan alat-alat kerja.
- Penandaan/pemberian nama pada material

Penandaan setiap bagian dari material yang telah ditandai harus diberi nama yang jelas agar nantinya tidak tertukar ataupun keliru pada saat perakitan. Nama tersebut disesuaikan dengan kode yang tercantum pada *material list* atau *marking list*. Nama material terdiri dari : Nomor kapal, Nama kapal, Nama komponen dan posisinya (Port side atau Starboard side), Posisi *marking* (up atau low marking).

#### □ Proses *Cutting* (pemotongan)

Proses *cutting* adalah proses pemotongan material-material yang telah di-*marking* terlebih dahulu dan telah mendapat persetujuan dari *Quality Assurance*.

Alat-alat pemotongan antara lain sebagai berikut :

- *Manual Gas Cutting*
- *Semi Automatic Gas Cutting (Scattor)*
- *NC Gas Cutting Machine*
- *Flame Planner Cutting Machine*

#### □ Proses *Forming* (Pembentukan)

Pada proses *forming* ini dibedakan antara pelat dan profil. Pada pembentukan material pelat ada dua cara yang dilakukan yaitu :

- *Heat Forming (Fairing)*

Cara ini banyak digunakan untuk bentuk-bentuk tiga dimensi atau sebagai penyempurnaan bentuk dari pelat yang telah dibending dengan mesin. Pada prinsipnya cara ini adalah memanaskan pelat kemudian mendinginkan secara mendadak.

- *Cold Forming*

Cara ini dengan menggunakan *bending machine*, biasanya untuk bentuk-bentuk sederhana, seperti : *Roll bending machine* dan *Horisontal bending machine*.

Sedangkan untuk profil, langkah-langkah yang diperlukan untuk pembentukan adalah :

- Persiapan material antara lain : *profile* yang akan dibending/*fairing*, rambu film, landasan untuk *fairing*, jig dan lain-lain. Harus diperhatikan gambar-gambar yang ada di rambu film dan gambar yang ditempatkan dengan kedudukan terbalik.
- Dilakukan penitikan pada tempat-tempat tertentu sesuai dengan rambu film, titik-titik tersebut dihubungkan dengan menggunakan *stroklat* kayu dan *sumitsasi* atau *sumitsubo*.
- Dilakukan pembendingan sedikit demi sedikit dan pengecekan setiap kali pembendingan.

### II.3.3. TAHAP SUB ASSEMBLY

Proses *sub assembly* ini merupakan proses kelanjutan dari bengkel

fabrikasi. Pekerjaan ini meliputi :

- Penyambungan pelat
- Penyambungan *stiffener*
- Merakit *Floor*
- Pemasangan *Face Plate*
- Merakit *Web Frame*

Dalam proses pekerjaan data-data yang diperlukan antara lain : *Yard plan*, *Working Drawing*, *Material List*, *Cutting plan*, *Marking List*.

#### II.3.4. TAHAP ASSEMBLY

Pada proses *assembly* ini pekerjaan yang dilakukan adalah merakit panel-panel datar, panel lengkap hingga menjadi panel datar, seksi lambung, seksi sekat dan sebagainya hingga menjadi satu blok.

##### □ Perakitan panel

Pada perakitan panel ini, pelat-pelat diletakkan pada *laticce floor* diatur dan dilas ikat. Urutan pengelasan ikat maupun pengelasannya dimulai dari tengah-tengah panel kemudian secara bertahap keluar, hal ini dilakukan untuk mengurangi deformasi.

Untuk panel-panel yang dibuat dari sambungan banyak pelat dilakukan dengan jalan memberikan beban berat untuk menekan pelat yang akan dilas supaya deformasi yang terjadi sekecil mungkin.

Setelah pelat dilas menjadi satu, pelat diletakkan di atas meja *jig* yang telah disediakan. *Jig* itu harus di-*check* levelnya maupun counter dari panel,

setelah itu dilakukan juga pemeriksaan posisi *reference line*, terutama untuk *center line*, *water line*, dan *frame line*-nya.

Selanjutnya profil-profil dipasang pada garis-garis *marking* yang telah ditentukan., kemudian profil ini kemudian dilas panel

#### □ Perakitan Blok

Perakitan blok ini merupakan kelanjutan dari perakitan panel-panel. Di sini perlu diperhatikan pengaturan letak atau posisi dari seksi-seksi sehingga dapat mengurangi pengelasan yang sulit. Pada umumnya bagian yang datar diletakkan pada bagian bawah.

### II.3.5. TAHAP ERECTION

Pada proses erection adalah kelanjutan dari proses-proses sebelumnya yaitu proses *sub-assembly* dan *assembly*. Jenis pekerjaan yang dilakukan pada tahap *erection* adalah :

#### □ Loading

Cara kerja :

- Blok yang ada di pelataran kerja diangkat dengan *crane* yang disesuaikan dengan kapasitas angkatnya.
- Blok yang diangkat harus seimbang dengan meletakkan *ballast* berupa batu cor dengan berat tertentu dan tempat tertentu pula.
- Letak dan besar kupingan harus diperhitungkan.
- Blok ditempatkan pada keel blok dan side blok yang telah diatur sesuai dengan *marking dock*.

### □ Adjusting

Cara kerja :

- Blok baru yang akan ditempatkan harus diluruskan *center line* dengan blok yang lama.
- Blok baru yang akan ditempelkan dikerjakan sesuai dengan ukuran yang telah ditentukan setelah *fitting* dan *welding*.
- Mengatur paju pada *keel block* bila kurang tepat atau rapat menempel pada badan kapal atau blok agar tidak terjadi gerakan.

### □ Fitting

Cara kerja :

- Bagaian *center line* harus lurus antara blok yang satu dengan blok yang lain.
- Melihat sisi-sisi bagian kapal apakah sudah tepat dan sesuai dengan ukuran. Bila tidak, maka dilakukan pengukuran jarak dari *center line* ke sisi badan kapal sesuai dengan gambar kerja. Bila terlalu lebar, maka *frame* dibongkar dan di-*fairing* atau di-*bending* menurut gambar kerja.
- Apabila sudah tepat maka ujung-ujung pelat dibentuk kampuh yang sesuai tebal pelat atau gambar kerja.
- Setelah dibuat kampuh las kemudian diperiksa lagi kelurusan *center line*, bagian pelat dasar, pelat sisi dan pealt geladak.
- Kemudian dilakukan las ikat di tempat-tempat tertentu menurut gambar kerja.

#### □ **Welding**

Cara kerja :

- Setelah di-*fitting* dilakukan pemeriksaan oleh *QA* dan *Class*.
- Bila *QA* dan *Class* telah menyetujui maka dilakukan pengelasan (*welding*) untuk menyambung blok-blok tersebut.
- Selain mengelas pelat sisi, pelat dasar dan pelat geladak juga dilakukan pengelasan atau penyambungan profil-profil pembujur yang dimulai dari *center line* lalu menepi.

#### □ **Fairing**

Cara kerja :

- Dinding yang cembung dipanasi dengan brander las sampai  $650^{\circ}$  –  $900^{\circ}$  C tergantung dari tebal pelat.
- Bagian dinding yang cekung disemprot dengan air untuk menarik struktur material akibat pengerutan material tersebut
- Pemanasan dilakukan merata berupa titik-titik diseluruh dinding yang mengalami deformasi.

### **II.4. PENGAWASAN PROSES PRODUKSI KAPAL**

Perkembangan teknologi produksi di bidang pembangunan kapal mengalami perkembangan yang sangat pesat dari tahun ke tahun. Salah satu bentuk teknologi produksi yang mendapat perhatian yang cukup serius adalah peningkatan pengawasan dan pengendalian mutu di galangan kapal

Pengawasan terhadap setiap proses produksi kapal sangat besar pengaruhnya untuk menjaga kualitas produk. Lebih-lebih untuk pembangunan kapal yang sudah menggunakan sistem *Integrated Hull Construction, Outfitting and Planning* (IHOP) dimana sistemnya sudah terintegrasi.

Fungsi pengawasan terhadap kualitas produksi yang terbatas pada usaha pemenuhan kualitas berdasarkan standar klasifikasi dan standar pemesan. Disamping kedua standar tersebut diatas, diperlukan pula standar galangan yang baru dapat dicapai apabila ditunjang dengan suatu standar kualitas galangan. Standar kualitas diperlukan sebagai tindakan preventif untuk mnghindarkan terjadinya kesalahan atau pekerjaan ulang pada setiap proses pekerjaan.

Tujuan akhir dari dilaksanakannya pengawasan produksi ini adalah untuk menekan biaya produksi dan meningkatkan mutu kapal hasil pembangunan suatu galangan.

Pengawasan haruslah dilakukan sedini mungkin yakni dimulai dari desain, produksi sampai dengan performance hasil produksi tersebut, sehingga penyimpangan-penyimpangan dari standar kualitas maupun spesifikasi dapat dihindari sedari awal dan apabila dialami atau ditemui penyimpangan-penyimpangan dapat diatasi sesuai dengan prosedur yang diijinkan. Dengan demikian biaya produksi dapat ditekan atau ditepati dan yang tidak kalah pentingnya adalah kualitas produksi dapat dijamin/dipertanggungjawabkan.

Dalam pelaksanaan pengawasan produksi, pengawasan dan pemeriksaan ketepatan ukuran dilakukan tiap hari menurut jadwal yang telah ditentukan yaitu :

jadwal pembangunan kapal, dengan kegiatan pokok mengadakan pemeriksaan, pengukuran, dan pencatatan data-data hasil pengukuran.

Pemeriksaan secara langsung atau visual mengenai mutu hasil pekerjaan pada setiap proses produksi dilakukan secara intensif oleh *checker*, *QA / QC* dan manager proyek. Sedangkan peranan *suveyor* BKI dan *owner surveyor* dalam hal pengawasan dan pengendalian mutu adalah memeriksa agar mutu kapal tersebut tidak menyimpang dari standar dan spesifikasi yang telah ditentukan sebelumnya.

Tempat-tempat yang perlu di-*check* dapat langsung diketahui oleh pekerja melalui lembar *checksheet* yang diterima oleh *QA/QC*. Untuk setiap tahap perjaan, item-item yang perlu diperiksa kecocokannya dengan standar yang telah ditentukan adalah sebagai berikut :

a. Tahap Persiapan

Meliputi : dimensi-dimensi secara keseluruhan termasuk kelebihan-kelebihan yang diberikan dan tanda-tanda yang dimaksud untuk fabrikasi, perakitan dan pemeriksaan pekerjaan.

b. Tahap Fabrikasi

Meliputi : dimensi-dimensi secara keseluruhan dari pelat-pelat atau bagian-bagian yang telah dipotong, persiapan-persiapan permukaan, deformasi, dan kelengkungan dari komponen yang melengkung.

c. Tahap Perakitan (sub assembly dan assembly)

Meliputi : penyetelan komponen-komponen yang dirakit, jarak antara dua bagian dilas dan menjaga kelurusan badan kapal.

#### d. Tahap Penyambungan (Erection)

Meliputi : penyambungan block-block, gap-gap untuk pengelasan dan pemeliharaan badan kapal.

*QA/QC* memberikan umpan balik (feed back) kepada bengkel *Assembly* mengenai masalah-masalah yang muncul pada blok. Umpan balik tersebut juga memberikan komentar mengenai titik-titik yang *vital* dan dimensi-dimensi yang berkaitan dengan proses *erection*.

Tujuan dari pada umpan balik tersebut adalah untuk memberikan informasi dan komentar mengenai bagaimana dan apa yang seharusnya dilakukan untuk memperbaiki detail-detail desain, toleransi-toleransi dan metode-metode.

#### **Sistem Pemeriksaan Kualitas Lambung**

Sistem pemeriksaan lambung dilaksanakan dengan *Check Sheet System* yang mana pemeriksaan kualitas hasil produksi dengan menggunakan check sheet (lembar pemeriksaan) sesuai dengan proses pembangunan kapal.

#### **Pemeriksaan pada kualitas lambung perlu dilakukan karena :**

- Hasil kerja dapat diawasi dan dievaluasi sedari awal sampai dengan selesai oleh Surveyor galangan.
- Surveyor Badan Klasifikasi pada beberapa pemeriksaan mungkin cukup hanya memeriksa hasil *check sheet* saja.
- Penyampaian informasi tentang problem teknik dapat diperoleh melalui analisa pada *check sheet* yang dihasilkan.

*QA/QC* yang beranggotakan pekerja pada bagian desain konstruksi bertugas membuat *check sheet* yang berisi tempat-tempat yang akan di-check, tempat pencatatan data, penyimpangan yang diijinkan, serta waktu pencatatan, *working area*, tahap pekerjaan, identitas pelaksana dan identitas petugas *QA/QC*. *Check sheet* harus dipersiapkan untuk semua tahapan proses produksi mulai dari *fabrikasi* sampai dengan *erection*.

#### □ **Check Sheet untuk Fabrikasi**

Fabrikasi adalah awal tahapan proses pembangunan kapal. Pekerjaan yang dilaksanakan adalah *marking*, *cutting* dan *bending*. Kesalahan atau penyimpangan yang kecil akan mempengaruhi proses selanjutnya. Oleh karena itu diperlukan standar akurasi tertentu. Komponen yang diperiksa dan diukur dipilih secara random oleh bagian *mould loft* dikarenakan material fabrikasi sangatlah banyak jumlahnya. Petunjuk-petunjuk pengukuran ditentukan oleh bagian *mould loft*.

Yang perlu diperhatikan pada waktu pengukuran adalah :

- Derajat kemiringan rambu-rambu.
- Ketepatan sisi pelat dengan garis tanda pada *template*.
- Jarak antara sisi *template* dengan pengukuran pelat.
- Kelengkungan memanjang dan melintang.
- Puntiran dan kelurusan garis pandang sumbu referensi pada rambu.

#### □ **Check Sheet untuk sub-blok Assembly**

Yang penting pada sub-blok *assembly* untuk *accuracy control* adalah menempatkan *stiffener*, letak *bracket*, *face plate* dan sebagainya. Bagaimana

menghindari deformasi dan penyusutan yang disebabkan oleh pengelasan.

Aktivitas pengukuran selama proses *sub-blok assembly* adalah :

- Pengecekan dimensi *fitting*.
- Pengukuran panjang, lebar dan diagonal pelat.
- Pengecekan untuk deformasi dan penyusutan dengan menggunakan *references lines* pada *web plate*.
- Pengukuran dimensi-dimensi seperti tercantum pada *check sheet*.

Mempertahankan akurasi pada blok *assembly* adalah penting karena ini adalah tahap akhir sebelum blok dierection di *building berth*. Pemeriksaan terhadap bentuk blok lurus dengan bentuk blok lengkung harus dibedakan. Hal-hal yang perlu diperhatikan pada pengukuran adalah :

- Pengukuran panjang, lebar dan diagonal setelah *base panel* di-*assembly*.
- Puntiran.
- Lokasi dari sub-blok dan internal part setelah dipasang pada *base panel*.

Untuk bentuk blok lengkung hal-hal yang perlu dipertahankan adalah :

- Pengukuran panjang, lebar dan diagonal blok.
- Pengukuran petunjuk ketepatan lokasi pelat lengkung untuk sebuah *base panel* pada sebuah *pin jig*.
- Sarana untuk pemeriksaan.
- Pengecekan pada dimensi-dimensi yang dibutuhkan pada tepi-tepi panel yang mana nantinya akan dibutuhkan pada waktu penyambungan di *erection*.

- Instruksi-instruksi bagaimana untuk mendeteksi dan mengoreksi deformasi yang disebabkan oleh pengelasan.
- Puntiran.

□ **Check Sheet untuk Erection**

Selama proses *erection* harus dimonitor dimensi-dimensi blok seperti panjang, lebar *leveling* dan tinggi karena ini menunjang ukuran kapal sesuai yang disetujui pemesan. Data-data hasil pemeriksaan serta variasi-variasi yang mungkin muncul harus dicatat. Anggota *QA/QC* secara periodik melakukan pengecekan terutama pada vital-vital point dari mulai *keel laying* sampai dengan *launching*.

## BAB III

# PEMERIKSAAN PENYIMPANGAN BENTUK PADA PROSES PRODUKSI KAPAL

### III.1. TAHAPAN PEMERIKSAAN PADA PROSES PRODUKSI KAPAL

Pada setiap proses produksi kapal di suatu galangan, pelaksanaan pengawasan/pemeriksaan produksi sangat besar pengaruhnya terhadap hasil pembangunan kapal, terutama untuk galangan-galangan yang telah menerapkan metode pembangunan kapal dengan sistem blok. Metode dengan sistem ini membutuhkan ketepatan ukuran yang sangat tinggi dalam proses produksinya.

Untuk itu pemeriksaan haruslah dilakukan sedini mungkin yakni dimulai dari persiapan produksi sampai dengan *performance* hasil produksi tersebut. Sehingga penyimpangan-penyimpangan dari standar kualitas maupun spesifikasi dapat dihindari sedari awal dan apabila ditemukan penyimpangan-penyimpangan dapat diatasi sesuai dengan prosedur yang telah diijinkan.

Dalam pelaksanaan pemeriksaan proses produksi haruslah dilakukan tiap hari menurut jadual yang telah ditentukan. Pemeriksaan secara langsung (*visual*) mengenai mutu hasil pekerjaan pada setiap proses produksi dilakukan secara intensif oleh *checker*. Tempat-tempat yang perlu di-*check* dapat langsung diketahui oleh pekerja melalui lembar *check sheet* yang diterima oleh *QA/QC*. Untuk tiap-tiap tahapan pemeriksaan terdapat item-item yang perlu diperiksa kecocokannya dengan standar yang telah ditentukan.

### III.1.1. PEMERIKSAAN PADA TAHAP FABRIKASI

#### □ Pemeriksaan Proses Marking

Pelaksanaan pemeriksaan dilakukan pada *marking list* berikut gambar-gambarnya, prosesnya serta keseluruhan pekerjaam *marking*. Pemeriksaan pada tahap awal dilakukan oleh intern bengkel yaitu *marker* yang disertai oleh kepala regunya dengan prosedur pemeriksaan seperti dalam proses-proses *marking* yang sudah ada. Kemudian kepala bengkel dan kasié melakukan observasi pada hasil-hasil *marking* yang telah diperiksa dan diadakan perbaikan kesalahannya dan memberikan komentar dari hasil observasinya dalam *inspection record*.

Untuk material-marked yang akan mengalami proses *cutting* harus mendapatkan rekomendasi *ok-cut* dari *QA/QC* dan *surveyor class*. Untuk hal ini *QA* melakukan pemeriksaan pendahuluan terhadap *marking list* dan gambarnya. Kemudian dilakukan pemeriksaan pendahuluan pada material-marked dengan pelaksanaan pemeriksaan sesuai kebutuhan gambar *marking* dari desain dimana sarana pemeriksaan dengan menggunakan *check sheet* untuk *Accuracy Control* bagi tahap fabrikasi tersebut (*marking* dan *cutting*) telah dicantumkan ukuran dimensi sebagai acuan pemeriksaaan untuk tiap-tiap bagian atau komponen. Di dalam *check sheet* ini juga disebutkan siapa yang memeriksa, bagian yang diperiksa serta keterangan mengenai ketentuan pemeriksaan seperti sampel yang harus diambil setiap hari. Kemudian juga disediakan kolom keterangan pemeriksa, hala ini berkaitan dengan informasi yang diperoleh dari pihak bengkel. Kemudian pihak *QA* bersama *surveyor class-owner*, setelah yakin dengan hasil *marking* menyetujui dan mengijinkan dilakukan dengan memberi tanda *ok-cut*.

Dari uraian di atas urutan pemeriksaan dengan pemeriksa mempunyai tiga tahap, yaitu :

Nama	Petugas	Warna Kapur
Pemeriksa no. 1	Kasi, Kabeng	Putih
Pemeriksa no. 2	Divisi QC	Kuning
Pemeriksa no. 3	Surveyor QC/A	Hijau
Final Inspeksi	Surveyor class dan surveyor owner	Merah

Tabel 3.1

#### □ Pemeriksaan Proses Cutting

Pemeriksaan pertama kali dilakukan oleh pekerja sendiri, dimana apabila terdapat perbaikan-perbaikan yang perlu dilaksanakan harus disetujui oleh kabeng. Hal ini berkaitan dengan schedule produksi dan ketentuan prosedur pekerjaan sebagai standarisasi pekerjaan perbaikan serta ketentuan akan besar *excess material* dan margin. Untuk keperluan pemeriksaan atau evaluasi oleh *QC/A*, kabeng membuat dan menyusun catatan pemeriksaan yang berisikan hasil pemeriksaan akurasi dimensi, prosedur-prosedur perbaikan, serta kondisi-kondisi pekerjaan dan peralatan yang digunakan.

Untuk urutan pemeriksaan pada proses *cutting* seperti yang diuraikan sebelumnya dalam proses *marking*, yaitu sesuai dengan urutan proses *cutting* yang benar.

#### □ Pemeriksaan Proses Bending

Seperti halnya *marking* dan *cutting*, prosedur pemeriksaan pada proses bending dilakukan dalam tiga tahap yaitu: pekerjaan kepala regu, kabeng-kasie

dan divisi QC, hal ini untuk intern bengkel. Kemudian dilanjutkan oleh surveyor QC/QA untuk sistem informasi standar dan persiapan *surveyor class-owner*. Semua informasi mengenai metode dan jadwal pemeriksaan diberikan oleh bagian QC/QA dimana semuanya disusun dalam suatu check sheet. Dana urutan pemeriksaannya sama dengan urutan pemeriksaan pada proses-proses sebelumnya.

Proses pemeriksaan pada tahap fabrikasi, berdasarkan urutan pemeriksaannya dan siapa yang melaksanakan dapat dibagi menjadi :

No	Pemeriksaan	Pemeriksa	Keterangan
1	Awal	Pekerja	Perbaikan dengan rekomendasi kabeng (record kondisi dan penyimpangan)
2	Periksa I	Kasi & Kabeng	Kapur putih dan check sheet (record data penyimpangan)
3	Periksa II	Divisi QC	Kapur kuning dan check sheet (record data penyimpangan dan keterangan)
4	Periksa III	Surveyor QC/A	Kapur hijau dan check sheet (record data penyimpangan dan keterangan)
5	Periksa Tahap Akhir	Surveyor class dan owner	Kapur merah dan check sheet (record data penyimpangan dan keterangan)

Tabel 3.2

### III.1.2. PEMERIKSAAN PADA TAHAP SUB ASSEMBLY

Pekerjaan yang ada pada tahap *sub assembly* sama dengan kegiatan-kegiatan yang ada pada tahap sebelumnya meliputi: *fitting*, *welding* dan *marking* akhir (*finishing*). Pelaksanaan pemeriksaan penyimpangan bentuk dan dimensi dilakukan pada tiap kegiatan tersebut, dimana perhatian difokuskan pada kegiatan

*fitting*. Hal ini dikarenakan kegiatan *fitting* merupakan pekerjaan penentu dari posisi lanjut dari *members/parts* yang dipasang, dimana bila terjadi kesalahan memerlukan waktu dan biaya operasi yang cukup besar.

Beberapa petunjuk yang dapat dipakai sebagai pertimbangan pelaksanaan pemeriksaan pada tahap *sub assembly* adalah sebagai berikut :

- Pekerjaan pada tahap ini merupakan langkah awal dalam memproses potongan-potongan pelat datar yang sesuai dengan gambar desain menjadi bentuk-bentuk tiga dimensi (3-D). Jika presisi dari dimensinya tidak terjamin, maka akan timbul berbagai masalah pada proses pekerjaan selanjutnya.
- Memeriksa gambar-gambar dan mengartikannya secara tepat untuk kemudian memilih prosedur dan metode *assembly* yang efisien
- Memeriksa tabel *members* atau *parts* untuk menentukan jumlah dan jenis *members*

### III.1.3. PEMERIKSAAN PADA TAHAP ASSEMBLY

Pemeriksaan yang dilakukan pada tahap *assembly* dilaksanakan pada proses pekerjaan sebagai berikut :

- **Pemasangan dan Penyambungan Pelat**
  - Pemeriksaan terhadap material pelat :
    - Pemeriksaan simbol-simbol pada pelat meliputi; nama, nomor dan lokasi pelat
    - Pemeriksaan terhadap marking dan posisi pelat
    - Pemeriksaan terhadap sudut bevel

- Pemeriksaan bentuk alur pengelasan *butt joint* terutama terhadap sudut-sudut bevel kampuh las dengan kesesuaiannya dengan besar gap dan keseragaman lebarnya sepanjang alur las tersebut.
  - Pemeriksaan kelurusan atau kedataran permukaan pada sisi sambungan dengan menggunakan *face alignment piece* disertai pengukurannya dengan *water pass*.
  - Komponen yang dihasilkan dari proses pengelasan tersebut diperiksa lagi yang meliputi kelurusan atau kedataran permukaan pelat dan bentuk puntiran sambungan. Pemeriksaan dengan menggunakan pengukuran lebar, panjang dan diagonal dari proyeksi bentuk komponen tersebut disertai dengan *levelling* dengan *water pass*.
- **Marking dan Cutting**
- Pengukuran lebar sambungan pelat yang disesuaikan dengan ketentuan pada gambar kerja.
  - Penandaan *basic frame line* pada arah memanjang dengan memeriksa ukuran diagonal sambungan pelat dan menandai *basic frame line* pada titik temu secara *orthogonal* ke arah memanjang di tengah-tengah blok.
  - Pengukuran panjang dari *basic frame line*, sesuai ukurannya.
  - Pemeriksaan *marking* dan *cutting* seperti pada tahap fabrikasi
- **Welding**
- Pengukuran ketegaklurusan *fitting* pembujur dan pelintang terhadap pelat yang dipasang.

- Pemeriksaan hasil pengelasan terhadap akibat deformasi pengelasan yaitu distorsi ataupun puntiran.
- Pemeriksaan puntiran dengan cara memberi tanda pada ketinggian tertentu di setiap ujung-ujung pembujur dengan menggunakan *water level*.
- Pemeriksaan terhadap cacat-cacat akibat pengelasan, dimana pengaruhnya terhadap akurasi bentuk dan dimensi terletak pada penyusutan ukuran karena proses perbaikannya.

#### III.1.4. PEMERIKSAAN PADA TAHAP ERECTION

Proses pemeriksaan bentuk dan dimensi pada tahap *erection* ini diutamakan pada daerah sambungan anatar blok maupun seksi. Pelaksanaan pemeriksaan secara global dilakukan pada hasil akhir pengelasan *erection*, yaitu terhadap ukuran-ukuran utama kapal setelah penyambungan semua blok dan seksi telah selesai.

Prosedur pelaksanaan pemeriksaan tetap sama dengan tahap sebelumnya yaitu dengan pemeriksaan tiga tahap, kabeng/kasie, divisi *QC*, surveyor *QC/QA* ditambah dengan pemeriksaan *surveyor class-owner*. Sedangkan sarana yang dipakai selama proses pemeriksaan adalah tetap *check sheet* dan sebagai pedoman akurasinya menggunakan standar yang telah ditetapkan.

Proses pemeriksaan komponen kapal selama proses pembangunan kapal seperti yang telah diuraikan di atas merupakan metode pemeriksaan yang digunakan selama ini. Dengan demikian kami mencoba mengembangkan dengan suatu cara dengan menggunakan komputer grafis. Dengan memvisualisasikan

komponen tiga dimensi (3-D) kapal dan dapat dipakai untuk proses analisa selanjutnya.

### **III.2. JENIS-JENIS PENYIMPANGAN BENTUK KARENA DEFORMASI**

Di dalam proses produksi kapal baja pekerjaan yang ada di dalamnya merupakan pekerjaan yang menimbulkan panas, baik pada proses pengerjaan dingin ataupun pada pengerjaan panas. Pada pengerjaan dingin misalnya pemotongan pelat menggunakan *guillotine*, bending pelat, *rolling* pelat. Sedangkan pengerjaan panas misalnya pada pemotongan pelat dengan *gas cutting*, pengelasan dan *hot forming*. Adanya panas tersebut akan menyebabkan terjadinya deformasi pada material pelat tersebut.

Deformasi yang sering terjadi adalah deformasi pelat pada konstruksi lambung kapal, terutama pada proses produksi tahap *sub assembly* dan *assembly*. Penyimpangan bentuk pada konstruksi lambung kapal yang berupa deformasi tersebut harus dilakukan perbaikan karena penyimpangan tersebut akan mempengaruhi proses produksi selanjutnya.

#### **III.2.1. PENYIMPANGAN BENTUK KARENA DEFORMASI PENYUSUTAN**

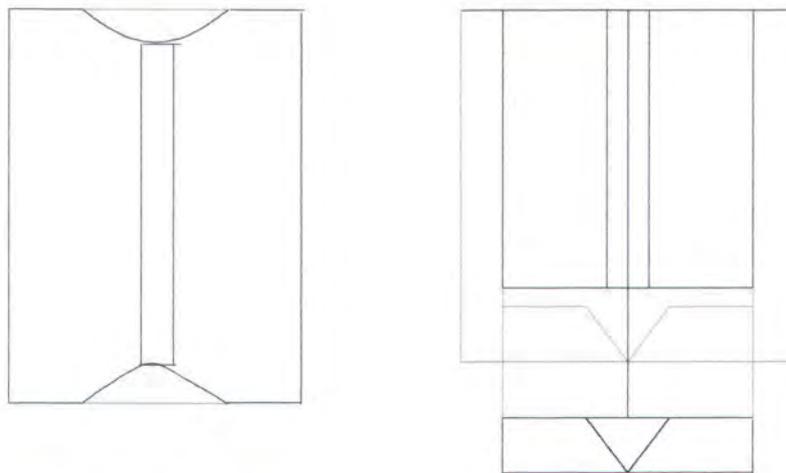
Penyusutan ini dikarenakan proses pengerjaan pada material menggunakan pemanasan, yaitu pada pengelasan dan pemotongan pelat menggunakan *gas cutting*. Pada proses pengelasan pada bagian yang dilas menerima panas setempat dan selama proses pengelasan berlangsung terjadilah distribusi panas yang tidak

merata. Pada bagian yang dilas, karena adanya panas maka pada bagian ini akan mengalami pengembangan termal, sedang bagian yang lain suhunya tidak berubah sehingga seolah-olah terjadi penghalang peregangan.

Sedangkan pada proses pemotongan pelat menggunakan *gas cutting* dapat menyebabkan terjadinya penggelombangan pelat pada daerah garis pemotongan dan pengurangan ukuran akibat adanya bagian pelat yang terbuang yang disebut *kerf*.

□ **Deformasi penyusutan karena pengelasan**

Deformasi penyusutan akibat pengelasan terdiri dari dua jenis yakni penyusutan melintang dan penyusutan memanjang dari konstruksi las-lasan. Penyusutan memanjang sangat kecil dibandingkan dengan penyusutan melintang, yaitu karena adanya perlawanan dari logam induknya. Hal ini dapat dilihat pada gambar dibawah ini :



Gambar 3.1. Penyusutan memanjang dan melintang

Sedangkan faktor-faktor yang mempengaruhi deformasi penyusutan akibat proses pengelasan yaitu :

1. Tebal Pelat

Ketebalan pelat mempengaruhi penyusutan melintang pada sambungan las tumpul. Semakin tebal pelat semakin kecil penyusutan melintangnya dan ketika sampai pada ketebalan pelat tertentu, penyusutannya akan konstan.

2. Berat Logam Las.

Penyusutan melintang akan semakin besar jika berat logam yang masuk pada saat pengelasan semakin bertambah besar.

3. Celah Pengelasan

Penyusutan melintang pada celah akar pengelasan dengan alur  $V$  akan bertambah besar dengan bertambah besarnya tebal pelat.

□ **Deformasi penyusutan karena proses pemotongan**

Deformasi penyusutan yang disebabkan oleh proses pemotongan menggunakan *gas cutting* dipengaruhi oleh beberapa faktor yaitu :

1. Penandaan

Penandaan pada pelat yang kurang teliti akan akan menyebabkan terjadinya pengurangan ukuran. Sehingga operator dituntut untuk melakukan penandaan dengan teliti, harus diperhatikan *margin* pemotongan agar pada saat dilakukan pemotongan dengan *gas cutting*, *kerf* yang terjadi tidak menyebabkan terjadinya penyusutan pada pelat. *Kerf* adalah kehilangan sebagian pelat karena proses pemotongan, dimana besarnya *kerf* tergantung dari tebal pelat, alat yang digunakan dan operator.

## 2. Penyetelan *Nozzel*

Penyetelan *nozzel* yang tidak benar dapat menyebabkan terjadinya distorsi pelat. Untuk menghindari adanya distorsi akibat dari pemotongan harus diperhatikan tinggi *nozzel* dan penyetelan api yang tepat serta kecepatan *torch*. Distorsi pelat akibat pemotongan dengan menggunakan *gas cutting* disebabkan adanya *internal deformation* karena proses pemanasan terpusat dan *external deformation* karena adanya penahanan dan penekanan.

## 3. Teknik Pemotongan

Apabila operator dalam melakukan pemotongan pelat kurang memperhatikan teknik pemotongan yang benar dapat menyebabkan terjadinya distorsi. Distorsi ini terjadi apabila salah satu ujung dari pelat yang dipotong diklem dan bagian yang tidak diklem akan bergerak ke arah penahannya sehingga menyebabkan terjadinya distorsi. Untuk menghindari hal tersebut perlu adanya urutan pemotongan yang benar serta memberikan penahanan berlawanan dengan arah deformasi.

### □ **Deformasi penyusutan karena material**

Deformasi penyusutan berdasarkan kondisi dari material dipengaruhi oleh beberapa faktor yaitu :

#### 1. Keadaan Fisik Material Pelat

Bentuk fisik material yang berupa kelengkungan, laminasi, cacat dan lain-lain akan berpengaruh pada ketelitian proses marking, cutting dan welding.

## 2. Sifat Material Pelat

Harga koefisien muai, koefisien perambatan panas, kadar karbon dan daya elastisitas baja merupakan sifat dari material yang berpengaruh pada deformasi penyusutan. Harga koefisien muai semakin besar maka semakin besar kemungkinan terjadinya distorsi. Harga koefisien muai ini semakin besar dengan adanya pertambahan suhu, sehingga penyusutan yang diakibatkan oleh adanya distorsi akan semakin besar.

## 3. Material Las

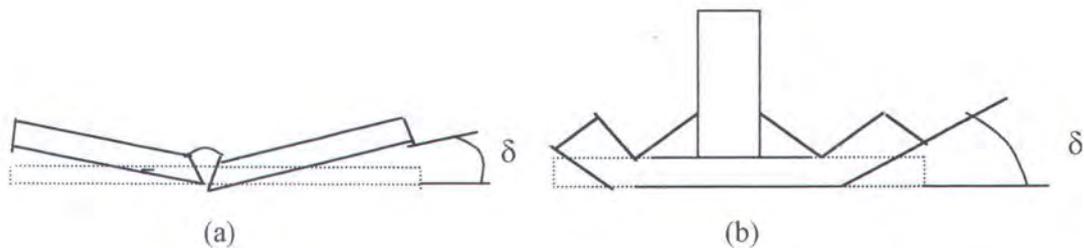
Material las mempengaruhi deformasi penyusutan pada sambungan las, sehingga perlu diperhatikan hal-hal sebagai berikut :

- Ukuran elektrode, dimana diameter yang besar akan memberikan penyusutan yang lebih besar, tetapi hal ini tidak selamanya berlaku.
- Jenis flux, dimana jenis flux ini akan mempengaruhi kecepatan perpindahan logam las ke base metal dan heat input. Dengan adanya perbedaan jenis flux akan menyebabkan terjadinya perbedaan besar penyusutan.

### III.2.2. PENYIMPANGAN BENTUK KARENA DEFORMASI SUDUT

Deformasi sudut dapat terjadi apabila pada dua permukaan pelat terdapat terdapat penyusutan atau pengembangan. Deformasi sudut akan membuat konstruksi yang direncanakan berubah dari bentuk dan ukurannya. Yang dimaksud dengan perubahan bentuk dan ukuran di sini adalah perubahan sudut pada konstruksi sambungan las.

Deformasi sudut meliputi dua macam bentuk yaitu deformasi sudut yang terjadi pada konstruksi sambungan las *butt joint* (las tumpul) dan konstruksi sambungan las *fillet* (las T). Deformasi sudut pada konstruksi sambungan las *fillet* lebih besar dari pada konstruksi sambungan las *butt joint*, dimana deformasi sudut pada sambungan las *fillet* besarnya dua kali deformasi sudut pada sambungan las *butt joint*. Deformasi sudut tersebut dapat digambarkan seperti pada gambar dibawah ini :



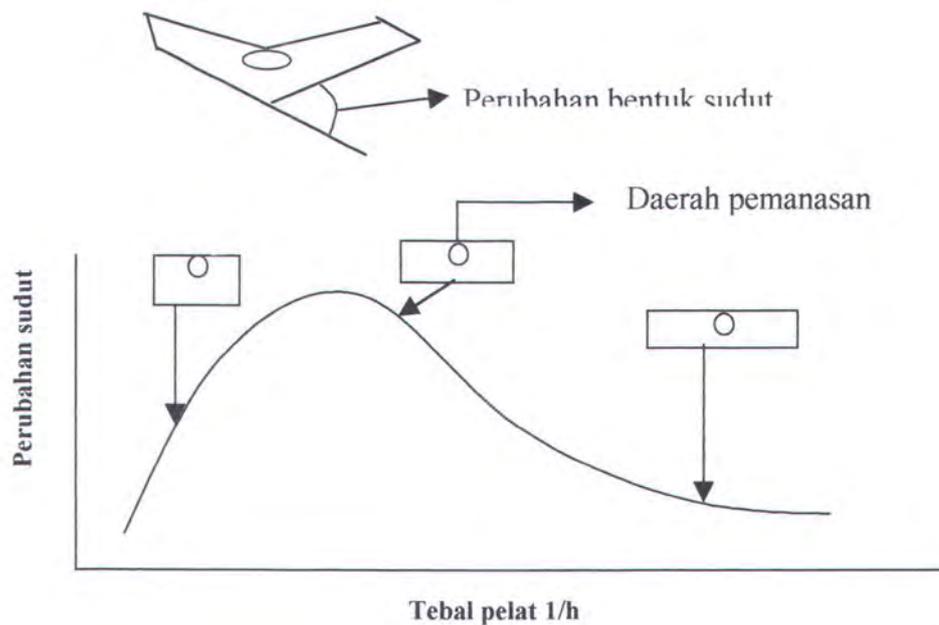
Gambar 3.2. Deformasi sudut sambungan *butt joint* (a) dan *fillet* (b)

Proses pengelasan merupakan suatu pengerjaan terhadap material pelat yang tidak mungkin untuk dihindari. Penyambungan pelat menggunakan sistem keling sudah tidak populer lagi karena membutuhkan waktu yang sangat lama dan hasil penyambungannya tidak sebaik pada penyambungan pelat menggunakan pengelasan. Namun demikian tidak berarti pada penyambungan pelat dengan pengelasan pelat tidak mengalami kerugian. Kerugian tersebut antara lain menyebabkan deformasi sudut pada daerah sambungan las.

Proses pengelasan erat kaitannya dengan pekerjaan *fitting* dan sekaligus memperkirakan deformasi sudut yang terjadi. Faktor-faktor yang menyebabkan deformasi sudut meliputi :

### 1. Tebal Material Pelat

Ketebalan material pelat mempengaruhi besar kecilnya deformasi sudut yang diakibatkan proses pengelasan. Diagram skematik hubungan antara deformasi sudut dengan ketebalan pelat pada kondisi las tetap dapat digambarkan seperti pada grafik di bawah ini :

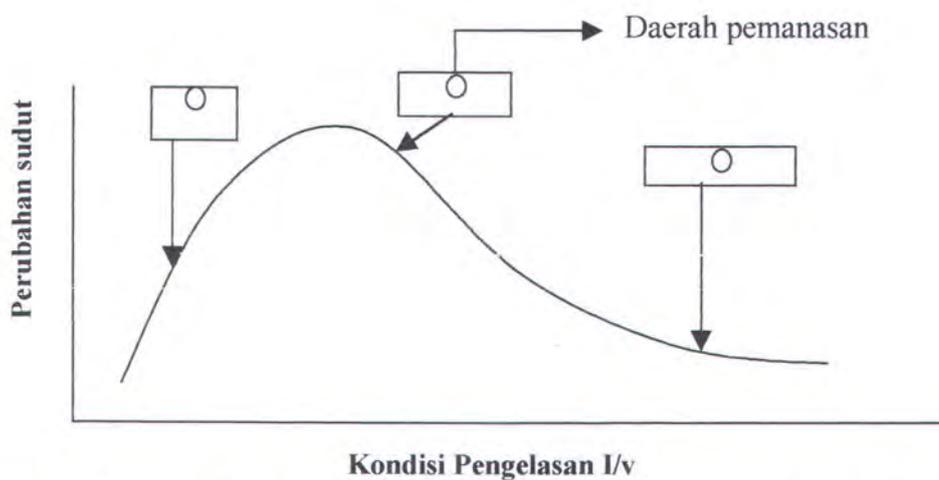


Gambar 3.3. Grafik hubungan antara perubahan sudut dengan tebal pelat

Pada pengelasan material pelat yang tipis perubahan sudut yang terjadi kecil, disebabkan bentuk weld metal mendekati simetri. Semakin tebal pelat maka deformasi sudut yang terjadi semakin besar, dikarenakan bentuk *weld metal* menjauhi bentuk simetri sehingga penyusutan atau pengembangan antara dua permukaan material akan semakin besar perbedaannya. Tetapi pada ketebalan pelat tertentu deformasi sudut akan yang terjadi akan semakin kecil, disebabkan adanya penahanan dari material pelat yang semakin besar dengan penambahan ketebalan pelat.

## 2. Kondisi Pengelasan

Kondisi pengelasan yang dimaksud adalah mengenai besarnya arus pengelasan, kecepatan pengelasan dan ketebalan material pelat. Diagram skematik hubungan antara kondisi pengelasan dalam batang uji manik dengan diameter elektrode 4 mm dan 6 mm dapat digambarkan seperti grafik dibawah ini :



Gambar 3.4. Grafik hubungan antara perubahan sudut dengan kondisi pengelasan

Pengaruh arus dan kecepatan pengelasan terhadap perubahan sudut adalah jumlah dan bentuk dari *weld metal* yang besar akan membuat bentuk *grove* mendekati simetri, sehingga perubahan sudut yang terjadi semakin besar. Pada kondisi tertentu sudut akan mengecil, disebabkan adanya penahanan dari pelat yang semakin besar dengan semakin besarnya tebal pelat.

## 3. Bentuk Alur

Pada proses pengelasan mengenal dua jenis bentuk alur pengelasan yaitu alur pengelasan bentuk *V* dan bentuk *X*. Kedua bentuk alur pengelasan

tersebut menyebabkan terjadinya deformasi sudut, dimana besar deformasi sudut yang disebabkan oleh alur pengelasan bentuk *V* lebih besar dibanding deformasi yang disebabkan oleh alur pengelasan bentuk *X*.

Pada alur bentuk *V* deformasi sudutnya lebih besar disebabkan adanya penyusutan dari weld metal pada kedua permukaan material pelat yang tidak sama. Sedangkan pada alur bentuk *X* deformasi sudutnya lebih kecil disebabkan penyusutan dari weld metal pada kedua permukaan material pelat hampir sama. Pada bentuk alur pengelasan yang semakin tidak simetri akan terjadi perbedaan penyusutan yang semakin besar, sehingga perubahan sudut yang terjadi semakin besar pula.

#### 4. Jumlah Lapisan

Jumlah lapisan pengelasan padangan sambungan las *fillet* mempengaruhi besarnya deformasi sudut. Perubahan sudut pada sambungan las *fillet* akan bertambah besar dengan semakin banyaknya jumlah *pass*. Hal tersebut disebabkan dengan semakin banyaknya *pass* maka semakin besar gaya yang timbul akibat penyusutan dari *weld metal*. Pada jarak yang semakin jauh dari pengelasan, perubahan sudut (rad) akan semakin kecil karena pengaruh dari *weld metal* juga semakin kecil.

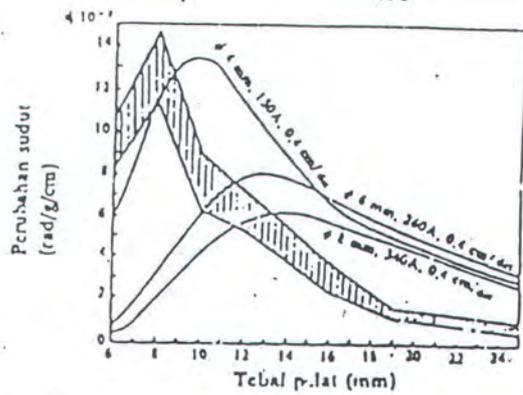
#### 5. Berat logam las

Berat logam las pada pengelasan akan berpengaruh pada deformasi sudut. Semakin besar berat logam las maka perubahan sudut (rad) akan semakin besar. Gaya penyusutan yang mengakibatkan terjadinya perubahan sudut

akan semakin besar dengan semakin bertambah beratnya logam las (weld metal).

6. Prosedur pengelasan

Prosedur pengelasan yang dimaksud adalah kondisi pengelasan dan tebal pelat yang digunakan. Secara diagram hubungan antara prosedur pengelasan dan perubahan sudut dapat dilihat pada gambar di bawah ini.



Gambar 3.5 Grafik hubungan antara prosedur pengelasan dan perubahan sudut

Diagram di atas dapat digunakan untuk memilih metode pengelasan yang sesuai agar perubahan sudut yang terjadi seminim mungkin. Dari diagram tersebut dapat dilihat bahwa tebal pelat kurang dari 10 mm, las busur listrik tangan dengan elektrode yang besar memberikan perubahan sudut yang lebih kecil dari pada yang terjadi karena busur redam. Tetapi bila material pelat yang dilas memiliki ketebalan lebih dari 12 mm maka pengelasan menggunakan busur redam akan memberikan perubahan sudut yang lebih kecil.

### III.2.3. PENYIMPANGAN BENTUK KARENA DEFORMASI MEMANJANG

Terjadinya deformasi memanjang disebabkan proses pengerjaan pada material pelat meliputi pemotongan, pengelasan dan gaya luar.

#### □ Deformasi memanjang karena proses pemotongan

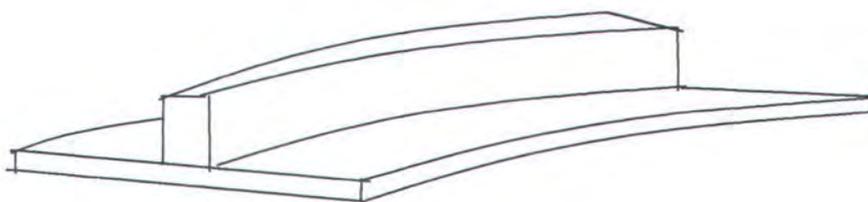
Pemotongan material dapat dilakukan dengan dua cara yaitu secara mekanis dan secara termal. Secara mekanis dapat dilakukan dengan menggunakan gunting. Pada pemotongan dengan menggunakan gunting tersebut pada bagian ujung-ujung pelat yang dengan dengan daerah pemotongan sering timbul deformasi. Akan tetapi deformasi tersebut sifatnya kecil sehingga mudah diperbaiki dengan cara pengerollan atau press. Sedangkan pemotongan secara termal dapat dilakukan dengan menggunakan gas *cutting*. Pada pemotongan termal akan menyebabkan material menerima suhu yang tidak merata, sehingga distribusi panas yang tidak merata akan menimbulkan tegangan dan perubahan bentuk.

Sifat deformasi karena proses pemotongan pada bermacam-macam ketebalan pelat adalah sama yaitu semakin besar lebar pelat yang dipotong maka deformasi yang ditimbulkan akan semakin kecil, hal tersebut disebabkan adanya penahanan dari material yang semakin besar.

#### □ Deformasi memanjang karena pengelasan

Deformasi memanjang karena pengelasan penyebabnya sama dengan deformasi karena pemotongan dengan termal yaitu distribusi panas yang tidak merata. Pada proses pengelasan pada bagian yang dilas menerima panas setempat

dan selama proses pengelasan berlangsung terjadilah distribusi panas yang tidak merata. Pada bagian yang dilas, karena adanya panas maka pada bagian ini akan mengalami pengembangan termal, sedang bagian yang lain suhunya tidak berubah sehingga seolah-olah terjadi penghalang peregangan sehingga terjadilah deformasi.



Gambar 3.6 Deformasi memanjang karena pengelasan

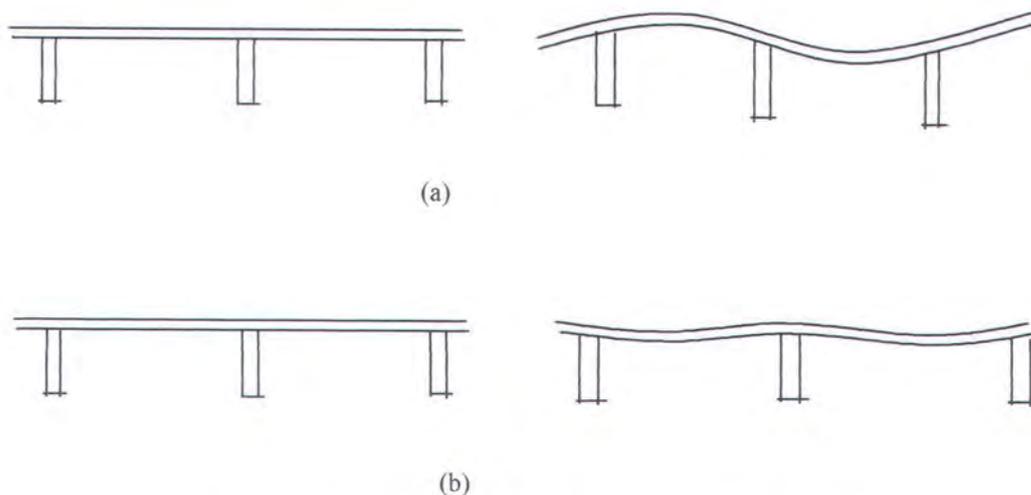
#### □ Deformasi memanjang karena gaya luar

Deformasi ini sering terjadi pada saat pengangkatan material pelat, misalnya pengangkatan material pelat menggunakan crane. Terutama pada pengangkutan bentuk seksi-seksi atau blok-blok yang berat, letak *lifting piece* harus tepat. Terjadinya deformasi ini disebabkan karena adanya gaya luar yaitu gaya berat dari konstruksi tersebut, gaya tarik (gaya dari proses pengangkutan), gaya geser dan lain-lain.

### III.2.4. PENYIMPANGAN BENTUK KARENA DEFORMASI FLEXURE DAN SHRINKAGE

Deformasi jenis ini sering terjadi pada konstruksi lambung kapal. Deformasi flexure adalah deformasi berupa perubahan bentuk melengkung dari

sebagian konstruksi las-lasannya dimana posisi dari penguatnya tidak mengalami perubahan. Sedangkan deformasi shrinkage adalah deformasi berupa perubahan bentuk melegkung dari sebagian konstruksi las-lasannya dimana posisi dari penguatnya ikut mengalami perubahan. Penyebab dari deformasi ini adalah distribusi panas yang tidak merata pada saat pengelasan.

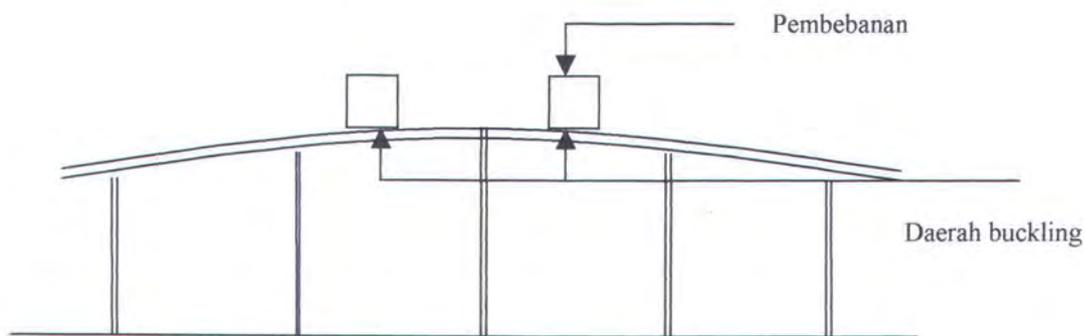


Gambar 3.7 Deformasi flexure (a) dan deformasi shrinkage (b)

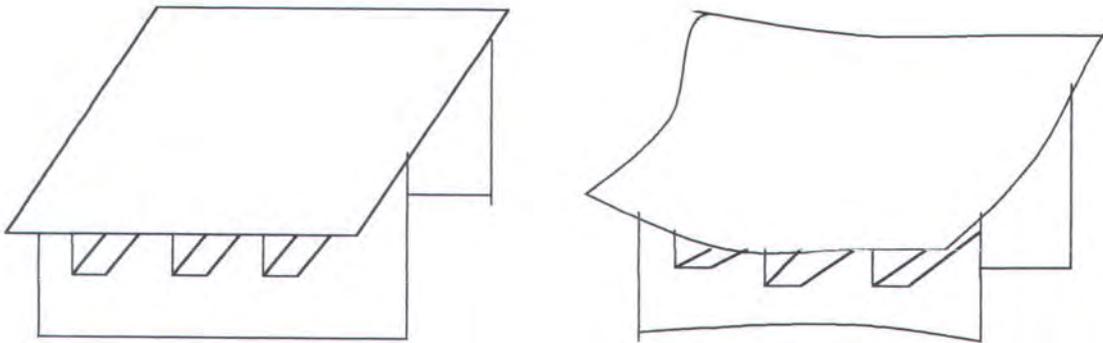
### III.2.5. PENYIMPANGAN BENTUK KARENA DEFORMASI BUCKLING

Deformasi buckling adalah deformasi yang berupa perubahan bentuk melengkung dari konstruksi suatu blok yang disebabkan oleh gaya luar (misalnya karena kejatuhan benda lain yang berat, penyimpanan waktu assembling yang kurang baik dan lain-lain). Namun demikian deformasi buckling dapat juga terjadi karena pengaruh termal (panas) pada saat proses pemotongan dan pengelasan.

Dengan adanya proses pemotongan dan pengelasan yang menghasilkan panas tidak merata menyebabkan timbulnya tegangan sisa. Adanya tegangan sisa pada konstruksi akan mempermudah terjadinya deformasi *buckling* apabila pada konstruksi tersebut menerima beban. Hal ini akan semakin cenderung terjadinya *buckling* apabila pengelasan dilakukan pada pelat yang panjangnya melampaui panjang yang telah ditentukan untuk tebal pelat yang diberikan. Untuk mengantisipasi agar tidak terjadi deformasi *buckling* dilakukan pembebanan atau penahanan yang diberikan saat proses pengelasan pada bagian-bagian yang panjang. Deformasi *buckling* mempunyai bentuk fisik yang lebih dari satu tempat permukaan dan mempunyai kecenderungan terjadinya *buckling* pada pelat yang mempunyai ketebalan yang tipis.



Gambar 3.8 Pembebanan pada proses pengelasan



Gambar 3.9 Deformasi buckling

Deformasi buckling ini sering terjadi pada pembangunan kapal dengan metode blok. Dengan demikian perlu sekali diperhatikan hal-hal tersebut agar deformasi yang ditimbulkan tidak mengganggu proses penggabungan blok.

### III.3. JENIS PERBAIKAN PENYIMPANGAN BENTUK DEFORMASI.

Seperti telah dijelaskan dimuka, deformasi yang terjadi pada suatu konstruksi akan mengurangi ketelitian dimensi (ukuran), keindahan dan kekuatan daripada konstruksi kapal tersebut. Apabila deformasi ini telah terjadi maka untuk memperbaiki kembali diperlukan waktu dan tenaga yang cukup banyak. Oleh karena itu deformasi sedapat mungkin dihindari sejak awal yaitu sebelum proses pengerjaan terhadap material tersebut dilaksanakan.

#### III.3.1. PENAGGULANGAN PENYIMPANGAN BENTUK DEFORMASI

Penyimpangan bentuk yang terjadi dalam pengelasan tidak hanya mengurangi ketelitian ukuran dan penampakkannya dari luarnya saja tetapi juga menurunkan kekuatannya. Bila penyimpangan bentuk ini terjadi, untuk

meluruskannya kembali diperlukan waktu dan kerja yang cukup banyak, karena itu sedapat mungkin harus dihindari dengan menentukan prosedurnya terlebih dahulu sebelum pelaksanaan pengelasannya.

Cara penanggulangan penyimpangan bentuk deformasi karena proses pengelasan dibagi menjadi dua yaitu sebelum pengelasan dan sesudah pengelasan.

- Sebelum proses pengelasan dilakukan, yang meliputi :
  - Meluruskan terlebih dahulu bagian yang dilas, sesuai dengan dimensi dan bentuk yang telah ditentukan.
  - Pemilihan elektrode yang sesuai dengan material yang akan dilas.
  - Pemilihan bentuk kampuh yang sesuai.
  - Pemilihan prosedur pengelasan yang sesuai.
  - Pemakaian tukang las yang benar-benar bermutu.
- Pada proses pengelasan dilaksanakan, pekerjaan ini meliputi :
  - Mengurangi masukan panas.
  - Mengurangi logam las, yaitu dengan cara mengurangi panjang las-lasan dan jumlah lapisan las-lasan serta memilih bentuk alur las yang sesuai.
  - Menentukan urutan pengelasan yang tepat.
  - Memberikan penahan pada bagian yang akan dilas.
  - Memberikan perubahan bentuk awal yang berlawanan arah dengan arah deformasi .

### III.3.2. PERBAIKAN PENYIMPANGAN BENTUK DEFORMASI

Dasar-dasar dalam usaha perbaikan penyimpangan bentuk dalam pengelasan adalah memanjangkan bagian yang menyusut dan menyusutkan bagian yang mengembang. Secara garis besar cara perbaikan ini dapat dibagi menjadi dua kelompok yaitu perbaikan termal dan perbaikan mekanik. Dalam perbaikan termal hal yang dilakukan adalah pemanasan dan pendinginan, sedangkan perbaikan mekanik yang dilakukan adalah pengerolan, penekanan dan penempaan atau pemukulan.

Perbaikan deformasi yang terjadi pada konstruksi lambung kapal umumnya menggunakan cara pemanasan atau lebih dikenal dengan nama *fairing*. Karena dengan pertimbangan mudah dilaksanakan dan hasilnya lebih baik dibanding dengan cara mekanis serta efek kerusakan yang ditimbulkan pada material kecil selama pekerjaan *fairing* dilaksanakan dengan prosedur yang benar.

#### III.3.2.1. PERBAIKAN DEFORMASI DENGAN CARA MEKANIK

Perbaikan material yang mengalami penyimpangan bentuk dengan cara mekanik pada dasarnya adalah perbaikan dingin yang dilakukan pada temperatur ruang dan proses ini pelaksanaannya dilakukan dengan menggunakan alat-alat mekanik, misalnya dongkrak, palu, rol dan lain sebagainya. Perbaikan dengan cara mekanik selain pelaksanaannya lama dan sulit, hasilnya kurang baik. Oleh karena itu untuk mendapatkan hasil perbaikan yang lebih baik dan efektif, pelaksanaannya dilakukan bersamaan dengan pemanasan dan biasanya disebut perbaikan dengan pemanasan dan penekanan.

Perbaikan mekanik ini biasanya dilakukan pada proses fabrikasi, dimana bentuk deformasi yang sering muncul masih mampu diperbaiki dengan cara mekanik (dingin). Perbaikan dengan cara mekanik selain pelaksanaannya lama dan sulit, hasilnya kurang baik.

### III.3.2.2. PERBAIKAN DEFORMASI DENGAN CARA PEMANASAN

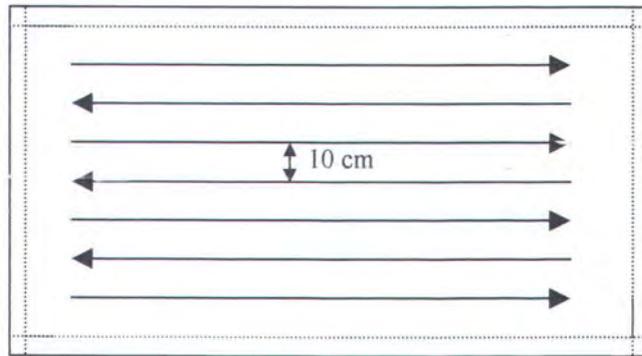
Untuk memperbaiki deformasi yang terjadi dilakukan pemanasan dan kemudian didinginkan dengan cara tiba-tiba. Caranya adalah dengan jalan memberi pemanasan pada daerah yang mengalami deformasi dengan temperatur tertentu, kemudian baru didinginkan dengan air secara mendadak. Macam-macam pemanasan untuk perbaikan deformasi meliputi :

□ **Line heating (pemanasan garis)**

Line heating adalah pemanasan dengan bentuk alur seperti garis lurus. Hampir 90 % perbaikan deformasi material pelat memakai cara ini, karena hasilnya paling bagus. Pemanasan garis banyak dikerjakan pada daerah gading-gading, ataupun pada material pelat yang mengalami deformasi dengan ketebalan yang cukup besar. Model-model dari pemanasan garis :

• *Line heating memanjang*

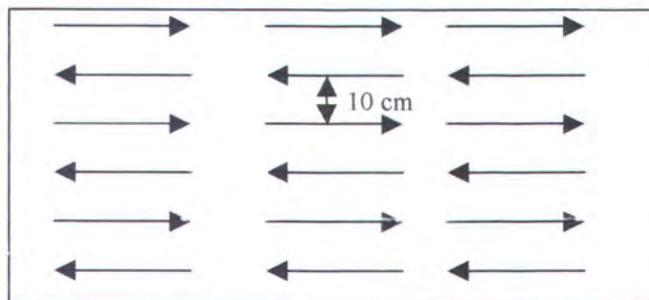
Cara ini dipergunakan untuk memperbaiki deformasi yang besar dan arah deformasinya keluar.



Gambar 3.10. Line heating memanjang

- Line heating pendek-pendek

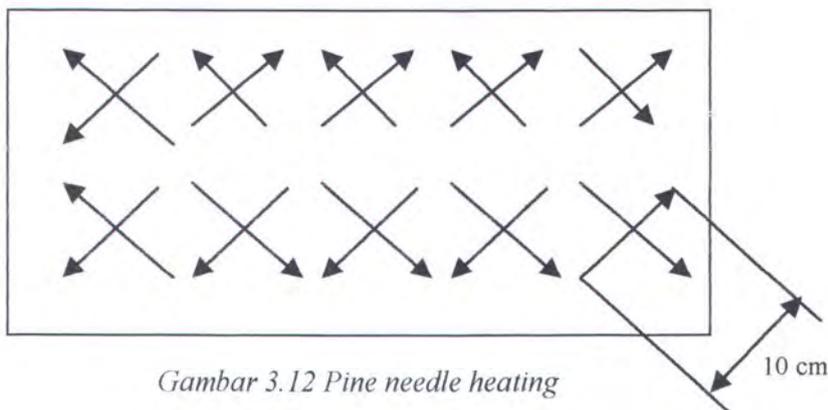
Cara ini dipergunakan untuk memperbaiki deformasi yang kecil.



Gambar 3.11 Line heating pendek-pendek

- Pine needle heating (Pemanasan cemara)

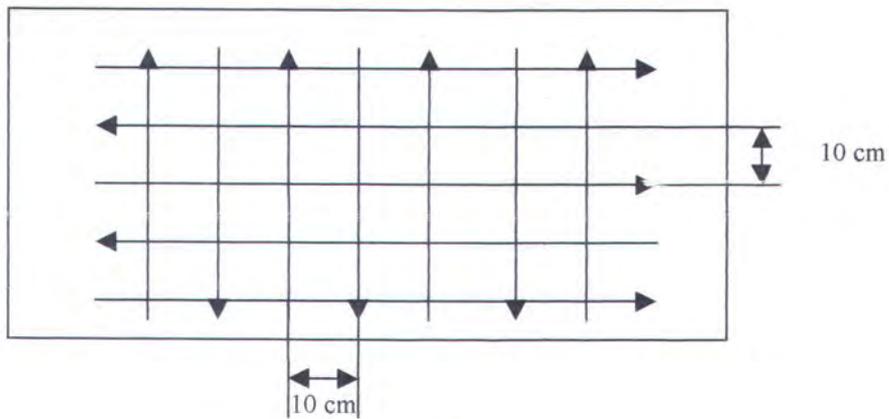
Cara ini digunakan untuk menghilangkan deformasi yang kecil .



Gambar 3.12 Pine needle heating

- *Lattice heating* (pemanasan kisi-kisi)

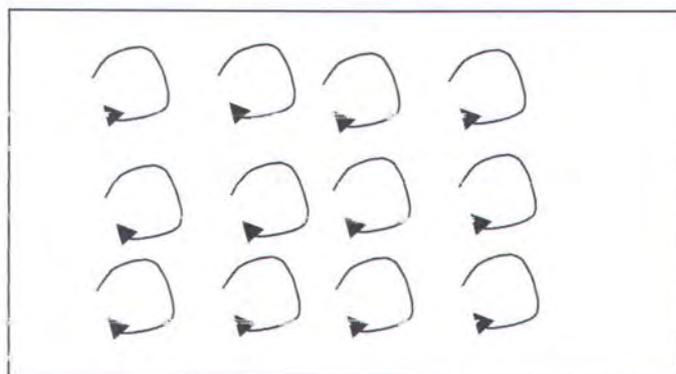
Cara ini dipergunakan untuk menghilangkan deformasi yang besar. Hasil perbaikan baik tetapi over heating sehingga dapat merusak komposisi kimia pelat.



Gambar 3.13 Lattice heating

- *Ring heating* (pemanasan cicin)

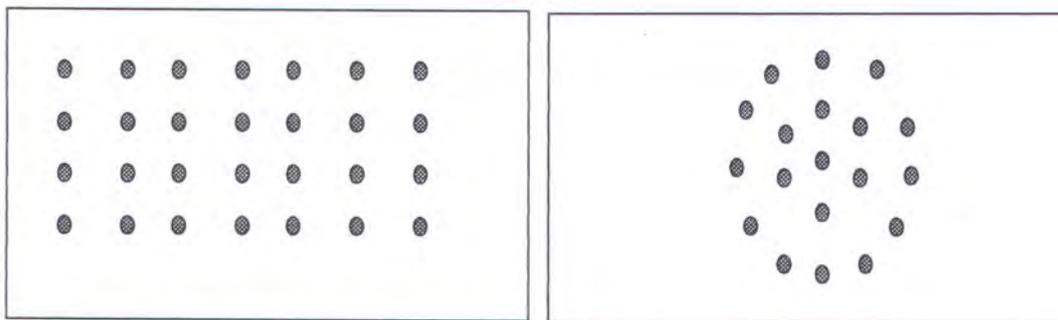
Cara ini digunakan untuk menghilangkan deformasi yang terjadi pada sambungan las tumpul (butt joint).



Gambar 3.14 Ring heating

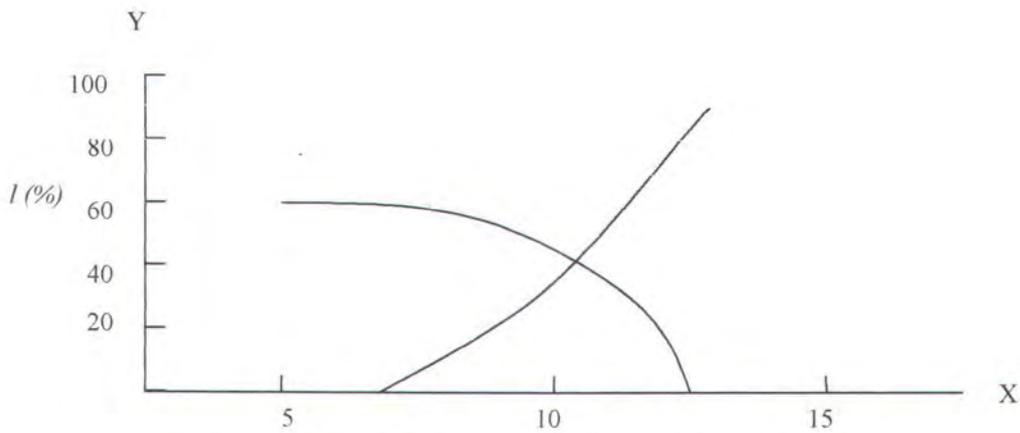
□ **Spot Heating (pemanasan titik)**

*Spot heating* adalah pemanasan setempat berupa titik-titik. Cara ini tidak dapat dipakai secara terpisah dari cara *line heating*, sebab akan menyebabkan terjadinya pengkerutan yang besar. Disamping itu *spot heating* hanya bisa dipergunakan untuk memperbaiki deformasi yang terjadi pada material pelat-pelat tipis saja, yaitu untuk daerah pelat antara dua gading. Model pemanasan titik meliputi susunan titik-titik yang lurus dan lingkaran.



Gambar 3.15 Model spot heating

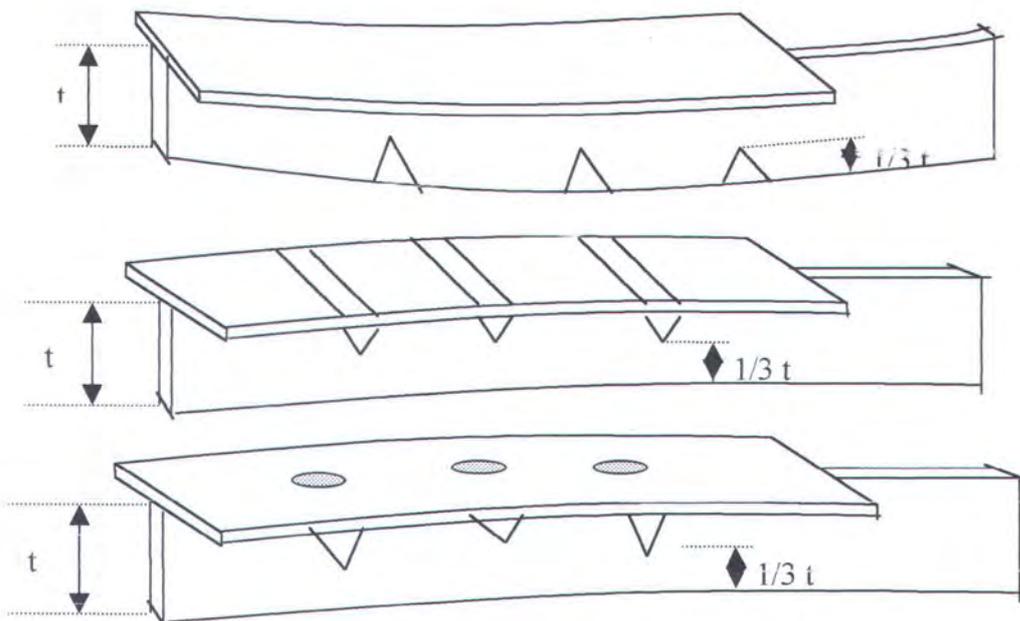
Apabila *line heating* dan *spot heating* digunakan secara terpisah maka hasil yang diperoleh dari pemakaian *line heating* lebih baik hasilnya jika dibanding dengan menggunakan *spot heating*. Pemilihan *line heating* atau *spot heating* juga dapat dilakukan berdasarkan tebal pelatnya, seperti ditunjukkan oleh grafik di bawah ini :



Gambar 3.16 Grafik hubungan spot heating dengan line heating

□ **Triangle heating**

*Triangle heating* adalah pemanasan dengan bentuk alur segitiga. Cara ini baru dilakukan setelah cara *line heating* dapat dikuasai dengan baik. Cara ini dipakai untuk memperbaiki deformasi memanjang pada pembuatan bentuk profil T dan akan berhasil lebih baik lagi apabila diberi bantuan gaya luar. Untuk perbaikan tempat-tempat yang tidak bisa memakai bantuan gaya luar ataupun pendinginan dengan air, dilakukan cara ini dua kali pemanasan dan dengan cara pendinginan udara.



Gambar 3.17 Jenis-jenis pemanasan segitiga

□ **Red Heating**

*Red heating* adalah pemanasan yang dilakukan pada permukaan material pelat, sampai keadaan pelatnya menjadi merah membara. Kemudian baru diberi bantuan tenaga dari luar, baik dengan pemukulan atau dengan tarikan. Cara ini dipergunakan untuk memperbaiki deformasi yang besar, atau untuk memperbaiki deformasi yang tidak bisa diperbaiki dengan *line heating*.

Perbaikan dengan menggunakan cara pemanasan khususnya untuk *line heating* telah mendapat persetujuan dari ABS, LR dan biro klasifikasi lain sebagai metode yang layak untuk melakukan pembendingan pelat-pelat kapal dan bagian-bagian lain, misalnya penegar-penegar serta untuk *fairing* (pelurusan) pada bagian yang mengalami deformasi. Dalam hubungannya dengan pemakaian *line heating* di galangan-galangan, ada beberapa persyaratan-persyaratan tertentu yang tidak boleh diabaikan seperti pembatasan suhu, prosedur pelaksanaan *line heating* untuk baja-baja high tensile serta untuk pelat dengan bermacam-macam ketebalan.

Batas suhu yang dipakai oleh galangan-galangan kapal di luar negeri yang secara teratur menggunakan *line heating* selama tahun 1981 dapat dilihat dalam gambar di bawah ini.

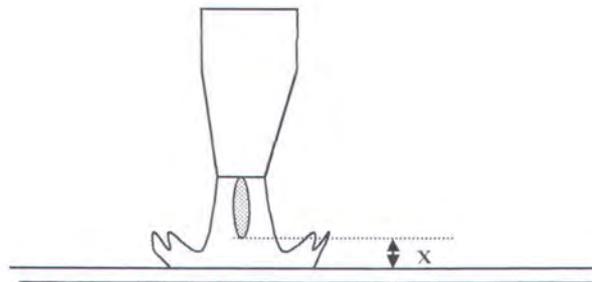
Material	Maximum permissible heating temperature	Maximum temperature for water cooling	Remarks
Mild steel	900 °C	850 °C	Air cooling and water cooling
High tensile steel	900 °C	--	Air cooling
	900 °C	500 °C	Water cooling
	650 °C	650 °C	Water cooling

Gambar 3.18 Batas suhu yang dipakai untuk *line heating*

Oleh karena itu harus dijaga jarak antara material pelat dengan brander agar suhu yang masuk dapat konstan. Jarak pemanasan nyala api dari brander terhadap pelat dapat dilihat dari tabel dibawah ini :

Tebal pelat (mm)		Jarak pemanasan: x (mm)	
3	4,5	-2	0
6	8	0	
10	14	0	3
16	22	3	4
24	28	4	5
30		6	10

Tabel 3.3 Hubungan tebal pelat dengan jarak nyala api



Gambar 3.19 Jarak nyala api brander dengan pelat

Selain jarak nyala api, ada hal lain yang harus diperhatikan yaitu masalah kecepatan pemanasan dan ukuran *nozzle* yang dipakai. Di bawah ini dapat dilihat tabel hubungan antara kecepatan *nozzle* dan nomor brander serta tebal pelat:

Tebal pelat (mm)		No. Brander	Kecepatan pemanasan (mm/min)	
3	4,5	0,500	0,800	1,500
5	8	1,000	0,700	1,000
9	12,7	1,600	0,500	1,000
13	16	2,000	0,400	0,800
17	22	2,500	0,350	0,800
23	28	3,150	0,300	0,600
29		3,500	0,250	0,500

Tabel 3.4 Hubungan kecepatan nozzle dan no brander serta tebal pelat

Gas yang dipakai adalah gas O<sub>2</sub> dan *acetyline* dengan perbandingan 1 : 1 atau 1 : 1,5. Untuk cara pendinginannya dilaksanakan segera setelah pemanasan, sehingga seolah-olah dilakukan secara bersama-sama.

Metode line heating telah diijinkan untuk semua grade baja ABS dan yang telah direkomendasikan oleh ABS. Sedangkan batas suhu untuk line heating yang dipakai oleh galangan-galangan kapal di Jepang adalah sebagai berikut :

- Suhu pemanasan dapat dinaikkan sampai 800° C – 900° C jika sesudah pemanasan didinginkan dengan angin/udara.
- Suhu pemanasan tidak boleh melebihi 600° C – 650° C jika sesudah pemanasan langsung didinginkan dengan air.
- Suhu pemanasan dapat dinaikkan sampai 800° C - 900° C jika dipakai pendinginan air setelah selang waktu tertentu didinginkan dengan udara.

Pendinginan dengan air tersebut dilakukan setelah suhu dibawah 500° C.

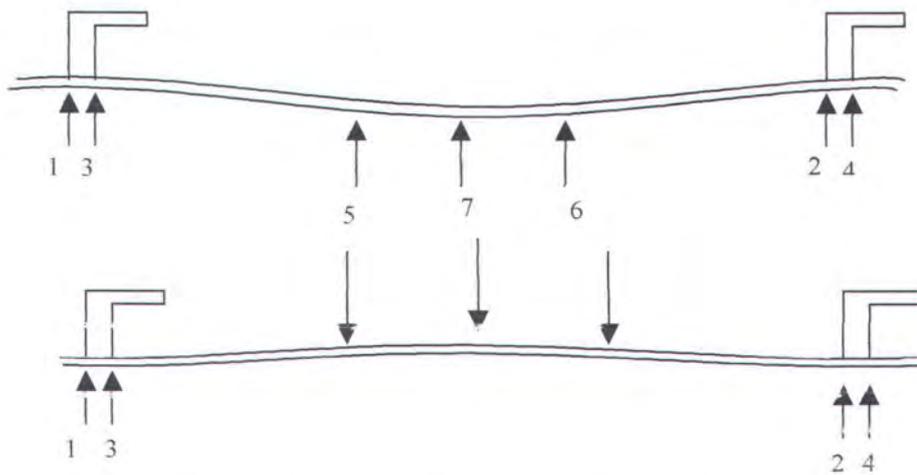
Sedangkan posisi untuk melakukan pendinginan ada dua macam yang dapat dilakukan yaitu :

- Sama dengan posisi pemanasan, bila udara ditempat tersebut dapat mengalir dengan baik.
- Berlawanan dengan posisi pemanasan, bila udara tidak bisa mengalir dengan baik.

### III.4. PERBAIKAN DEFORMASI PADA KONSTRUKSI LAMBUNG KAPAL

Pada umumnya di galangan-galangan besar untuk melakukan perbaikan deformasi menggunakan cara pemanasan. Karena efektifitasnya tinggi dan hasilnya cukup baik jika dilakukan dengan prosedur yang benar.

1. Pemilihan pemanasan berdasarkan ketebalan pelat
  - a. Ketebalan pelat lebih besar dari 10 mm (pelat  $\geq 10$  mm)
    - Check lokasi dari penguat dan beri tanda yang sama pada pelatnya, dibalik penguatnya.
    - Panasi belakang permukaan las (2 lines) atau ditengah-tengah penguatnya (1 line).
    - Panasi bagian pelat yang melengkung berdasarkan panjang deformasi.
    - Temperatur pemanasan  $\pm 900^{\circ}$  C kemudian didinginkan menggunakan air setelah suhunya menurun menjadi  $500^{\circ}$  C.



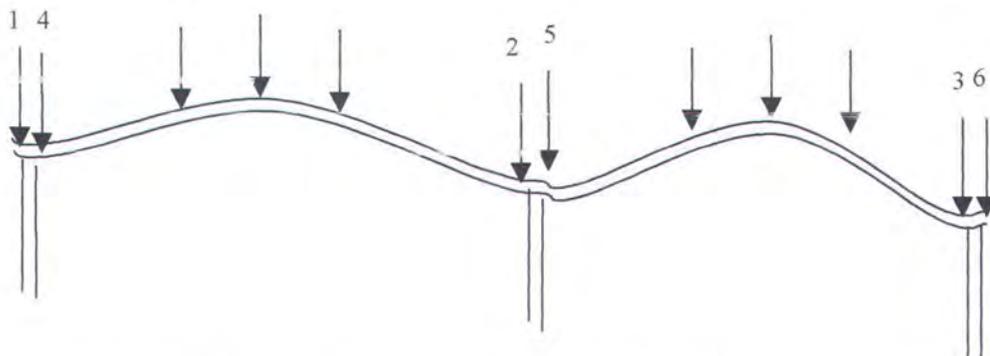
Gambar 3.20 Perbaikan deformasi pelat antara 2 gading dengan line heating

*Keterangan gambar :*

Angka-angka menunjukkan urutan pengerjaan pemanasan pada daerah gading-gadingnya.

b. Ketebalan pelat lebih besar dari 10 mm (pelat  $\leq$  10 mm)

- Check lokasi dari penguat dan beri tanda yang sama pada pelatnya, dibalik penguatnya.
- Panasi belakang permukaan las (2 lines) atau di tengah-tengah penguatnya (1 line).
- Panasi bagian pelat yang cembung, kemudian panasi dan luruskan daerah yang melekuk. Sebab bila pekerjaan distorsi pada daerah-daerah yang cembung selesai, pekerjaan distorsi dari daerah yang melekuk telah berubah sendiri, kira-kira setengah akibat pekerjaan distorsi pada daerah yang cembung tadi.
- Temperatur pemanasan  $\pm 650^{\circ}$  C kemudian dinginkan menggunakan air dengan cepat sebagaimana memanasi dengan pelat.



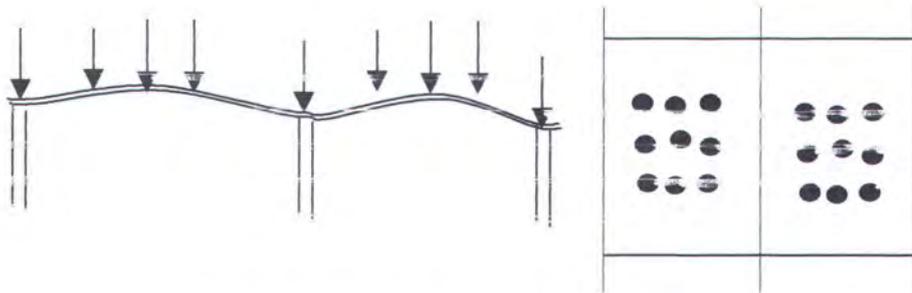
Gambar 3.21 Perbaikan deformasi pelat antara 2 gading dengan line heating

Keterangan gambar :

Angka-angka menunjukkan urutan pengerjaan pemanasan pada daerah gading-gadingnya.

c. Ketebalan pelat lebih kecil dari 5 mm (pelat  $\leq 5$ mm)

- Check lokasi yang mengalami deformasi.
- Panasi belakang permukaan las (2 lines) atau di tengah-tengah penguatnya (1 line).
- Panasi bagian yang mengalami deformasi dengan menggunakan pemanasan titik (spot heating).
- Temperatur pemanasan  $650^{\circ}$  C kemudian dinginkan menggunakan air dengan cepat sebagaimana memanasi pelat.

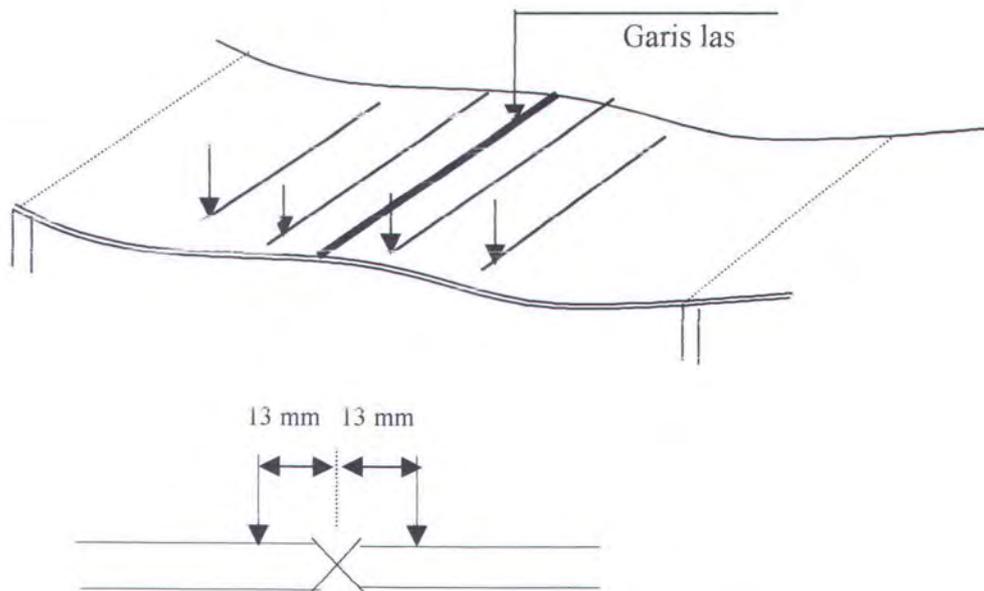


Gambar3.22 Pemanasan setempat (spot heating)

2. Deformasi pada sambungan las tumpul

- Check lokasi yang mengalami deformasi.
- Panasi belakang permukaan las (2 lines) atau ditengah-tengah penguatnya (1 line).
- Panasi sebelah kanan dan kiri dari garis las sejarak 13 mm.

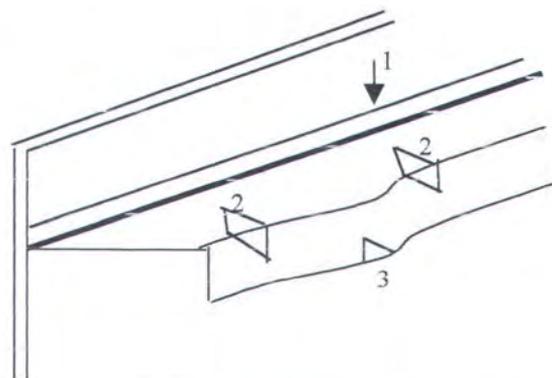
- Pemanasan bagian yang mengalami deformasi dengan cara pemanasan dan temperatur pemanasan sesuai dengan ketebalan pelat.
- Pendinginan disesuaikan dengan temperatur pemanasan.



Gambar 3.23 Pemanasan pada sambungan las tumpul

### 3. Deformasi pada penguat dalam

Deformasi yang meliputi pada bagian gading memanjang, balok, penegar dan lain-lain. Dilakukan pengecekan kelurusan pada penguat dan lakukan pekerjaan-pekerjaan seperti dibawah ini :



Gambar 3.24 Pemanasan pada penguat dalam

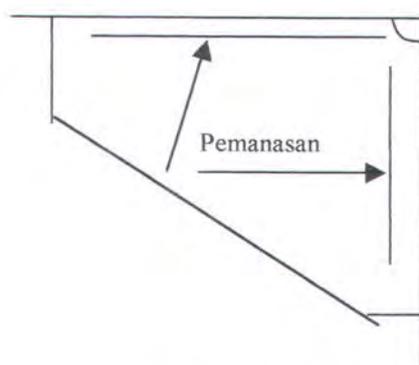
*Keterangan gambar :*

Angka-angka menunjukkan urutan pengerjaan pemanasan pada daerah yang mengalami deformasi.

- Panasi pelat kulit atau pelat sekat  $\pm 5$  mm di atas permukaan las atau  $\pm 5$  mm di bawah permukaan las penegar.
- Panasi sisi-sisi bagian penguat yang kena tekukan besar dengan pemanasan segitiga.
- Pemanasan bagian penguat yang kena tekukan besar dengan pemanasan segitiga setelah pemanasan nomor 2 didinginkan.

4. Deformasi pada bracket

- Panasi bracket yang melekok, di sisinya permukaan las dengan pemanasan garis.
- Panasi dengan api dan pukul dengan palu untuk meluruskan bracket yang melengkung tersebut.

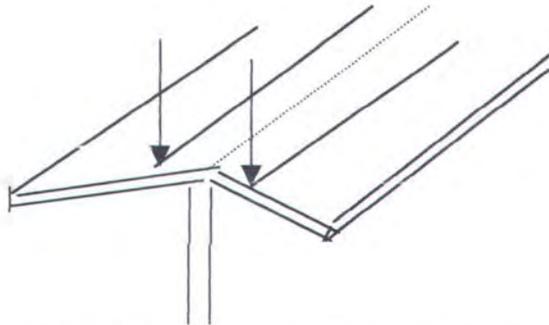


*Gambar 3.25 Pemanasan pada bracket*

5. Deformasi pada *face plate*

□ Face plate naik

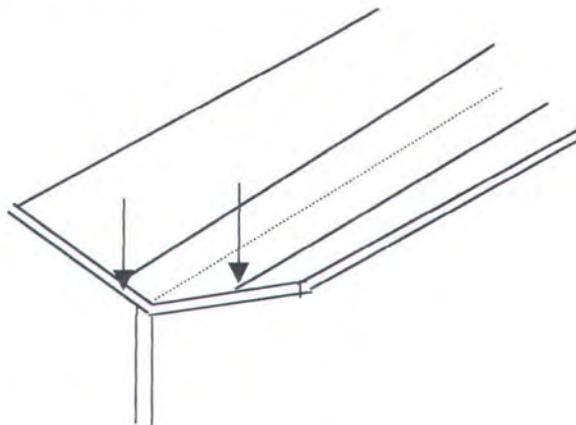
- Panasi sisi belakang dari *center* untuk pelat *web* dengan satu line pemanasan.
- Panasi sisi belakang permukaan las dua lines pemanasan.



Gambar 3.26 Pemanasan deformasi pada *face plate* yang naik

□ *Face plate* turun

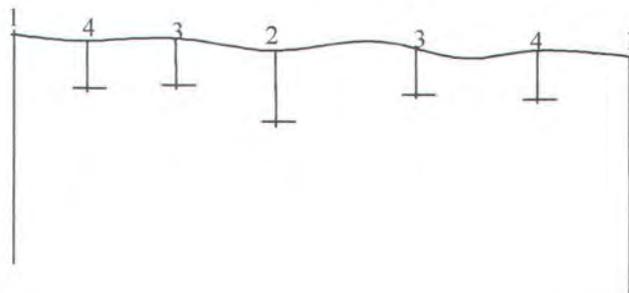
- Panasi sisi-sisi permukaan las *fillet* dengan pemanasan garis
- Panasi sisi belakang permukaan *fillet* dengan pemansan garis dan dorong *face plate* dari sisinya.



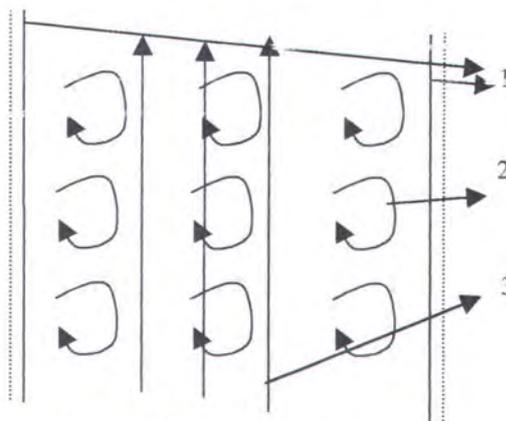
Gambar 3.27 Pemanasan deformasi pada *face plate* yang turun

6. Deformasi pada *deck* dan *wall* kedap

- Panasi dari pelat yang kuat ke pelat yang agak lemah.
- Panasi sisi penguat kemudian dinginkan dengan air secepat waktu memanasi, dari belakang daerah yang dipanasi.
- Panasi pelat dengan pemanasan cincin, kemudian dinginkan dengan air secepatnya seperti pemanasan pada pelat.
- Kemudian lakukan pemanasan garis lagi.
- Urut-urutan pemanasan harus sesuai dengan Gambar 3.28 dan Gambar 3.29 dibawah ini:



Gambar 3.28 Urut-urutan pemanasan pada *deck* dan *wall* kedap

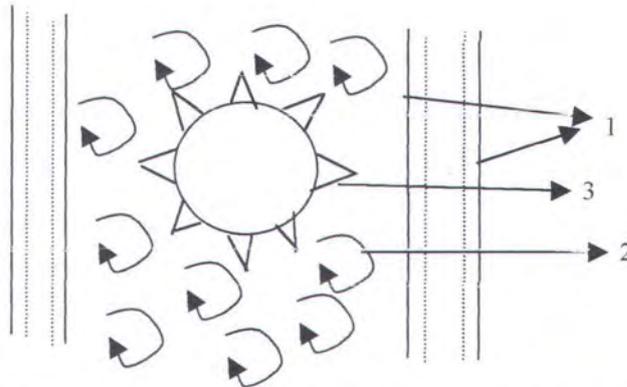


Gambar 3.29 Urut-urutan pemanasan

7. Deformasi pelat di daerah sekitar lubang.

- Panasi sisi belakang penguat dengan pemanasan garis.

- Panasi pelat antara penguat dan penguat dengan pemanasan cincin,
- Panasi pelat sekitar lubang dengan pemanasan sigitiga.
- Urut-urutan pemanasan harus sesuai dengan gambar 3.30 di bawah ini :



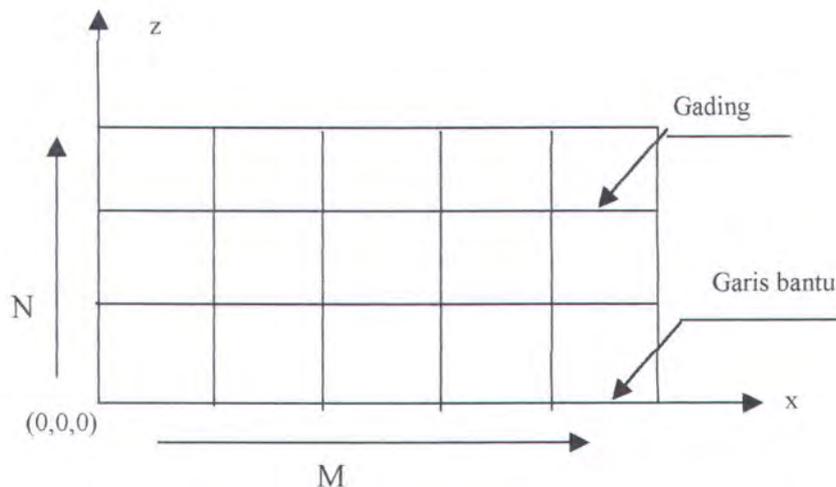
Gambar 3.30 Urut-urutan pemanasan pada deformasi pelat yang berlubang

## BAB IV

### PENYAJIAN KOMPUTER GRAFIS KOMPONEN KAPAL

#### IV.1. PENYAJIAN GRAFIS 3-D KOMPONEN KAPAL

Penyajian gambar komponen kapal menggunakan metode 3-D *polygon mesh* atau lebih dikenal dengan sebutan 3-D *mesh*. Suatu *polygon mesh* dari suatu matrik  $M \times N$  dimana  $M$  dan  $N$  berfungsi untuk menentukan posisi kolom dan baris yang membentuk *vertex*. *Polygon mesh* dapat berbentuk terbuka dan tertutup. Suatu *mesh* dikatakan terbuka jika garis-garis kolom atau barisnya tidak saling berhubungan, sebaliknya untuk *polygon mesh* dikatakan tertutup jika garis-garis kolom atau barisnya saling berhubungan. Aturan *mesh*  $M \times N$  ini yang digunakan untuk urutan memasukkan data.



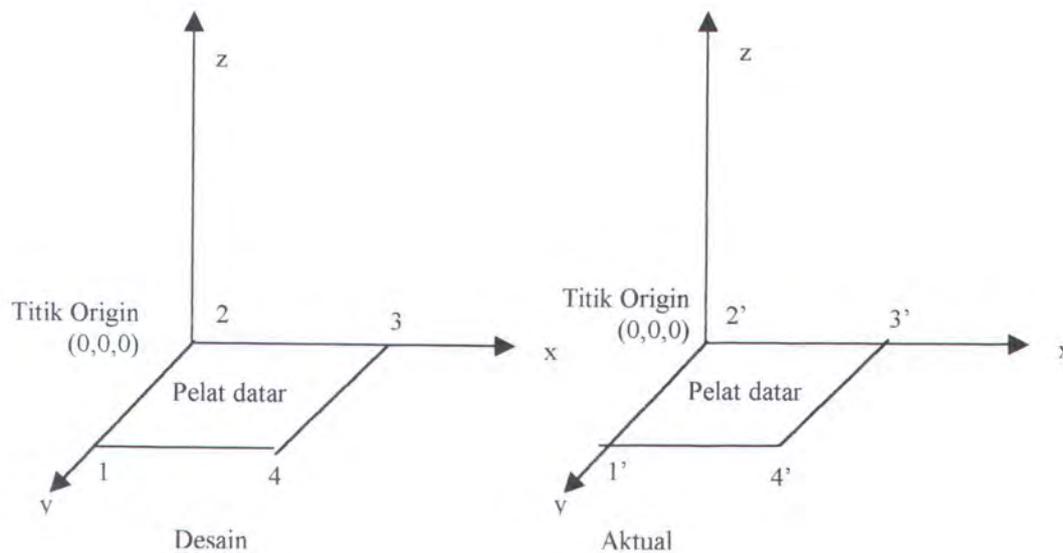
Gambar 4.1 Arah M dan N pada pelat kulit

Arah M dan N tergantung pada posisi komponen pada lambung kapal dan titik letak origin (0,0,0), titik pertama M x N terletak pada titik origin komponen. Untuk komponen memanjang kapal arah M dan titik origin komponen (0,0,0) ke arah memanjang kapal mengikuti bentuk komponen sesuai sumbu positifnya. Sedangkan arah N dari titik origin dan komponen mengikuti bentuk komponen (melintang ataupun vertikal) sesuai dengan sumbu positif. Untuk komponen melintang kapal (misalnya sekat atau gading) arah M dari titik origin komponen ke arah melintang kapal mengikuti bentuk komponen yang sesuai dengan sumbu positifnya. Sedangkan arah N dari titik origin komponen dan mengikuti bentuk komponen (biasanya vertikal) sesuai sumbu positif. Atau lebih mudahnya bahwa arah M sesuai dengan arah horisontal pelat dan arah N sesuai dengan arah vertikal pelat atau arah tegak lurus M dan M x N dari titik origin komponen ke arah sumbu positifnya masing-masing seperti Gambar 4.1 untuk komponen pelat memanjang kapal (pelat lambung).

#### **IV.2. SISTEM KOORDINAT PENYAJIAN GRAFIS 3-D**

Penggambaran komponen-komponen 3-D kapal, data input yang digunakan berupa koordinat tiga dimensi yang disebut *vertex* dimana bila titik-titik ini digambarkan dapat membentuk suatu grafis komponen 3-D kapal.

Ketentuan sistem koordinat yang digunakan dalam penggambaran komponen kapal menggunakan sistem koordinat lokal yang dapat diilustrasikan seperti gambar dibawah ini :



Gambar 4.2 Koordinat Cartesian dalam penggambaran bentuk komponen

- Titik yang diasumsikan sebagai titik origin sumbu koordinat 3-D antara desain dan data aktual harus mempunyai letak yang sama, seperti Gambar 4.2 Koordinat salah satu ujung pelat adalah  $(0,0,0)$ .
- Titik origin untuk komponen dengan letak melintang kapal pada salah satu ujung pelat sesuai dengan bentuk masing-masing komponen.
- Arah sumbu x untuk desain dan aktual sesuai dengan arah memanjang kapal.
- Arah sumbu y untuk desain dan aktual sesuai dengan arah melintang kapal.
- Arah sumbu z untuk desain dan aktual sesuai dengan arah vertikal kapal.
- Posisi komponen saat pengukuran dan penggambaran sesuai dengan posisi aslinya pada konstruksi kapal.

Untuk simulasi mengambil beberapa *member/part* yang diambil dari komponen kapal OHBC 42500 DWT yang telah diselesaikan oleh PT PAL INDONESIA. Dalam pembuatan simulasi ini, mengambil empat bagian yang

dapat mewakili bentuk-bentuk komponen secara umum. Bentuk komponen pada tahap assembly meliputi :

- *Flat unit*
- *Curve Unit*
- *Orthogonal Unit*
- *Non Orthogonal Unit*

Komponen kapal yang digunakan simulasi untuk standarisasi *fairing work* adalah komponen *Orthogonal* dan *Non Orthogonal unit*.

#### □ **Data Desain**

Data desain berisikan titik-titik koordinat 3-D yang diambil dari gambar desain pembuatan suatu komponen kapal. Pelaksanaan pembacaan data desain komponen kapal membutuhkan beberapa gambar yaitu *working drawing* (gambar kerja) dan *body plan*.

#### □ **Data Aktual**

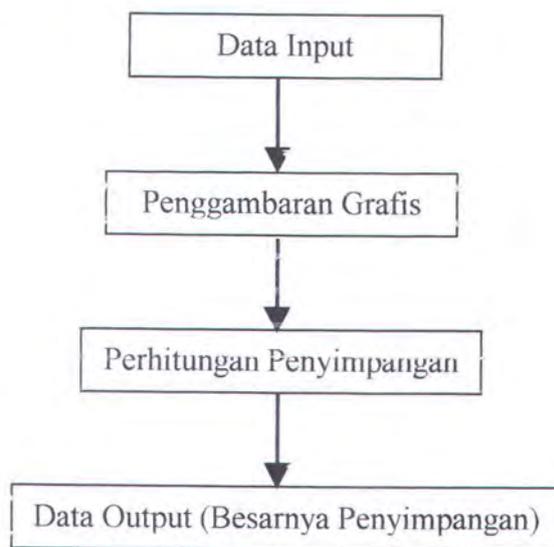
Data aktual diperoleh dengan cara simulasi, yaitu dengan menyimpangkan data desain. Penyimpangan yang disimulasikan adalah penyimpangan deformasi ke arah bidang pelat atau material. Titik-titik koordinat baik desain maupun aktual mempunyai posisi yang sama dengan sistem satuan dimensi yang sama pula.

### **IV.3. PENCARIAN BESARNYA PENYIMPANGAN BENTUK**

Penyimpangan bentuk yang terjadi dapat diketahui dengan cara membandingkan antara data desain dengan data aktual. Untuk mengetahui besarnya penyimpangan dilakukan dengan cara *me-matching*-kan data aktual

dengan data desain. Apabila data desain dan aktual diimpitkan, maka akan terdapat beberapa bagian komponen yang tidak berimpit menjadi satu. Bagian yang tidak berimpit inilah yang dihitung selisihnya, sehingga akan diperoleh data penyimpangan bentuk.

Langkah pencarian besarnya penyimpangan tersebut dapat digambarkan seperti diagram berikut :



Gambar 4.3 Langkah pencarian besar penyimpangan

Keterangan Gambar 4.3, langkah-langkah pencarian besarnya penyimpangan :

□ **Data Input**

Input ini digunakan untuk proses pemasukan dan pembacaan data yang dimasukkan oleh *user*. Disusun menggunakan bahasa pemrograman Delphi 3. Data input yang digunakan berupa koordinat tiga dimensi yang disebut *vertex* tiga dimensi dimana bila titik-titik ini digambarkan akan membentuk suatu gambar grafis komponen 3-D kapal. Oleh karena itu yang perlu diperhatikan adalah cara pemasukan data yang berhubungan dengan metode penggambaran 3-D *mesh*.

Langkah-langkah untuk penggambaran 3-D *mesh* pertama-tama memasukkan jumlah baris (M) dan jumlah kolom (N) komponen. Sebelumnya komponen dibagi dengan sejumlah garis bantu M dan N yang mempunyai jumlah titik  $M \times N$  (vertex). Kemudian langkah kedua memasukkan titik-titik *vertex* tersebut sesuai dengan urutan yang diberikan oleh program input. Data input yang diperlukan untuk pelaksanaan pencarian besarnya penyimpangan terdiri dari data desain dan aktual.

#### □ **Penggambaran Grafis**

Penggambaran komponen kapal dalam mencari besarnya penyimpangan menggunakan fasilitas grafis yang ada pada Delphi 3. Ketebalan komponen 3-D yang akan digambar diabaikan, sehingga bentuk gambar berupa permukaan (surface) sesuai titik-titik koordinat 3-D input. Gambar permukaan (surface) pada media tampilan yang dihasilkan harus dapat mewakili bentuk benda sebenarnya. Sesuai dengan input berupa  $M \times N$  titik, gambar yang dihasilkan yaitu berupa *mesh* atau jaring-jaring *polygon*.

#### □ **Perhitungan Penyimpangan**

Perhitungan penyimpangan menggunakan bahasa pemrograman Delphi 3. Tiap titik koordinat 3-D dihitung menggunakan ketentuan bahwa besar selisih antara koordinat desain dengan aktual dianggap tidak menyimpang jika masih di dalam batasan toleransi. Begitu pula sebaliknya jika selisih antara koordinat desain dengan aktual diluar batasan toleransi maka dianggap menyimpang. Langkah-langkah perhitungan penyimpangan adalah sebagai berikut :

- Mencari selisih awal ( $\Delta x'$ ,  $\Delta y'$ ,  $\Delta z'$ ), titik-titik koordinat 3-D aktual sebesar ( $x'$ ,  $y'$ ,  $z'$ ) dikurangi dengan titik-titik koordinat desain sebesar ( $x$ ,  $y$ ,  $z$ ).
- Mencari titik dengan selisih koordinat 3-D paling kecil, nilai masing-masing  $\Delta x'$ ,  $\Delta y'$  dan  $\Delta z'$ .
- Menghitung besar penyimpangan pada semua titik yang menyimpang ( $\Delta x''$ ,  $\Delta y''$ ,  $\Delta z''$ ) ; tiap titik koordinat 3-D ( $\Delta x'$ ,  $\Delta y'$ ,  $\Delta z'$ ), dikurangi dengan ( $\Delta x'$ ,  $\Delta y'$ ,  $\Delta z'$ ) terkecil.

#### □ Output

Setelah kedua data input baik desain dan aktual digambarkan maka akan diketahui bentuk dan besar penyimpangan. Gambar komponen yang menyimpang tersebut akan di-*matching*-kan ke *database* penyimpangan, untuk dilakukan standarisasi perbaikan.

## BAB V

# PENERAPAN DATABASE MANAGEMENT SYSTEM DALAM PENGOLAHAN DATA

### V.1 KONSEP DATABASE MANAGEMENT SYSTEM (DBMS)

*Database Management System* (DBMS) adalah sekumpulan data yang saling berhubungan dan set program untuk mengakses data tersebut. *Database Management System* (DBMS) terdiri dari :

#### □ DATABASE

*Database* adalah sekumpulan file-file yang saling berkaitan antara file yang satu dengan file yang lain sehingga membentuk bangunan data untuk menginformasikan sebuah perusahaan atau instansi dalam batasan tertentu. Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk dan merupakan kumpulan *entity* yang seragam. Setiap satu *record* terdiri *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut satu pengertian yang lengkap dan direkam dalam satu *record*. Apabila kita ingin menyebut isi dari *field* maka digunakan atribut atau judul dari kelompok *entity* tertentu.

#### □ PROGRAM PENGELOLA

Program pengelola merupakan satu paket program yang dibuat agar memudahkan dan mengefisiensikan pengorganisasian data ke dalam *database*.

Pengelolaan data meliputi :

- Menambah data

- Menghapus data
- Mengambil data
- Membaca data
- Membuat *query*
- Mengurutkan data
- Mencetak data

Pengelolaan data tersebut di atas harus diorganisasikan dengan sistematis agar dapat bekerja dengan baik.

## V.2. PERANAN PROGRAM DATABASE

Di dalam perusahaan sering muncul permasalahan yang diakibatkan oleh rumitnya mengatur dan mengolah data. Pengolahan data yang ada di perusahaan sering menjadikan masalah tersendiri. Hal ini diakibatkan begitu kompleksnya data yang ada, sehingga diperlukan suatu metode yang tepat agar data-data tersebut dapat diatur sedemikian rupa. Masalah-masalah yang timbul pada proses pengolahan data tersebut dapat dikelompokkan menjadi :

- *Redundansi dan Inkonsistensi Data*
- Kesulitan Pengaksesan Data
- Isolasi Data untuk Standarisasi
- *Multiple User*
- *Data Security*
- *Integrasi Data*
- *Independence Data*

### **Redundansi dan Inkonsistensi Data**

*Redundansi* data adalah penyimpanan data di beberapa tempat yang sama. *Redundansi* terjadi apabila file-file *database* dan program pengelola dibuat pada selang waktu yang lama oleh programmer yang berbeda. Sehingga ada beberapa bagaian data yang sama pada file-file yang berbeda dalam satu *database*. Adanya redundansi akan menyebabkan pemborosan pada ruang penyimpanan data dan mengakibatkan terjadinya *inkonsistensi* data.

### **Kesulitan Pengaksesan Data**

Perhitungan data-data di suatu perusahaan yang menggunakan spesifik khusus akan menyulitkan dalam percetakannya. Hal ini disebabkan kriteria yang diminta tidak terdapat dalam program yang dibuat untuk mencetak data tersebut. Untuk mengatasi hal tersebut adalah diaplikasikannya *database management system* (DBMS) yang mampu mengambil data secara langsung dengan bahasa yang familiar dan mudah digunakan yaitu melalui penyediaan menu-menu program pengelolanya.

### **Isolasi Data untuk Standarisasi**

Data yang tersebar dalam beberapa file dalam bentuk format yang tidak sama akan menyulitkan dalam menulis program pengelolanya untuk dapat mengambil dan menyimpan data. Untuk mengatasi hal tersebut, data dalam suatu *database* harus dibuat dalam satu format yaitu *format text database* sehingga akan memudahkan dalam pembuatan program pengelolanya.

### Multiple User

Alasan mengapa *database* dibangun, karena data-data dapat digunakan oleh orang banyak dalam waktu yang sam atau atau berbeda. Pengaksesan data dilakukan oleh program yang sama tetapi berbeda orang dan waktu. Oleh karena itu sistem *database* mengijinkan banyak pemakai untuk meng-*update* data secara bersama-sama atau bergantian.

### Data Security

Tidak semua pemakai sistem *database* diperbolehkan untuk mengakses semua data dan melakukan pengorganisasian data yang telah diaksesnya. Data Security ini dapat diatur lewat program yang dibuat programmer atau fasilitas keamanan dari *operting system*-nya, misalnya; *Novel Netware* untuk *Local Area Network* dan *Direct Acces* untuk *Personal Computer*.

### Integrasi Data

*Database* berisi file-file yang saling berkaitan antara file yang satu dengan file yang lain. Permasalahan yang timbul adalah bagaimana mengaitkan antara file yang satu dengan file yang lain. Mengatasi permasalahan tersebut adalah bagaimana menentukan *field* kunci yang mampu mengaitkan satu file ke file lain ke dalam *database*.

### Independence Data

Apabila suatu program *database* mengalami perubahan struktur pada *file*-nya maka program tersebut harus dirubah maka dapat dikatakan bahwa pengelola *database* tersebut tidak bebas terhadap *database* yang ada. Lain halnya jika pengelolaan *database* menggunakan pengolahan *database management*

*system* (DBMS), apapun perubahan yang terjadi pada struktur file tidak akan merubah program tersebut. Hal ini berarti perintah-perintah dalam paket DBMS bebas terhadap *database* yang dikelolanya. Apapun perubahan dalam *database*, semua perintah akan mengalami kesetabilan tanpa perlu diubah-ubah. Namun demikian harus dipikirkan bagaimana jika ada syarat-syarat terhadap *database* yang ada.

### V.3. DATA ABSTRACTION (ABSTASKSI DATA)

Database Management System mempunyai kegunaan agar *user* mampu menyusun suatu pandangan abstraksi dari data. Bayangan mengenai data akan terbawa ke suatu pemikiran bagaimana suatu data masuk ke *database*, bagaimana penyimpanannya ke dalam *disk* dan di sektor manakah disimpan. Secara keseluruhan harus mampu dibayangkan bagaimana data diabstraksikan atau digambarkan menyerupai kondisi yang dihadapi pemakai sehari-sehari. Adapun sistem yang sesungguhnya dalam *database management system* adalah bagaimana cara untuk menyembunyikan kerumitan, yang kemudian diungkapkan ke dalam bahasa dan gambar yang mudah dimengerti oleh setiap orang.

Jadi pada dasarnya abstraksi data adalah suatu kemampuan data untuk membentuk suatu pandangan tertentu pada *user*. *User* dikelompokkan menjadi tiga tingkatan abstraksi saat memandang suatu *database* yaitu :

- Level fisik
- Level konseptual
- Level pandangan pemakai

Masing-masing level mempunyai tingkat abstraksi yang berbeda.

### **Level Fisik**

Level fisik adalah level abstraksi yang paling rendah. Hal ini menggambarkan bagaimana data disimpan dalam kondisi sebenarnya, seperti :

- Bagaimana proses data diubah dalam bentuk bahasa mesin.
- Bagaimana proses penambahan sector bila data telah ditambah dan disimpan dalam media penyimpanan.
- Bagaimana proses pembagian sector untuk data yang tidak sama. Level ini merupakan level paling kompleks dan digunakan oleh seorang system analyst.

### **Level Konseptual**

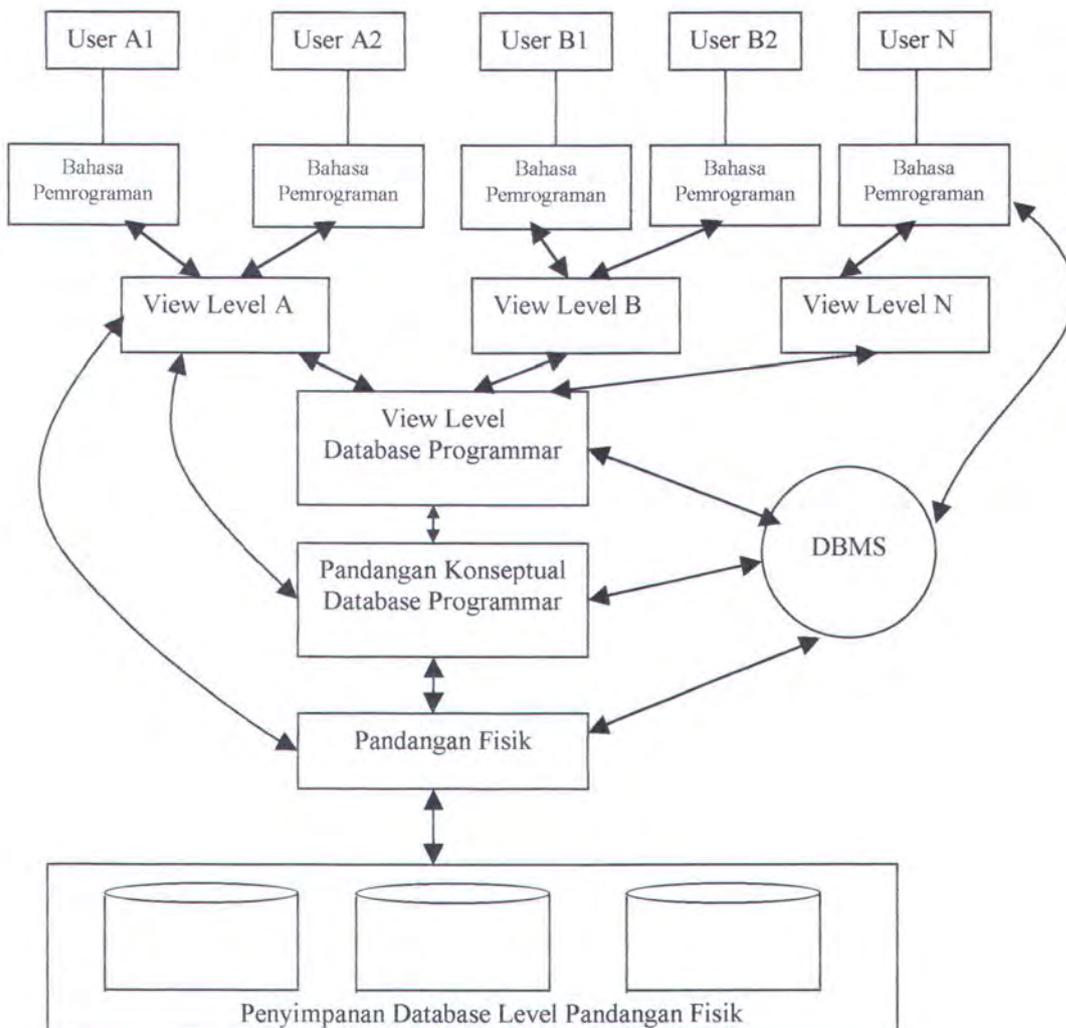
Level konseptual merupakan level abstraksi data yang lebih tinggi daripada level fisik. Pada level ini menggambarkan data apa yang disimpan dalam database, dan hubungan yang terjadi antar data. Pemakai tidak lagi menemukan kerumitan seperti pada level fisik. Penggambaran cukup memakai kotak, garis keterangan secukupnya. Level ini digunakan oleh *database administrator* (programmer) yang memutuskan informasi apa saja yang akan dipakai dan dipelihara oleh *user* dalam *suatu database*.

### **Level Pandangan Pemakai (view level)**

Level pandangan pemakai merupakan level abstraksi tertinggi yang menggambarkan hanya satu bagian dari keseluruhan data. Bila pada level konseptual data merupakan suatu kumpulan informasi yang besar dan kompleks, maka pada level ini data hanya sebagian saja yang dilihat dan dipakai. Orang

yang bekerja pada level ini sering *user* program. Tingkatan abstraksi data di atas didefinisikan untuk memudahkan hubungan antara user pada level pandangan pemakai dengan program pengelola *database* yang dibuat pada level konseptual.

Dengan menggunakan level abstraksi data di atas maka kita dapat lebih mudah memahami arsitektur *database management system* seperti pada gambar dibawah ini :



Gambar 5.1 Architecture DBMS

#### V.4. PAKET BAHASA YANG DIGUNAKAN DALAM DATABASE MANGEMENT SYSTEM

Dalam mengolah pengorganisasian *database*, menggunakan paket-paket bahasa yang dapat dijadikan sebagai alat komunikasi. Di dalam *database management system* mengenal tiga paket bahasa yaitu :

- Data Definition Language
- Data Manipulation Language
- Query Language

Ketiga paket di atas mempunyai kegunaan sendiri-sendiri di dalam pengorganisasian, tetapi ketiga paket bahasa tersebut akan bekerja bersama-sama dalam proses pengorganisaian.

##### **Data Definition Language**

*Data definition language* merupakan satu paket bahasa yang berisi satu set definisi perintah khusus untuk menspesifikasikan pola *database* yang diorganisasikan. Hasil spesifikasi dari perintah-perintah khusus tersebut adalah satu set tabel yang disimpan dalam file khusus yang disebut data *dictionary* atau *directory*. Setiap *directory* adalah satu file yang berisi metadata, yaitu data mengenai data. File ini dikonsultasikan dahulu sebelum data sebenarnya dibaca atau dimodifikasi dalam satu sistem *database*.

Struktur penyimpanan dan metode akses yang digunakan sistem *database* dispesifikasikan dengan satu set definisi dalam satu tipe *database definition language* (DDL) yang disebut *data storage* dan *definition language*. Hasil

kompilasi dari definisi ini adalah satu set perintah yang menspesifikasikan suatu terapan yang rinci dari pola *database* yang biasanya tersembunyi dari *user*.

### **Data Manipulation Language**

*Data manipulatoin language* merupakan paket bahasa yang mengizinkan pemakai untuk mengakses dan memanipulasi data sebagai yang telah diorganisasikan sebelumnya dalam model data yang tepat. *Database management system user* untuk :

- Mengambil informasi baru ke dalam *database*.
- Menyisipkan informasi baru ke dalam *database*.
- Menghapus informasi dari *database*.

Di dalam *database management* ada dua tipe, yaitu:

#### □ Procedural Data Manipulation Language

*Data manipulation language* yang membutuhkan masukan dari pemakai untuk menspesifikasikan data apa saja yang dibutuhkan dan bagaimana cara untuk mendapatkannya.

#### □ Non Procedural Data Manipulation Language

*Data manipulation language* yang membutuhkan masukan dari user untuk menspesifikasikan data apa saja yang dibutuhkan tanpa menspesifikasikan bagaimana cara untuk mendapatkannya. Tipe Non Procedural Data Manipulation Language sangat mudah dipelajari oleh user dibanding dengan Procedural Data Manipulation Language. Keduanya akan dipakai dalam pembuatan program *database* standarisasi perbaikan penyimpangan bentuk komponen kapal.

### Query Language

*Query language* adalah pernyataan yang berisi kriteria tertentu untuk diajukan dalam mengambil suatu informasi menurut kriteria tersebut. Berdasarkan cara kerjanya *query* ini termasuk *data manipulation language*.

## V.5. PERANCANGAN DATABASE

Untuk merancang suatu *database* diperlukan suatu keahlian dan pengetahuan tentang *database* itu sendiri. Hal ini karena merancang *database* merupakan pekerjaan yang tidak mudah. Kesulitan yang dihadapi dalam merancang *database* adalah bagaimana merancang pola *database*-nya sehingga dapat memuaskan *user* sesuai dengan yang diinginkan. Untuk itu perancangan model konseptual akan menunjukkan *entity* dan relasinya berdasarkan proses yang diinginkan oleh organisasi perusahaan. Ketika menentukan *entity* dan relasinya dibutuhkan analisis data tentang informasi yang ada dalam spesifikasi di masa datang. Semua ini adalah tugas *database administrator*. Pada perancangan model konseptual, penekanan dilakukan pada struktur data dan struktur program.

Pendekatan yang dilakukan pada model konseptual ada dua cara yaitu :

- Teknik Normalisasi

Teknik normalisasi ini dipakai dalam pembuatan program untuk mengelola satu file *database* tanpa ada hubungan dengan file yang lain dalam satu *database*. Jadi teknik ini hanya dipakai untuk satu file *database* saja. Dalam pelaksanaannya teknik normalisasi ini dikenal dengan dua jenis kunci yaitu :

- Field Key (Field kunci)

*Field* kunci adalah suatu *field* dalam file yang dapat mewakili suatu *record* tertentu. Setiap pencarian *record* cukup menyebut salah satu dari *field* di atas maka akan didapatkan data lengkap mengenai *record* yang diinginkan. Namun ada kemungkinan bahwa beberapa *record* yang diwakili oleh salah satu kunci *field*. Oleh karena itu kunci *field* kurang efektif bila dipakai untuk mengidentifikasi suatu kejadian spesifik dari *entity* dalam mencari suatu *record*.

- Field Candidate Key (Field kunci kandidat)

*Field* kunci kandidat adalah suatu kunci *field* yang dapat mengidentifikasi secara unik suatu kejadian spesifik dari suatu *entity*. Karena kekhususannya maka *field* kunci kandidat dapat digunakan untuk mencari *record* tertentu yang mempunyai ciri-ciri berbeda dengan *record* yang lain, baik yang melalui perintah *search locate* dan *find* teknik normalisasi ini hanya digunakan untuk mengorganisasi file saja, sehingga teknik ini belum merupakan teknik pemrograman *database* yang dapat memberikan hasil yang optimal bagi *user*.

- Teknik Entity Relationship

Teknik *entity relationship* digunakan untuk membuat program pengelola file-file yang saling berkaitan. Teknik ini memerlukan suatu *field* kunci relasi (Relation Field Key), yaitu *field* kunci yang terdapat dalam file-file suatu *database*. *Field* ini menjadi *field* penghubung antara *field* yang satu ke yang lain, sehingga bila programmer menghapus *field* kunci ini maka *field* ini akan hilang dari

file-file yang memuatnya. Ada beberapa jenis hubungan antar *field* dalam satu *database* yang ditentukan oleh jenis *field* kunci relasinya, yaitu :

- One to one Relationship

*One to one relationship* adalah hubungan antara dua file yang berbeda dalam satu atau lebih *database*. Hubungan ini diwujudkan dengan penentuan satu atau lebih *field* kunci relasi sehingga pengontrolan *record* pada dua file tersebut dapat dilakukan melalui *field* kunci relasi tersebut. Misalnya programmer ingin membuat *query* pada dua file terhubung pada *field* kunci relasi tertentu, maka ia akan mendapatkan hasil proses *query* yang sesuai kriteria dari ke dua file yang bersangkutan.

- One to Many Relationship

*One to many relationship* adalah hubungan antara dua file atau lebih yang berbeda dalam satu atau lebih *database*. Hubungan ini diwujudkan dengan penentuan satu atau lebih *field* kunci relasi. Bila dilakukan suatu proses akan memberikan hasil yang berhubungan dengan file-file yang terkait. Apabila diadakan perubahan *field* kunci relasi tersebut, maka data value dari *record* di dalamnya terdapat *field* tersebut akan berubah. Untuk itu penerapan teknik *entity relationship* ini diperlukan pengalaman dan harus hati-hati dalam memakainya, karena suatu kesalahan yang tak disengaja yang dilakukan pada *field* kunci relasi akan mengakibatkan perubahan data *record* file-file terkait. Teknik *entity* ini sangat berguna dan akan memberikan hasil yang optimal bila untuk merancang program *database* yang mempunyai.

## BAB VI

### PROTOTYPE PROGRAM STANDARISASI FAIRING WORK PENYIMPANGAN BENTUK KOMPONEN KAPAL

#### VI.1. PEMILIHAN BAHASA PEMROGRAMAN

Bahasa pemrograman merupakan software yang digunakan untuk pembuatan pembuatan program-program baik dalam bahasa tingkat tinggi maupun tingkat rendah. Bahasa tingkat tinggi mempunyai perintah mirip dengan bahasa manusia, mudah dimengerti namun kemampuannya terbatas. Sedangkan bahasa tingkat rendah mempunyai perintah yang berbeda dengan bahasa manusia, sulit dimengerti, namun kemampuannya tinggi. Bahasa tingkat tinggi tersebut antara meliputi :

- BASIC
- PASCAL
- COBOL
- Visual Basic
- Borland Delphi

Sedang bahasa tingkat rendah antara lain meliputi :

- Assembler
- C<sup>++</sup>

Bahasa pemrograman yang digunakan dalam tugas akhir ini menggunakan bahasa pemrograman tingkat tinggi *Borland Delphi for Windows Release 3.0*. Dasar

pertimbangan pemilihan bahasa pemrograman tingkat tinggi *Borland Delphi for Windows Release 3.0* dalam penyusunan tugas akhir ini adalah :

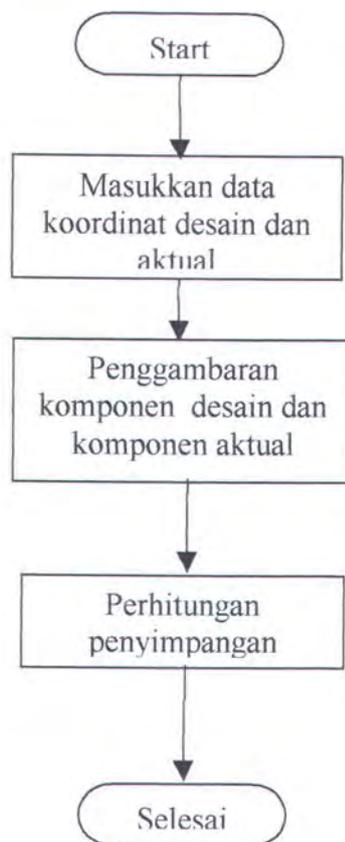
- Merupakan bahasa pemrograman yang mempunyai kemampuan pada pengolahan database, baik di dalam menata maupun mengelola informasi database yang masih tersimpan dalam bentuk daftar, map dan tabel. Penyimpanan data yang masih manual tersebut dapat diorganisasikan ke dalam database management system yang mampu dikerjakan oleh Borland Delphi, sehingga data tersebut dapat disederhanakan.
- Mempunyai kemampuan membuat program pengolah database melalui cara penulisan Syintak dalam jendela Command dan melalui Screen builder, Menu builder dan Report writer.
- Mempunyai kemampuan untuk membuat grafis tiga dimensi, sehingga dalam pembuatan prototipe program tugas akhir ini tidak perlu menggunakan bahasa pemrograman tingkat tinggi lainnya yang dapat memperlambat running program.
- Waktu pengaksesan program kurang lebih delapan kali lebih cepat
- Mempunyai fasilitas aplikasi visual yang menarik yang tercakup dalam Screen builder, Menu builder dan Report builder.
- Mempunyai fasilitas dasar dari sistem operasi microsoft windows yaitu DDE (Dynamic Data Exchange) yang menyebabkan Microsoft Borland Delphi for Windows ini dapat berkomunikasi dengan program aplikasi Windows yang lain sehingga dapat menukar data dengan spreads sheet, word processor, sistem e-mail dan OLE (Obyect Linking and Embedding) yang digunakan

untuk berhubungan dengan program aplikasi windows yang lain kemudian mengambil tampilan program aplikasi tersebut seperti tabel, bunyi, spread sheet, gambar dan lain-lain.

## VI.2. FLOWCHART PROGRAM

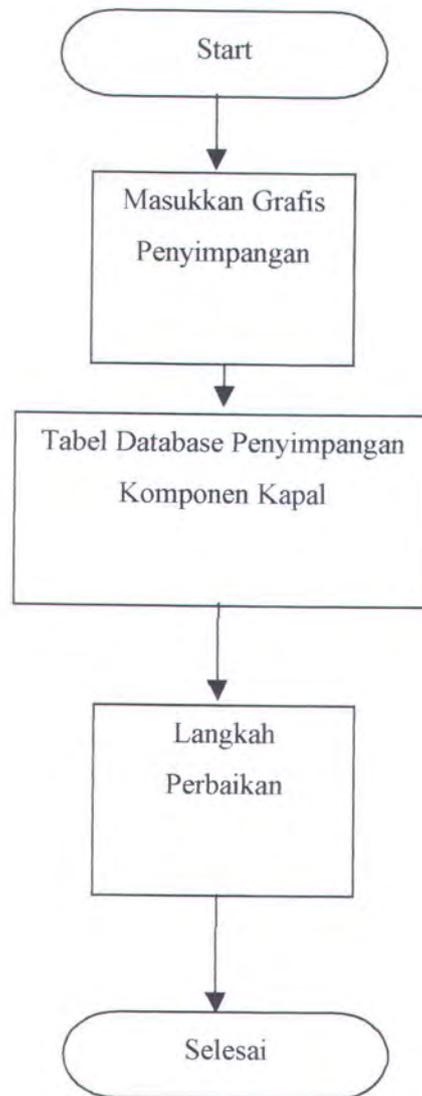
Di dalam program ini selain menggambarkan sebuah penyimpangan yang terjadi pada komponen kapal juga menentukan standarisasi perbaikannya dengan menggunakan program *database*.

- Flowchart Penggambaran Penyimpangan



Gambar 6.1 Flowchart penggambaran penyimpangan

- Flowchart identifikasi dan perbaikan penyimpangan



Gambar 6.2 Flowchart penggunaan program database

### VI.3. STRUKTUR PROGRAM

Dalam Borland Delphi 3.0 struktur program disusun dalam form yang membentuk suatu tampilan window (jendela). Satu form membentuk satu tampilan window. Dalam suatu form dapat disusun suatu listing program sesuai dengan kebutuhan yang diperlukan. Stuktur protitipe program ini disusun menjadi beberapa form yaitu :

□ **Form Screen Judul**

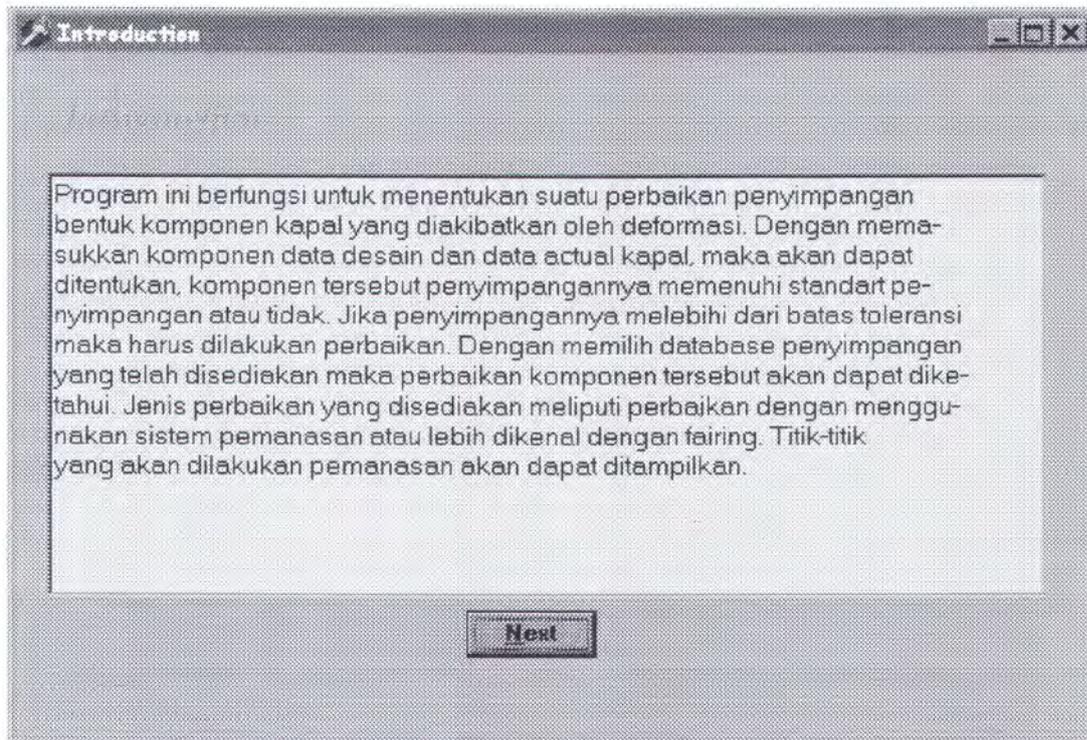
Merupakan tampilan judul dari program.



Gambar 6.3 Tampilan Judul

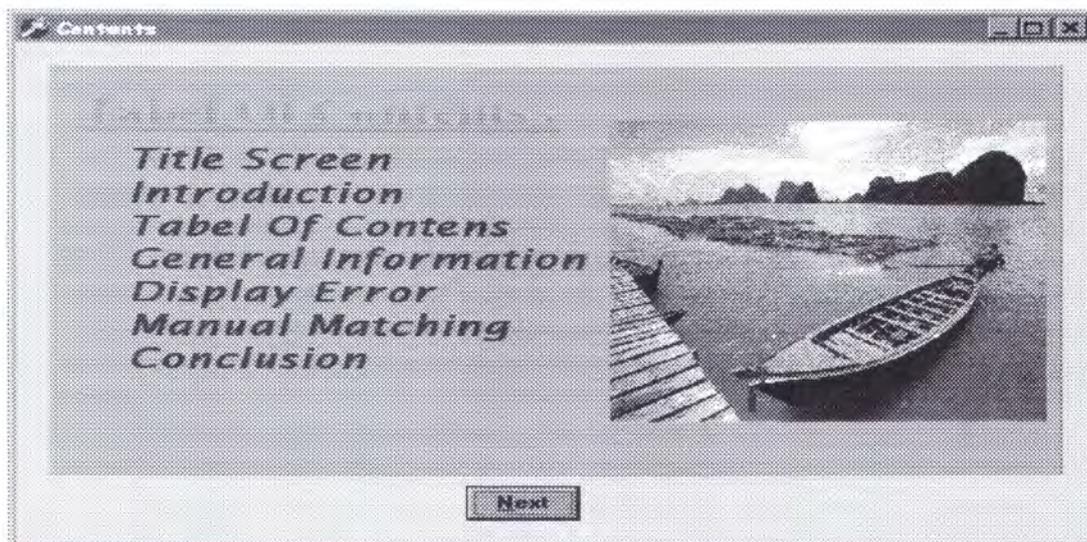
#### □ Form Introduction

Merupakan tampilan yang berisi tentang kegunaan program.



Gambar 6.4 Tampilan Introduction

#### □ Form Daftar Isi

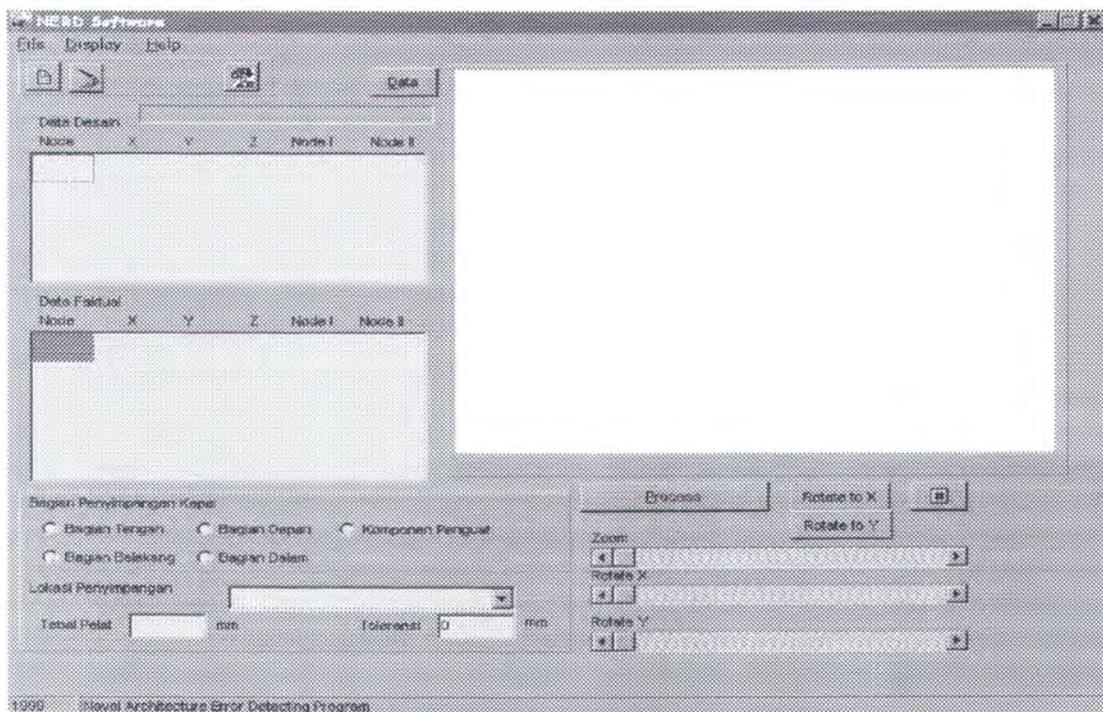


Gambar 6.5 Tampilan Daftar Isi Program

□ **Form Utama**

Di dalam Main. Frm berisikan program utama tentang pembacaan data input yang dibagi menjadi beberapa sub program, yaitu :

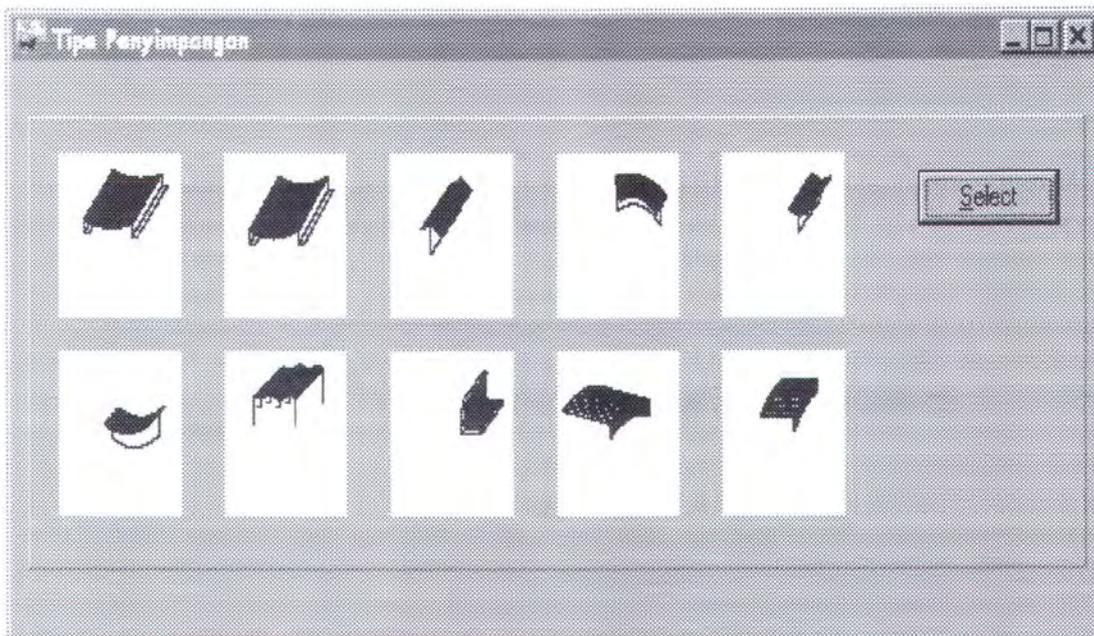
- New Project; berfungsi memulai file baru yang berisikan form tentang pengisian data utama kapal.
- Open Project, berfungsi membuka project yang berisi data desain dan data aktual.
- Open data desain; berfungsi membuka sebuah file data desain.
- Open data faktual; berfungsi membuka sebuah file data aktual.
- Save data desain; berfungsi berfungsi menyimpan data desain.
- Save data faktual; berfungsi berfungsi menyimpan data aktual.
- Save project; berfungsi berfungsi menyimpan semua project.
- Exit; berfungsi untuk keluar dari program.



Gambar 6.6 Tampilan Form Utama

□ **Form Display Tipe Penyimpangan**

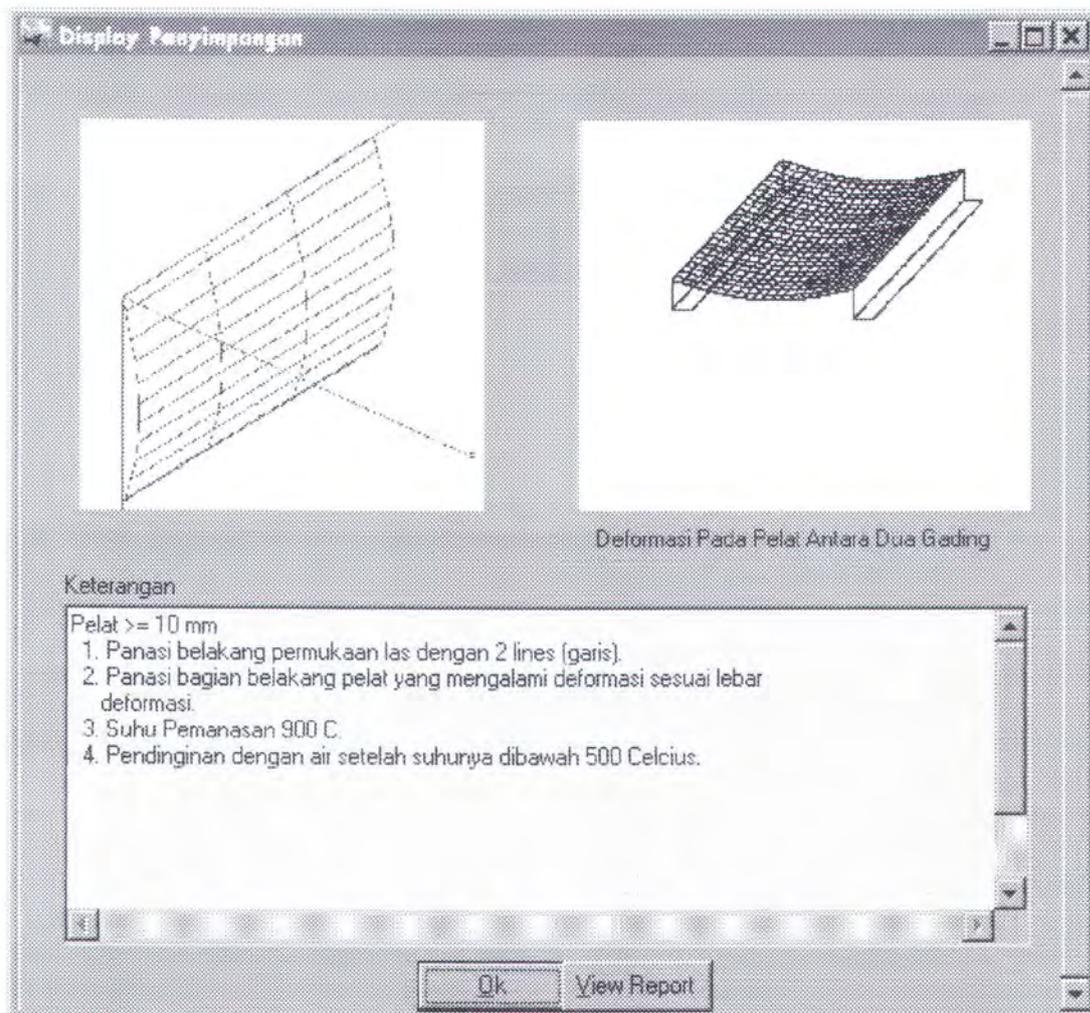
Di dalam display tipe penyimpangan berisi gambar-gambar penyimpangan pada komponen kapal akibat deformasi. Gambar-gambar tersebut merupakan gambar pilihan untuk menentukan jenis penyimpangan yang terjadi. Setelah memilih salah satu gambar yang ada dengan cara diklik maka program akan melakukan proses selanjutnya yaitu melihat perbaikan dari penyimpangan yang ada.



Gambar 6.7 Tampilan Tipe Penyimpangan Dalam Bentuk Grafis

□ **Form Display Penyimpangan**

Di dalam display penyimpangan berisi grafis komponen kapal yang mengalami penyimpangan dan grafis tipe penyimpangan serta instruksi perbaikan terhadap komponen yang mengalami penyimpangan tersebut.



Gambar 6.8 Tampilan Prnyimpangan dan Instruksi Perbaikan

#### VI.4. RUNNING PROGRAM

Langkah-langkah menjalankan Prototipe Program 'Penerapan Grafik Database Untuk Standarisasi Fairing Work Penyimpangan Bentuk Komponen Kapal' adalah sebagai berikut :

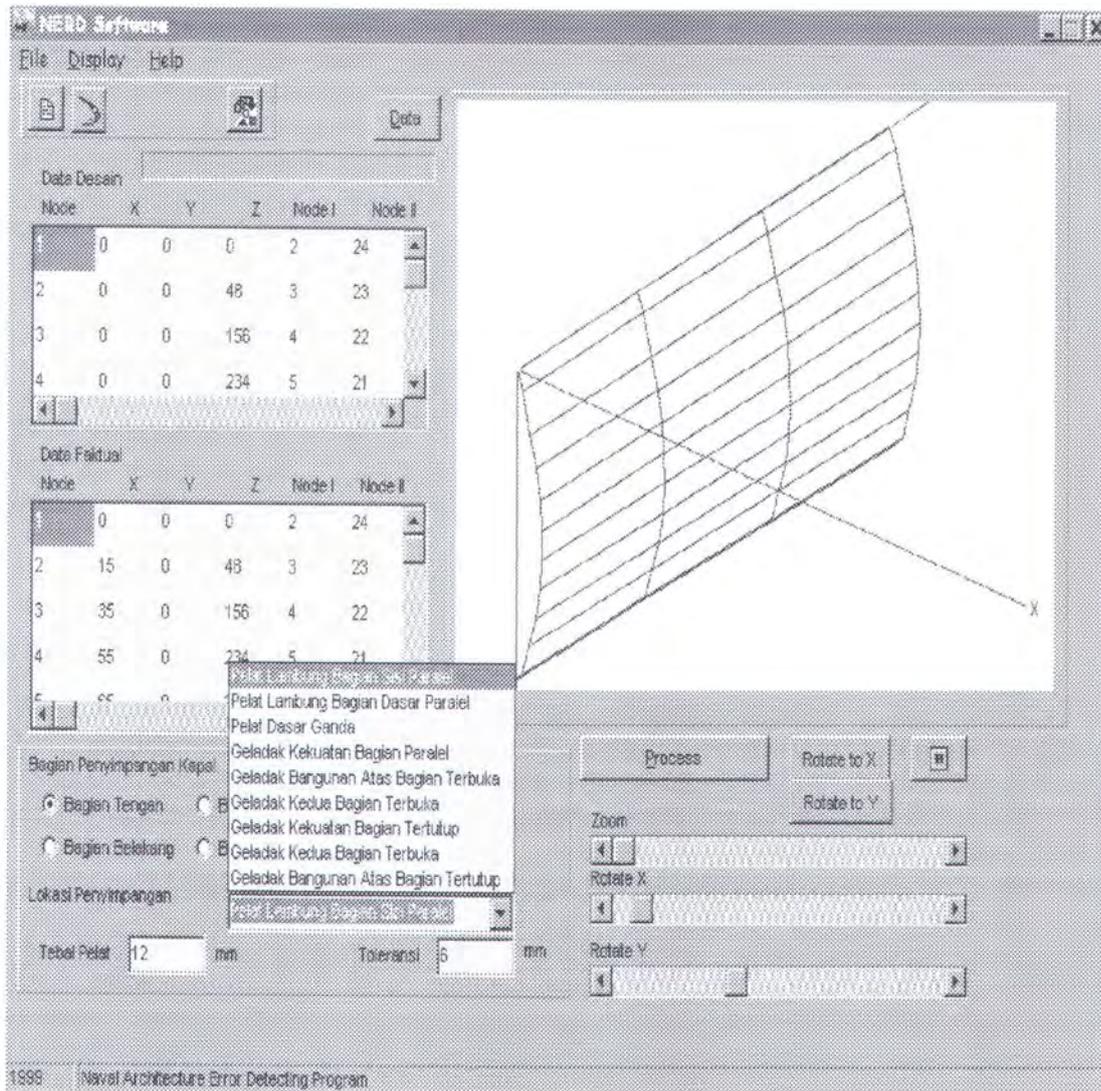
1. Pilih program *Naval Architect Error Detecting* pada Start Menu. Program secara berurutan akan menampilkan Form Screen Judul, Form Introduction, Form Daftar Isi dan Form Utama.

Form Utama terdiri dari :

- Menu bar :  
File, display, help
- Grid tabel 'Data desain' dan grid tabel 'Data aktual'
- Text 'Bagian tengah', 'Bagian depan', 'Bagian belakang', 'Bagian dalam' dan 'Komponen penguat'.
- Command botton 'Data', 'Process', 'Rotate to x', 'Rotate to y', 'Pause'.
- Scrollbar 'Zoom', 'Rotate x', 'Rotate Y'
- Image Gambar komponen 3-D

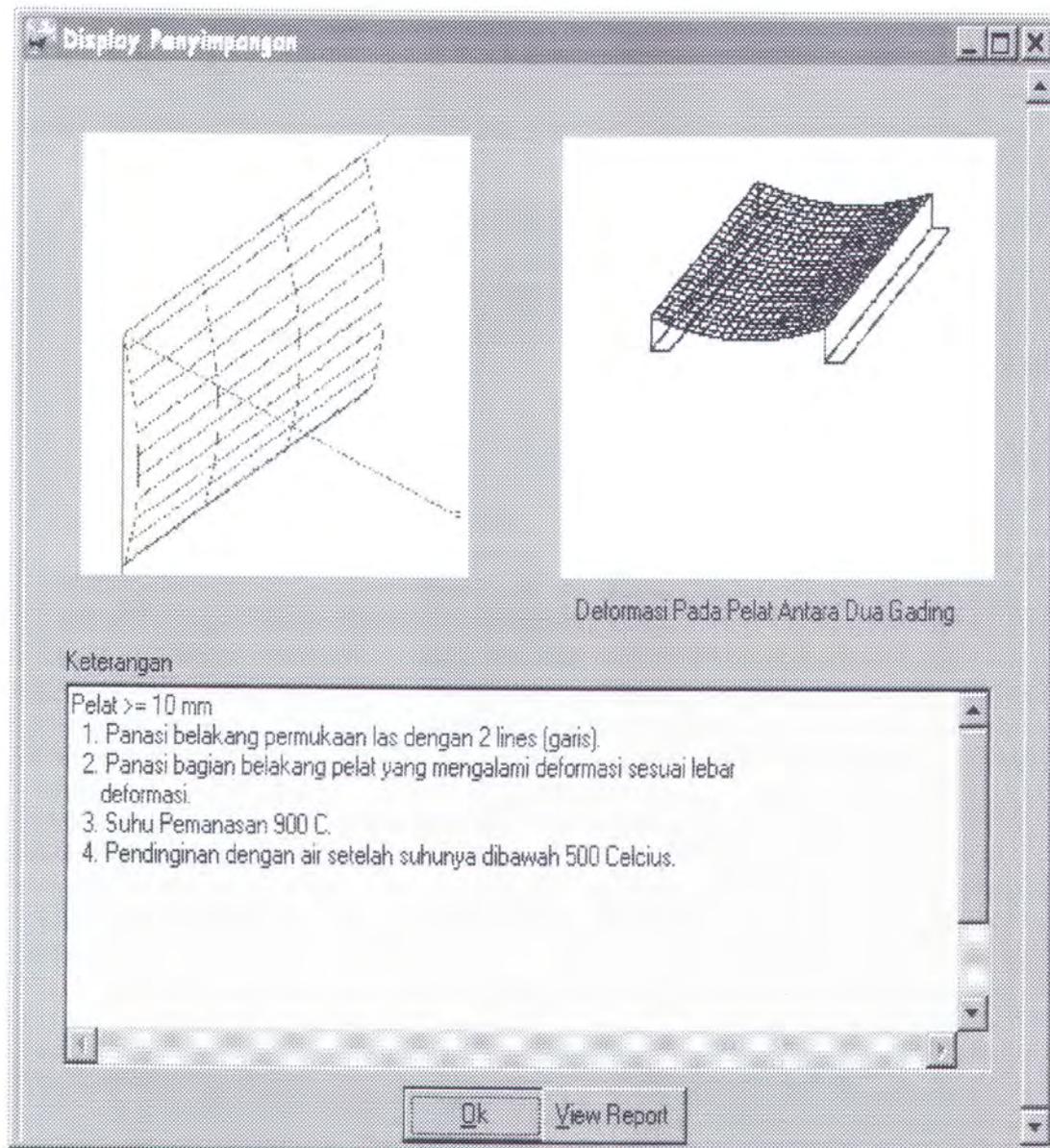
Langkah penggunaan :

- Memasukkan data desain
- Memasukkan data aktual
- Memasukkan lokasi penyimpangan
- Masukkan tebal pelat
- Tekan command button 'process' untuk memilih tipe penyimpangan yang sesuai



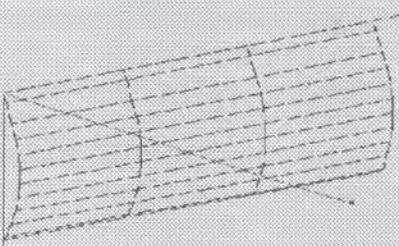
Gambar 6.9 Penggambaran grafis komponen 3-D

2. Memilih tipe penyimpangan yang sesuai dengan cara mengklik salah satu gambar tipe penyimpangan, kemudian mengklik command button 'Select' untuk melihat instruksi perbaikan.



Gambar 6.10 Tampilan Instruksi Perbaikan

3. Dengan mengklik command button 'View Report' maka akan ditampilkan sebuah form yang berisi sheet hasil pemeriksaan beserta lokasi penyimpangan, jenis penyimpangan dan instruksi perbaikan.

ITS Surabaya		Instruksi Perbaikan		Nama Kapal : OI-510	
Tahap : ASSEMBLY		Nomor Sekel :		DWT : 45 000 ton	
				Tanggal :	
					
Lokasi Penyimpangan : Pelat Lambung Bagian Sisi Paralel Jenis Penyimpangan : Deformasi Pada Pelat Antara Dua Gading Langkah Perbaikan Pelat >= 10 mm 1. Panas belakang permukaan las dengan 2 lines (gates) 2. Panas bagian belakang pelat yang mengalami deformasi sesuai lebar deformasi 3. Suhu Pemanasan 800 C. 4. Pendinginan dengan air setelah sukunya dibawah 500 Celcius					
Koordinat Pemanasan Z1 = 0 Z2 = 90 Z3 = 180 Z4 = 290 Z5 = 360 Z6 = 480 Z7 = 600 Z8 = 690 Z9 = 800					

Printed using Naval Architecture Error-Detecting Program

by Mr. Joesli Taher

Gambar 6.10 Report Instruksi Perbaikan

## BAB VII

### DISKUSI DAN REKOMENDASI

Pembangunan kapal merupakan pekerjaan yang memiliki variasi pekerjaan yang sangat kompleks. Setiap pekerjaan pada tahap-tahap produksi kapal harus senantiasa dikontrol, karena setiap pekerjaan yang ada selalu berhubungan dengan pekerjaan yang menghasilkan panas. Panas yang ditimbulkan akibat dari pekerjaan dapat menyebabkan terjadinya penyimpangan. Penyimpangan pada proses produksi kapal harus dilakukan perbaikan dengan cepat dan tepat agar meningkatkan efektifitas kerja dan mengurangi *rework*.

Di dalam tugas akhir ini penulis mencoba menyusun suatu uraian mengenai metode pemeriksaan penyimpangan bentuk komponen kapal dan perbaikannya. Metode ini menggunakan sistem komputerisasi dengan harapan bahwa pelaksanaannya dapat cepat dan hasil yang baik.

Metode pencarian penyimpangan yang dapat ditampilkan dalam bentuk grafis memerlukan dua data input, yaitu data desain dan data aktual. Data desain digunakan sebagai dasar perhitungan penyimpangan yang terjadi. Sedangkan data aktual diperoleh dari dari komponen yang telah jadi menggunakan alat tertentu. Output dari dari pencarian penyimpangan ini adalah untuk mengidentifikasi penyimpangan yang terjadi. Apabila penyimpangan masih di dalam batas toleransi maka penyimpangannya diabaikan, sedangkan apabila penyimpangan di luar batas toleransi maka harus dilakukan perbaikan. Langkah perbaikan yang dilakukan adalah dengan *me-matching*-kan penyimpangan yang terjadi ke dalam

*database penyimpangan*. Sehingga output dari program ini adalah standarisasi langkah-langkah perbaikan terhadap penyimpangan bentuk komponen kapal disertai tampilan grafisnya.

Hasil yang diperoleh dari program ini masih terdapat beberapa kekurangan anatara lain :

- Data desain yang diperlukan oleh program ini sangat tergantung dari ketelitian pembacaan gambar rencana maupun gambar kerja.
- Pembacaan data aktual sangat tergantung pada kemampuan alat ukur yang digunakan.
- Sulitnya pembacaan data komponen kapal dalam bentuk yang kompleks dengan jumlah titik koordinat 3-D yang sangat banyak.
- Tingkat ketepatan bentuk penggambaran grafis komponen tergantung pada jumlah titik koordinat yang digunakan.
- Variasi *database* penyimpangan yang masih sedikit serta langkah perbaikan masih dibatasi dengan metode pemanasan (*fairing*).

Kekurangan-kekurangan yang ada harus dilakukan penyempurnaan untuk pengembangan selanjunya. Oleh karena itu perlu dilakukan pengkajian lebih lanjut agar pengembangan selanjutnya dapat lebih baik lagi. Saran yang dapat penulis berikan adalah :

- Pembacaan data desain menggunakan sistem terpadu *database* antara bagian perencanaan dan bagian produksi (*manufacturing*) agar data desain langsung dapat digunakan tanpa perlu pembacaan manual.

- Pembacaan data aktual menggunakan suatu alat yang mempunyai akurasi yang tinggi, misalnya peralatan optik (laser).
- Dalam penggunaan nyata di lapangan kedua cara pembacaan data input di atas digabungkan menjadi suatu sistem yang terpadu beserta *interface*-nya masing-masing, untuk pencarian penyimpangan yang lebih kompleks sehingga hasil yang diperoleh lebih optimal.

## BAB VIII

### KESIMPULAN

#### VIII.1 KESIMPULAN

Pada pembangunan kapal yang telah menerapkan sistem blok dimana semua proses produksi sudah terintegrasi, diperlukan suatu pengawasan pada setiap proses produksinya. Pengawasan ini dilakukan agar mutu dari kapal tetap terjaga sesuai dengan standar yang telah ditetapkan. Pemeriksaan terhadap penyimpangan bentuk dengan menggunakan sistem komputerisasi merupakan salah satu upaya agar keakurasian pemeriksaan dapat dilaksanakan dengan cepat dan hasil yang baik.

Penyimpangan bentuk yang terjadi pada setiap proses produksi harus dilakukan perbaikan. Deformasi merupakan salah satu dari bentuk penyimpangan akibat proses pengerjaan pada material. Jenis penyimpangan akibat deformasi ini dapat terjadi pada setiap tahap produksi, khususnya pada tahap *assembly*. Metode perbaikan penyimpangan akibat deformasi dapat dilakukan dengan cara termal dan mekanis serta gabungan antara keduanya. Perbaikan dengan cara termal yang dikenal dengan *fairing* lebih banyak digunakan dari pada dengan cara mekanis dengan pertimbangan bahwa perbaikan dengan cara termal, hasilnya lebih baik dan mudah dilaksanakan. Dengan demikian akan lebih meningkatkan produktivitas karena *rework* dapat ditekan seminimal mungkin.

Dengan demikian agar *rework* dapat ditekan seminimal mungkin penggunaan komputer untuk melakukan pengawasan terhadap penyimpangan bentuk dan standarisasi perbaikan akan lebih cepat dilakukan.

## VIII.2 SARAN

Untuk pengembangan lebih lanjut masih diperlukan beberapa penyempurnaan sebagai berikut :

- Pembacaan data desain menggunakan sistem terpadu *database* antara bagian perencanaan dan bagian produksi agar data desain langsung dapat digunakan untuk pembacaan manual.
- Menggabungkan pembacaan data input menjadi suatu sistem yang terpadu beserta *interface*-nya masing-masing sehingga hasil yang diperoleh akan lebih baik.

## DAFTAR PUSTAKA

1. Bowers, David S, *From Data to Data*, Chapman & Hall, London, New York, Tokyo, Melbourne, 1993
2. Chirillo, Louis, *Line Heating*, US Departement of Transportation, Maritime Administration in Cooperation with Todd Pacific Shipyards Corporation, 1982
3. Jones, JA, *Database in Theory and Practice*, Chapman & Hall, London, New York, Tokyo, Melbourne, First Edition, 1986
4. Okianto, Dani, *Paduan Belajar Borland Delphi 3.0*, Elex Media Komputindo, Jakarta, 1997
5. Okumora, Wiryosumarto, *Teknologi Pengelasan Logam*, Pradnya Paramita, Jakarta 1996
6. Soejitno, *Teknik Produksi Kapal*, ITS, Surabaya
7. Storch, Richard Lee, *Ship Production*, Cornel Maritim Press, Inc, Maryland USA, 1988
8. PT PAL INDONESIA, *Japanesse Shipbuilding Quality Standard, Qauality Assurance Manual*, 1980
9. The Society of Naval Architects of Japan, *Distortion Fairing*, Japan International Cooperation, Maritim Technology and Safety Bureu, Ministry of Transport Overseas Ship Building Cooperation Centre

10. Wing, Wiryanto, *Studi Pengaruh Pemakaian Line Heating Terhadap Kekuatan Mekanis Pelat Baja SS 41 Produksi PT Krakatau Steel Yang Dipakai Pada Pembuatan Kapal*, ITS, Surabaya, 1994



## LAMPIRAN A

### PEMBACAAN DATA DESAIN

Pelaksanaan pembacaan data desain memerlukan beberapa gambar yaitu; gambar kerja, body plan, dan bukaan kulit. Adapun cara-cara pembacaan data desain dengan adalah sebagai berikut :

#### □ **Pelat Datar (panel)**

- Letak vertikal

Pembacaan data desain pelat datar vertikal:

- Menentukan titik origin pada ujung pelat dengan posisi paling dekat dengan base line dan AP.
- Nilai koordinat y semua titik adalah nol.
- Pembacaan nilai koordinat x sesuai dengan letak titik sepanjang sumbu x (memanjang kapal).
- Semua nilai koordinat z diperlukan dari gambar bantuan *body plan* kapal dan bukaan kulit. Dari gambar bukaan kulit diperoleh posisi pelat datar pada lambung kapal. Kemudian pelat datar tersebut digambarkan pada *body plan* sesuai dengan letaknya pada gambar bukaan kulit tersebut, dari sini dapat dibaca nilai z.

- Letak horisontal

Pembacaan data desain pelat datar horisontal:

- Menentukan titik origin pada ujung pelat dengan posisi paling dengan base line dan AP.
- Pembacaan nilai koordinat y sesuai dengan letak titik sepanjang sumbu y (melintang kapal).
- Pembacaan nilai koordinat x sesuai dengan letak titik sepanjang sumbu x (memanjang kapal).
- Nilai koordinat z semua titik adalah nol.

□ **Pelat Lengkung (Simetris dan Tidak Simetris)**

Pembacaan data desain pelat lengkung :

- Menentukan titik origin pada ujung pelat dengan posisi paling dekat dengan base line dan AP
- Pembacaan nilai koordinat 3-D (x,y,z) berdasarkan letak masing-masing komponen pada lambung kapal (dari gambar bukaan kulit) yang digambarkan pada *body plan* kapal.

□ **Inner Part**

- Letak Melintang Kapal
  - Menentukan titik origin pada salah satu ujung pelat
  - Pembacaan nilai koordinat y sesuai dengan letak sepanjang sumbu y (melintang kapal)

- Nilai koordinat x semua titik adalah nol
- Pembacaan nilai koordinat z sesuai dengan letak sepanjang sumbu z (vertikal kapal).
- Letak Memanjang Kapal
  - Menentukan titik origin pada salah satu ujung pelat.
  - Nilai koordinat y semua titik adalah nol
  - Pembacaan nilai koordinat x sesuai dengan letak titik sepanjang sumbu x (memanjang kapal).
  - Pembacaan nilai koordinat z sesuai dengan letak titik sepanjang sumbu z (vertikal kapal).

## LAMPIRAN B

### LISTING PROGRAM

```

unit ScreenThreeDLab;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls, Spin, Menus, Grids,
  ComCtrls, Buttons;

const MaxArray = 300;
      MaxJoint = 1000;
      MaxElement = 300;
      Pembagi = 500;

type
  TSelection = (IsNone, IsJoint, IsElement);
  TAction =
    (Zooming, Rotating, Translating, InfoData);
  TPointXYZ = record
    x, y, z : double;
  end;

TFormLab3D = class(TForm)
  Panel3DLab: TPanel;
  Image: TImage;
  MainMenu1: TMainMenu;
  File1: TMenuItem;
  OpenData1: TMenuItem;
  OpenFileDialog1: TOpenDialog;
  Exit1: TMenuItem;
  PopupMenu1: TPopupMenu;
  Zoom1: TMenuItem;
  Rotate1: TMenuItem;
  Geser1: TMenuItem;
  Info1: TMenuItem;
  Label1: TLabel;
  Label2: TLabel;
  Desain: TStringGrid;
  Factual: TStringGrid;
  NewProject1: TMenuItem;
  Help1: TMenuItem;
  Help2: TMenuItem;
  N2: TMenuItem;
  About1: TMenuItem;
  SaveProject1: TMenuItem;
  SaveDialog1: TSaveDialog;
  X: TLabel;
  Label3: TLabel;
  Node: TLabel;
  Label5: TLabel;
  Label4: TLabel;
  Label6: TLabel;
  Label7: TLabel;
  Label8: TLabel;
  ComboBox1: TComboBox;
  Label9: TLabel;
  Button2: TButton;
  Edit1: TEdit;
  RadioButton1: TRadioButton;
  RadioButton2: TRadioButton;
  RadioButton3: TRadioButton;
  Panel1: TPanel;
  RadioButton4: TRadioButton;
  Label12: TLabel;
  Label11: TLabel;
  Label13: TLabel;
  OpenDataFaktual1: TMenuItem;
  SaveDataDesain1: TMenuItem;
  SaveDataFaktual1: TMenuItem;
  StatusBar1: TStatusBar;
  Label14: TLabel;
  Display1: TMenuItem;
  DatabasePenyimpangan1: TMenuItem;
  ScrollBar1: TScrollBar;
  ScrollBar2: TScrollBar;
  Label10: TLabel;
  Label15: TLabel;
  BitBtn1: TBitBtn;
  ScrollBar3: TScrollBar;
  Label16: TLabel;
  BitBtn2: TBitBtn;
  BitBtn3: TBitBtn;
  RadioButton5: TRadioButton;
  Label17: TLabel;
  Edit2: TEdit;
  Edit3: TEdit;
  Label18: TLabel;
  Label19: TLabel;
  Label20: TLabel;
  Label21: TLabel;
  Refresh: TButton;
  Label22: TLabel;
  NomorNode1: TMenuItem;
  ProgressBar1: TProgressBar;
  Timer1: TTimer;
  Panel2: TPanel;
  SpeedButton2: TSpeedButton;
  SpeedButton1: TSpeedButton;
  SpeedButton3: TSpeedButton;
  OpenFileDialog2: TOpenDialog;
  OpenProject1: TMenuItem;
  N3: TMenuItem;
  N4: TMenuItem;

```

```

    procedure FormCreate(Sender:
    TObject);
    procedure OpenData1Click(Sender:
    TObject);
    // procedure Joint1Click(Sender:
    TObject);
    // procedure Element1Click(Sender:
    TObject);
    procedure Exit1Click(Sender: TObject);
    procedure ImageMouseDown(Sender:
    TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
    procedure Zoom1Click(Sender:
    TObject);
    procedure Rotate1Click(Sender:
    TObject);
    procedure Geser1Click(Sender:
    TObject);
    procedure ImageMouseUp(Sender:
    TObject; Button: TMouseButton;
    Shift: TShiftState; X, Y: Integer);
    procedure ImageMouseMove(Sender:
    TObject; Shift: TShiftState; X,
    Y: Integer);
    procedure Info1Click(Sender: TObject);
    procedure FormShow(Sender: TObject);
    procedure FormClose(Sender: TObject;
    var Action: TCloseAction);
    procedure NewProject1Click(Sender:
    TObject);
    procedure SaveProject1Click(Sender:
    TObject);
    procedure Button2Click(Sender:
    TObject);
    procedure RefreshClick(Sender:
    TObject);
    procedure ComboBox1Click(Sender:
    TObject);
    procedure RadioButton1Click(Sender:
    TObject);
    procedure RadioButton5Click(Sender:
    TObject);
    procedure RadioButton2Click(Sender:
    TObject);
    procedure RadioButton3Click(Sender:
    TObject);
    procedure RadioButton4Click(Sender:
    TObject);
    procedure
    OpenDataFaktual1Click(Sender: TObject);

```

```

    procedure SaveDataDesain1Click(Sender:
    TObject);
    procedure SaveDataFaktual1Click(Sender:
    TObject);
    procedure FactualEnter(Sender: TObject);
    procedure DrawAxis;
    procedure ScrollBar1Scroll(Sender: TObject;
    ScrollCode: TScrollCode;
    var ScrollPos: Integer);
    procedure BitBtn1Click(Sender: TObject);
    procedure ScrollBar2Change(Sender: TObject);
    procedure ScrollBar3Change(Sender: TObject);
    procedure ScrollBar2Scroll(Sender: TObject;
    ScrollCode: TScrollCode;
    var ScrollPos: Integer);
    procedure ScrollBar3Scroll(Sender: TObject;
    ScrollCode: TScrollCode;
    var ScrollPos: Integer);
    procedure Timer1Timer(Sender: TObject);
    procedure BitBtn2Click(Sender: TObject);
    procedure BitBtn3Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure About1Click(Sender: TObject);
    procedure SpeedButton2Click(Sender:
    TObject);
    procedure SpeedButton1Click(Sender:
    TObject);
    procedure SpeedButton3Click(Sender:
    TObject);
    procedure OpenProject1Click(Sender:
    TObject);
    procedure DesainKeyPress(Sender: TObject;
    var Key: Char);
    procedure FactualKeyPress(Sender: TObject;
    var Key: Char);
    procedure OpenFileDialog2Show(Sender:
    TObject);
    private
    { Private declarations }
    AngleX, AngleY, ToCamera : Integer;
    Action3D : TAction;
    FirstClick, AutoX, AutoY : Boolean;
    SaveX, SaveY : LongInt;
    FileOpen : Boolean;
    LastSelect : TSelection;
    SaveData : Integer;
    TempScroll, TempScroll2, TempScroll3 :
    integer;
    procedure ShowFigure;
    procedure DrawNode(No : integer; Mode :
    Boolean);

```

```

    procedure DrawError;
    procedure GetMaximum (XYZ :
integer);
    function CheckXY : integer;
    function GetFileName:string;
    procedure LoadTextID;
    public
    { Public declarations }
    Joint : Array [1..MaxJoint] of
TPointXYZ;
    JmlJoint, JmlElement : Integer;
    Node1,Node2 : integer;
    NewClick : boolean;
end;

```

```

var
    FormLab3D: TFormLab3D;

TebalPelat,NodeA,NodeB,CountError,IMa
x,StartNode,LastNode,LastNode2 : integer;
    TempList : TStringGrid;

```

```

ErrorX,ErrorY,ErrorZ,ComboClick,Project
Click,Saveproject : boolean;
    Error2X,Error2Y,Error2Z,OpenClick :
boolean;
    ErrorData : array[1..20] of integer;
    ErrorPoint,HeatPoint: array[1..20] of
double;
    ListErr : TStringList;
    Max : double;

```

```

implementation
{SR *.DFM}
USES
    GraphicsMathLibrary, {TMatrix}
    GraphicsPrimitivesLibrary,
{TPantoGraph}
    Info, UDataUtama, Error, Splash,
About; {DrawCube}
//Data Input for New Project
var Pantograph : TPantoGraph;

```

```

procedure
TFormLab3D.FormCreate(Sender:
TObject);
    VAR
    Bitmap: TBitmap;
begin
    Panel3DLab.Width:=Screen.Width-350;

```

```

    Panel3DLab.Height:=Panel3DLab.Width-100;
    Image.Width:=Panel3DLab.Width-15;
    Image.Height:=Panel3DLab.Width-125;
    Bitmap := TBitmap.Create;
    Bitmap.Width := Image.Width;
    Bitmap.Height := Image.Height;
    Image.Picture.Graphic := Bitmap;
    Action3D:=Zooming;
    AngleX:=20;
    AngleY:=15;
    ToCamera :=75;
    CentX:=0.25;
    CentY:=0.45;
    SaveX:=0; SaveY:=0;
    FileOpen:=False;
    ErrorX:=false;
    ErrorY:=false;
    ErrorZ:=false;
    CountError:=1;
    ProjectClick:=false;
end;

```

```

procedure TFormLab3D.DrawError;
var
    i,j : integer;
    BitMap2 : TBitMap;
begin
    BitMap2:=TBitMap.Create;

ErrorDisplay.Image2.Picture.Graphic:=BitMap2;
    Action3D:=Zooming;
    AngleX:=20;
    AngleY:=15;
    ToCamera :=55;
    CentX:=0.25;
    CentY:=0.45;
    Pantograph.SetColor(clRed);
    with ErrorDisplay.Image2 do
    for i:=1 to 20 do
    begin
        j:=ErrorData[i];
        if j<>0 then
            begin

```

```

                Pantograph.MoveTo(Vector3D(Joint[j].x/Pembagi
                ,Joint[j].y/Pembagi,Joint[j].z/Pembagi));

```

```

                Pantograph.LineTo(Vector3D(Joint[j].x/Pembagi,
                Joint[j].y/Pembagi,Joint[j].z/Pembagi));
            end
        end
    end
end

```

```

end;

procedure TFormLab3D.DrawAxis;
var
  i: integer;
  XYZPoint : array[1..3] of TPointXYZ;
begin
  XYZPoint[1].x:=0;
  XYZPoint[1].y:=4;
  XYZPoint[1].z:=0;
  XYZPoint[2].x:=3;
  XYZPoint[2].y:=0;
  XYZPoint[2].z:=0;
  XYZPoint[3].x:=0;
  XYZPoint[3].y:=0;
  XYZPoint[3].z:=3;

  PantoGraph.TextOut(Vector3D(XYZPoint
[1].x+10/Pembagi,XYZPoint[1].y+10/Pem
bagi,XYZPoint[1].z+10/Pembagi),'Y');

  PantoGraph.TextOut(Vector3D(XYZPoint
[2].x+10/Pembagi,XYZPoint[2].y+10/Pem
bagi,XYZPoint[2].z+10/Pembagi),'X');

  PantoGraph.TextOut(Vector3D(XYZPoint
[3].x+10/Pembagi,XYZPoint[3].y+10/Pem
bagi,XYZPoint[3].z+10/Pembagi),'Z');

  for i:=1 to 3 do
  begin
    Pantograph.MoveTo(Vector3D(0,0,0));

    Pantograph.LineTo(Vector3D(XYZPoint[i
].x,XYZPoint[i].y,XYZPoint[i].z));
  end;
end;

PROCEDURE TFormLab3D.ShowFigure;
VAR
  a : TMatrix;
  i,j,k : integer;
  TempStr1,tempStr2 : string;
  temp : TPointXYZ;
  Node : array[1..2] of integer;
BEGIN
// If FileOpen then begin
  WITH Image DO
    Canvas.FillRect( Rect(0,0, Width,
Height) );

```

```

{Use whole canvas as viewport}
pantograph.ViewPort (0.00,1.00, 0.00,1.00);
a := ViewTransformMatrix(
  coordSpherical,
  ToRadians(AngleX),
  ToRadians(AngleY),
  40,
  4,
  4,
  ToCamera);
pantograph.WorldCoordinatesRange (0.0, 1.0,
0.0, 1.0);
pantograph.SetTransform(a);
PantoGraph.SetColor(clRed);
DrawAxis;
for i:=1 to JmlJoint do
begin
  PantoGraph.SetColor(clBlue);

  PantoGraph.Canvas.Font:=FormLab3D.Font;
  PantoGraph.Canvas.Font.Size:=7;
  Temp.x:=Joint[i].x/Pembagi;
  Temp.y:=Joint[i].y/Pembagi;
  Temp.z:=Joint[i].z/Pembagi;
  TempStr1 := Desain.Cells[4,i-1];
  TempStr2:=Desain.Cells[5,i-1];
  Node[1] :=StrToInt(TempStr1);
  Node[2] :=StrToInt(TempStr2);
  k:=Node[1];
  for k:=0 to 1 do
  begin

    PantoGraph.MoveTo(Vector3D(Joint[i].x/Pembagi
i,Joint[i].y/Pembagi,Joint[i].z/Pembagi));

    PantoGraph.LineTo(Vector3D(Joint[Node[k+1]].
x/Pembagi,Joint[Node[k+1]].y/Pembagi,Joint[No
de[k+1]].z/Pembagi));
  end;
  PantoGraph.SetColor(clBlue);
  PantoGraph.Canvas.Font:=FormLab3D.Font;
  PantoGraph.Canvas.Font.Size:=7;

  //PantoGraph.TextOut(Vector3D(Joint[i].x/Pemba
gi,Joint[i].y/Pembagi,Joint[i].z/Pembagi),IntToStr
(i))
  end;
  DrawNode(SaveData,False);
// end
END {ShowFigure};

```

```

// this function now run properly....
function
TFormLab3D.GetFileName:string;
var
  LengthFile,i : integer;
  FName,temp : string;
begin
  if OpenClick and (ProjectClick=false)
  then
    begin
      OpenFileDialog2.Execute
    end;

  FName:=OpenDialog2.FileName;
  LengthFile:=length(FName);
  for i:=1 to (LengthFile-3) do
    begin
      Temp:=Temp + FName[i]+ ";
    end;
  GetFileName :=temp;
  if Temp="" then exit;
end;

procedure
TFormLab3D.OpenData1Click(Sender:
TObject);
var
  fl          : textfile;
  i           : integer;
  Temp,xs,ys,zs,N1 : String;
  Error : Boolean;
begin
  Desain.RowCount:=1;
  Error:=true;
  OpenClick:=true;
  Temp:=GetFileName;
  if Temp<>"" then
    begin
      assignfile(fl,Temp+'DSG');
      Reset(fl);
      while not eof(fl) do begin
        read(fl,i);

Readln(fl,Joint[i].x,Joint[i].y,Joint[i].z,
N1);
      // Input Data to Grid
      xs := FloatToStr(Joint[i].x);
      ys := FloatToStr(Joint[i].y);
      zs := FloatToStr(Joint[i].z);
      Desain.Cells[0,i-1]:=IntToStr(i);

```

```

      Desain.Cells[1,i-1]:=xs;//Joint[i].;
      Desain.Cells[2,i-1]:=ys;
      Desain.Cells[3,i-1]:=zs;
      Desain.Cells[4,i-1]:= N1[2]+N1[3];
      Desain.Cells[5,i-1]:=N1[4]+ N1[5];
      if (N1[3]=' ') then Desain.Cells[4,i-
1]:=N1[2];
      if (N1[4]=' ') then Desain.Cells[5,i-
1]:=N1[5]+N1[6];
      if (N1[6]=' ') then Desain.Cells[5,i-
1]:=N1[5];
      Desain.RowCount:=Desain.RowCount +
1;

ProgressBar1.Position:=ProgressBar1.Position +
1;
    end;
    ProgressBar1.Position:=100;
    Desain.RowCount:=Desain.RowCount - 1;
    JmlJoint:=i;
    ProgressBar1.Position:=0;
  end
  else exit;
  closefile(fl);
  ShowFigure;
end;

procedure TFormLab3D.Exit1Click(Sender:
TObject);
begin
  FormLab3D.Close
end;

procedure
TFormLab3D.ImageMouseDown(Sender:
TObject; Button: TMouseButton;
Shift: TShiftState; X, Y: Integer);
var i,X1,Y1 : integer;
    u : TVector;
begin
  if Button=mbLeft then
    If (Action3D<>InfoData) then begin
      FirstClick:=True;
      SaveX:=X;
      SaveY:=Y
    end
  Else begin
    for i:=1 to JmlJoint do begin
      u:=Vector3D(Joint[i].x/Pembagi,Joint[i].y/Pemba
gi,Joint[i].z/Pembagi);

```

```

PantoGraph.GetCoord(u,X1,Y1);
If ((X<=X1+2) and (X>=X1-2)) and
((Y<=Y1+2) and (Y>=Y1-2))
then begin

```

```

InfoFrm.Left:=Panel3D.Lab.Left+X+15;

```

```

InfoFrm.Top:=Panel3D.Lab.Top+Y+15;
InfoFrm.JointLabel.Caption:='No

```

```

Joint : '+IntToStr(i);

```

```

InfoFrm.LabelX.Caption:='X :

```

```

'+FloatToStr(Joint[i].X);

```

```

InfoFrm.LabelY.Caption:='Y :

```

```

'+FloatToStr(Joint[i].Y);

```

```

InfoFrm.LabelZ.Caption:='Z :

```

```

'+FloatToStr(Joint[i].Z);

```

```

InfoFrm.Show;

```

```

Break

```

```

end

```

```

end;

```

```

end

```

```

end;

```

```

procedure

```

```

TFormLab3D.Zoom1Click(Sender:

```

```

TObject);

```

```

begin

```

```

Action3D:=Zooming

```

```

end;

```

```

procedure

```

```

TFormLab3D.Rotate1Click(Sender:

```

```

TObject);

```

```

begin

```

```

Action3D:=Rotating

```

```

end;

```

```

procedure

```

```

TFormLab3D.Geser1Click(Sender:

```

```

TObject);

```

```

begin

```

```

Action3D:=Translating

```

```

end;

```

```

procedure

```

```

TFormLab3D.ImageMouseUp(Sender:

```

```

TObject; Button: TMouseButton;

```

```

Shift: TShiftState; X, Y: Integer);

```

```

begin

```

```

FirstClick:=False;

```

```

If Action3D=InfoData then InfoFrm.Hide
end;

```

```

procedure

```

```

TFormLab3D.ImageMouseMove(Sender:

```

```

TObject; Shift: TShiftState; X,

```

```

Y: Integer);

```

```

begin

```

```

If FirstClick then begin

```

```

if (SaveX<>X) or (SaveY<>Y) then begin

```

```

Case Action3D of

```

```

Zooming : begin

```

```

ToCamera:=ToCamera+(SaveY-

```

```

Y) div 2;

```

```

if ToCamera<0 then

```

```

ToCamera:=0

```

```

end;

```

```

Rotating : begin

```

```

AngleX:=AngleX+(X-SaveX);

```

```

AngleY:=AngleY-(Y-SaveY);

```

```

end;

```

```

Translating : begin

```

```

CentX:=CentX+(X-

```

```

SaveX)/Image.Width;

```

```

CentY:=CentY-(Y-

```

```

SaveY)/Image.Height

```

```

end;

```

```

end

```

```

end;

```

```

ShowFigure;

```

```

SaveX:=X;

```

```

SaveY:=Y

```

```

end

```

```

end;

```

```

procedure TFormLab3D.Info1Click(Sender:

```

```

TObject);

```

```

begin

```

```

Action3D:=InfoData

```

```

end;

```

```

procedure TFormLab3D.FormShow(Sender:

```

```

TObject);

```

```

begin

```

```

WITH Image DO

```

```

Canvas.FillRect( Rect(0,0, Width, Height) );

```

```

pantograph :=

```

```

TPantoGraph.Create(Image.Canvas);

```

```

PantoGraph.SetClipping(False)

```

```

end;

```

```

procedure
TFFormLab3D.FormClose(Sender: TObject;
var Action: TCloseAction);
begin
  pantograph.free;
  Application.Terminate
end;

```

```

procedure TFormLab3D.DrawNode(No :
integer; Mode: Boolean);
begin
  If Mode then
  PantoGraph.SetColor(clBlue)
  else PantoGraph.SetColor(clYellow);

  PantoGraph.PointAt(Vector3D(Joint[No].x
/Pembagi,Joint[No].y/Pembagi,Joint[No].z/
Pembagi))
end;

```

```

procedure
TFFormLab3D.NewProject1Click(Sender:
TObject);
// Set the Desain Grid to editable
begin
  NewClick:=true;
  with fDataUtama do
  begin
    Edit1.Text:=' ';
    Edit2.Text:=' ';
    Edit3.Text:=' ';
    Edit4.Text:=' ';
    Edit5.Text:=' ';
    Edit6.Text:=' ';
    Edit7.Text:=' ';
    Edit8.Text:=' ';
    Edit9.Text:=' ';
  end;
  Desain.RowCount:=1;
  Factual.RowCount:=1;
  RefreshClick(Sender);
end;

```

```

procedure
TFFormLab3D.SaveProject1Click(Sender:
TObject);
var
  FName : String;
begin
  Saveproject:=true;
  SaveDialog1.Execute;

```

```

FName := SaveDialog1.FileName ;
fDatautama.Button1Click(Sender);
fDataUtama.Data1.SaveToFile(FName+'.IDD');
fDataUtama.ProjectID.SaveToFile(FName+
'.PRJ');
SaveDataDesain1Click(Sender);
SaveDataFaktual1Click(Sender);
fDataUtama.Data1.Free;
fDataUtama.ProjectId.Free;
SaveProject:=false;
end;

```

```

//procedure untuk draw error;
procedure GetError;

```

```

var
  i,j,k : integer;
  Point1,Point2 : double;
  TempErr,TempErr2 : double;
  PointErr,TempPoint : array[1..20] of double;//
maksima 20 titik error
begin

```

```

  ListErr := TStringList.Create;
  Point1:=HeatPoint[1];
  Point2:=HeatPoint[LastNode];
  TempErr:=HeatPoint[Imax];
  TempErr2:=HeatPoint[Imax];

  j:=0;
  while (TempErr>Point1) and ((TempErr-
Point1)>100)do
  begin
    j:=j+1;
    PointErr[j]:=TempErr-100;
    TempErr:=PointErr[j];
  end;

```

```

  i:=1;
  TempPoint[1]:=Point1;
  while j>0 do
  begin
    i:=i+1;
    TempPoint[i]:=PointErr[j];
    j:=j-1;
  end;
  i:=i+1;
  TempPoint[i]:=TempErr2;
  TempErr:=HeatPoint[Imax];

```

```

  while (TempErr<Point2) AND ((Point2-
TempErr)>50) do

```

```

begin
  i:=i+1;
  TempPoint[i]:=TempErr + 100;
  TempErr:=TempPoint[i];
end;
TempPoint[i]:=Point2;

//masukkan data-data titik error ke list
for k:=1 to i do
if Error2X then
  ListErr.Add('X' + IntToStr(k) + ' = : ' +
FloatToStr(TempPoint[k]))
else if Error2Y then ListErr.Add('Y' +
IntToStr(k) + ' = : ' +
FloatToStr(TempPoint[k]))
else ListErr.Add('Z' + IntToStr(k) + ' = : '
+ FloatToStr(TempPoint[k]));
end;

function TFormLab3D.CheckXY : integer;
var
  i,j,Count : integer;
begin
  StartNode:=0;
  Count:=0;
  j:=0;
  if ErrorZ then
    begin
      if (Factual.Cells[1,0]='0') AND
(Factual.Cells[1,1]='0') then
Error2Y:=true;
      if (Factual.Cells[2,0]='0') AND
(Factual.Cells[2,1]='0') then
Error2X:=true;
      end;
      if ErrorY then
        begin
          if (Factual.Cells[1,0]='0') AND
(Factual.Cells[1,1]='0') then Error2Z:=true;
          if (Factual.Cells[3,0]='0') AND
(Factual.Cells[3,1]='0') then
Error2X:=true;
          end
        end
      else //if ErrorX
        begin
          if (Factual.Cells[2,0]='0') AND
(Factual.Cells[2,1]='0') then Error2Z:=true;

```

```

      if (Factual.Cells[3,0]='0') AND
(Factual.Cells[3,1]='0') then Error2Y:=true;
      end;

      //get node awal error
      for i:=0 to Desain.RowCount-1 do
        begin
          if Error2X then
            if Factual.Cells[1,i] <> Desain.Cells[1,i] then
              begin
                Count:=i+1;
                CheckXY := i+1;
                j:=j+1;
              end;
            if (Error2Y) then //if Factual.Cells[2,i]<>
Desain.Cells[2,i] then
              begin
                CheckXY :=i+1;
                Count:=i+1;
                j:=j+1;
              end
            else //if Error2Z
              begin
                CheckXY :=i+1;
                Count:=i+1;
                j:=j+1;
              end
            end;//endfor
            StartNode:=Count-j;
          end;

          procedure TFormLab3D.GetMaximum (XYZ :
integer);
          var
            ErrorXYZ : array[1..MaxArray] of double;
            X1,X2 : Double;
            i : integer;
          begin
            i:=0;
            Max:=0;
            repeat
              X1:=StrToFloat(Desain.Cells[XYZ,i]);
              X2:=StrToFloat(Factual.Cells[XYZ,i]);
              ErrorXYZ[i+1]:=abs(X1-X2);
              //Get maximum
              if (ErrorXYZ[i+1]>Max) then
                begin
                  Max:=ErrorXYZ[i+1];
                  IMax :=i+1; //get max index

```



```

begin
  CountError:=CountError + 1;

HeatPoint[j]:=StrToFloat(Factual.Cells[3,i]
);
  Last :=CountError + 1; // Node
menyimpang
  j:=j+1;
  end;
  end;
  end;
//-----
-----
  if
(Desain.Cells[1,1]<>Factual.Cells[1,1])
AND (Factual.Cells[1,0]<>"") then
  begin
    ErrorX:=true;
    IndexError:=CheckXY;
    if (Error2Y) then
      begin
        GetMaximum(1);// on z axis
        j:=1;
        for i:=StartNode to LastNode do
          begin
            CountError:=CountError + 1;

HeatPoint[j]:=StrToFloat(Factual.Cells[2,i]
);
          Last :=CountError + 1; // Node
menyimpang
          j:=j+1;
          end;
          end;
// jika error pada z
if (Error2Z) then
  begin
    GetMaximum(1);// on z axis
    j:=1;
    for i:=StartNode to LastNode do
      begin
        CountError:=CountError + 1;

HeatPoint[j]:=StrToFloat(Factual.Cells[3,i]
);
      Last :=CountError + 1; // Node
menyimpang
      j:=j+1;
      end;
      end;
      end;
end;

```

```

if Max<StrToInt(Edit2.Text) then
begin
  Error:=false;
  MessageDlg('Tidak ada
penyimpangan',mtInformation,[mbok],0);
  exit;
end;
  GetError;
  ErrorDisplay.Image2.Picture:=Image.Picture;
  ErrorDisplay.ShowModal;
// end
end;

//procedure ShowErrorOnClick
//procedure ini akan menjalankan fungsi error
yang dihasilkan
// dari process...
procedure TFormLab3D.RefreshClick(Sender:
TObject);
var
  Error, Tolerance,i,j : integer;
begin
  //ShowFigure;
  fDataUtama.ShowModal;
end;

//bagian tengah
procedure
TFormLab3D.RadioButton1Click(Sender:
TObject);
begin
  ComboBox1.Clear;
  ComboBox1.Items[0]:='Pelat Lambung Bagian
Sisi Paralel'; //a>=6
  ComboBox1.Items[1]:='Pelat Lambung Bagian
Dasar Paralel'; // --
  ComboBox1.Items[2]:='Pelat Dasar Ganda';
  ComboBox1.Items[3]:='Geladak Kekuatan
Bagian Paralel'; //--
  ComboBox1.Items[4]:='Geladak Bangunan
Atas Bagian Terbuka';//--
  ComboBox1.Items[5]:='Geladak Kedua Bagian
Terbuka'; //a>8
  ComboBox1.Items[6]:='Geladak Kekuatan
Bagian Tertutup';//8
  ComboBox1.Items[7]:='Geladak Kedua Bagian
Terbuka'; //>=10

```

```

    ComboBox1.Items[8]='Geladak
    Bangunan Atas Bagian Tertutup';//>=10
    ComboBoxClick:=true;
end;

```

```

//Bagian depan
procedure
TFormLab3D.RadioButton3Click(Sender:
TObject);
begin
    ComboBox1.Clear;
    ComboBox1.Items[0]='Forecastle
    Bagian Terbuka';//if a>=6mm
    ComboBox1.Items[1]='Forecastle
    Deck'; //if a>=9mm
    ComboBox1.Items[2]='Pelat Lambung
    Haluan';//if a>=7mm
    ComboBoxClick:=true;
end;

```

```

//Bagian belakang..
procedure
TFormLab3D.RadioButton2Click(Sender:
TObject);
begin
    ComboBox1.Clear;
    ComboBox1.Items[0]='Poop Deck
    Bagian Terbuka'; //a>=8
    ComboBox1.Items[1]='Poop Deck
    Bagian Tertutup';//a>=9
    ComboBox1.Items[2]='Pelat Lambung
    Buritan';//a>=7
    ComboBoxClick:=true;
end;

```

```

//bagian dalam.....
procedure
TFormLab3D.RadioButton4Click(Sender:
TObject);
begin
    ComboBox1.Clear;
    ComboBox1.Items[0]='Dinding
    Pemisah Bagian Tertutup';//9
    ComboBox1.Items[1]='Dinding
    Pemisah Sisi Luar';//6
    ComboBox1.Items[2]='Dinding
    Pemisah Sisi Dalam';//--
    ComboBox1.Items[3]='Sekat
    Memanjang dan Melintang';//8

```

```

    ComboBox1.Items[4]='Dinding Pemisah
    Bagian Tertutup';//9
    ComboBox1.Items[5]='Cross Deck '; //a>=7
    ComboBoxClick:=true;
end;

```

```

procedure
TFormLab3D.RadioButton5Click(Sender:
TObject);
begin
    ComboBox1.Clear;
    ComboBox1.Items[0]='Girder';//8
    ComboBox1.Items[1]='Beam';//8
    ComboBox1.Items[2]='Stiffener';//--
    ComboBox1.Items[3]='Frame Transversal';//8
    ComboBox1.Items[4]='Pilar';//6
    ComboBox1.Items[5]='Profil';//3
    ComboBoxClick:=true;
end;

```

```

procedure
TFormLab3D.ComboBox1Click(Sender:
TObject);
var
    x,Tolerance : integer;
begin
    // if Bagian Tengah
    if (RadioButton1.Checked) then
        begin
            x :=ComboBox1.ItemIndex;
            case (x) of
                0..4 : Edit2.Text:='6';
                5,6 : Edit2.Text:='8';
                7,8 : Edit2.Text:='10';
            end;
        end;
end;

```

```

//if bagian depan , set tolerance
if (RadioButton2.Checked) then
begin
    x:=ComboBox1.ItemIndex;
    case (x) of
        0 : Edit2.Text:='6';
        1 : Edit2.Text:='9';
        2 : Edit2.Text:='7';
    end;
end;
end;

```

```

// bagian belakang
if (RadioButton3.Checked) then
begin

```

```

x:=ComboBox1.ItemIndex;
case (x) of
0 : Edit2.Text:='8';
1 : Edit2.Text:='9';
2 : Edit2.Text:='7';
end;
end;

//bagian dalam
if (RadioButton4.Checked) then
begin
x:=ComboBox1.ItemIndex;
case (x) of
0 : Edit2.Text:='9';
1 : Edit2.Text:='6';
2 : Edit2.Text:='6';
3 : Edit2.Text:='8';
4 : Edit2.Text:='9';
5 : Edit2.Text:='7';
end;
end;

// Komponen Penguat
if (RadioButton5.Checked) then
begin
x:=ComboBox1.ItemIndex;
case (x) of
0 : Edit2.Text:='8';
1 : Edit2.Text:='8';
2 : Edit2.Text:='8';
3 : Edit2.Text:='8';
4 : Edit2.Text:='6';
5 : Edit2.Text:='3';
end;
end;

Tolerance :=StrToInt(Edit2.Text);
end;

procedure
TFormLab3D.OpenDataFaktual1Click(Sender: TObject);
var
fl          : textfile;
i           : integer;
FName,temp,xs,ys,zs,N1 : String;
Error, ReadEnd : Boolean;
begin
Factual.RowCount:=1;
Error:=true;

```

```

//if OpenFileDialog1.Execute then begin
OpenClick:=true;
Temp:=GetFileName;
if Temp<>'' then
begin
assignfile(fl,Temp + 'fct');
Reset(fl);
while not eof(fl) do begin
read(fl,i);

Readln(fl,Joint[i].x,Joint[i].y,Joint[i].z,N1);
// Input Data to Grid
xs := FloatToStr(Joint[i].x);
ys := FloatToStr(Joint[i].y);
zs := FloatToStr(Joint[i].z);
Factual.Cells[0,i-1]:=IntToStr(i);
Factual.Cells[1,i-1]:=xs;//Joint[i].;
Factual.Cells[2,i-1]:=ys;
Factual.Cells[3,i-1]:=zs;
Factual.Cells[4,i-1]:= N1[2]+N1[3];
Factual.Cells[5,i-1]:=N1[4]+ N1[5];
if (N1[3]=' ') then Factual.Cells[4,i-1]:=N1[2];
if (N1[4]=' ') then Factual.Cells[5,i-1]:=N1[5]+N1[6];
if (N1[6]=' ') then Factual.Cells[5,i-1]:=N1[5];
Factual.RowCount:=Factual.RowCount +
1;

ProgressBar1.Position:=ProgressBar1.Position +
1;
end;
ProgressBar1.Position:=100;
Factual.RowCount:=Factual.RowCount - 1;
JmlJoint:=i;
ProgressBar1.Position:=0;
end
else exit;
closefile(fl);
ShowFigure
end;

procedure
TFormLab3D.SaveDataDesain1Click(Sender:
TObject);
var
TempSave : TStringList;
i : integer;
St : String;
begin

```

```

TempSave := TStringList.Create;
for i:=0 to Desain.RowCount-1 do
begin
  St := Desain.Cells[0,i] + '' +
Desain.Cells[1,i] + '' + Desain.Cells[2,i] +
'' +
'' + Desain.Cells[3,i] + '' +
Desain.Cells[4,i] + '' + Desain.Cells[5,i];
  TempSave.Add(St);
end;
if SaveProject=false then
begin
  SaveDialog1.Execute;

TempSave.SaveToFile(SaveDialog1.FileName + '.DSG');
  TempSave.Free;
end
else begin

```

```

TempSave.SaveToFile(SaveDialog1.FileName + '.DSG');
  TempSave.Free;
end
end;

```

```

procedure
TFormLab3D.SaveDataFaktual1Click(Sender: TObject);
var
  TempSave : TStringList;
  i : integer;
  St : string;
begin
  TempSave := TStringList.Create;
  for i:=0 to Factual.RowCount-1 do
  begin
    St:=Factual.Cells[0,i] + '' +
Factual.Cells[1,i] + '' + Factual.Cells[2,i] +
'' +
'' + Factual.Cells[3,i] + '' +
Factual.Cells[4,i] + '' + Factual.Cells[5,i];
    TempSave.Add(St);
  end;

```

```

if SaveProject=false then
begin
  SaveDialog1.Execute;

```

```

TempSave.SaveToFile(SaveDialog1.FileName + '.FCT');

```

```

TempSave.Free;
end
else begin
  TempSave.SaveToFile(SaveDialog1.FileName +
'.FCT');
  TempSave.Free;
end;
end;

```

```

procedure TFormLab3D.FactualEnter(Sender:
TObject);
begin
  Factual.RowCount:=Factual.RowCount + 1;
end;

```

```

procedure TFormLab3D.ScrollBar1 Scroll(Sender:
TObject;
  ScrollCode: TScrollCode; var ScrollPos:
Integer);
begin
  Action3D:=Zooming;
  if TempScroll<ScrollBar1.Position then
    ToCamera:=ToCamera +
(ScrollBar1.Position-TempScroll)
  else ToCamera:=ToCamera - (TempScroll-
ScrollBar1.Position);
  if ToCamera<0 then ToCamera:=0;
  TempScroll:=ScrollBar1.Position;
  ShowFigure;
end;

```

```

procedure TFormLab3D.BitBtn1Click(Sender:
TObject);
begin
  AutoX:=true;
  Timer1.Enabled:=true;
end;

```

```

procedure
TFormLab3D.ScrollBar2Change(Sender:
TObject);
begin
  Action3D:=Rotating;
  if TempScroll2<ScrollBar2.Position then
    AngleX:=AngleX + (ScrollBar2.Position-
TempScroll2)
  else AngleX:=AngleX - (TempScroll2-
ScrollBar2.Position);
  TempScroll2:=ScrollBar2.Position;
  ShowFigure;

```

```

end;

procedure
TFormLab3D.ScrollBar3Change(Sender:
TObject);
begin
  Action3D:=Rotating;
  if TempScroll3<ScrollBar3.Position then
    AngleY:=AngleY +
    (ScrollBar3.Position-TempScroll3)
  else AngleY:=AngleY - (TempScroll3-
    ScrollBar3.Position);
  TempScroll3:=ScrollBar3.Position;
  ShowFigure;
end;

```

```

procedure
TFormLab3D.ScrollBar2Scroll(Sender:
TObject);
  ScrollCode: TScrollCode; var ScrollPos:
Integer);
begin
  Action3D:=Rotating;
  if TempScroll2<ScrollBar2.Position then
    AngleX:=AngleX +
    (ScrollBar2.Position-TempScroll2)
  else AngleX:=AngleX - (TempScroll2-
    ScrollBar2.Position);
  TempScroll2:=ScrollBar2.Position;
  ShowFigure;
end;

```

```

procedure
TFormLab3D.ScrollBar3Scroll(Sender:
TObject);
  ScrollCode: TScrollCode; var ScrollPos:
Integer);
begin
  Action3D:=Rotating;
  if TempScroll3<ScrollBar3.Position then
    AngleY:=AngleY +
    (ScrollBar3.Position-TempScroll3)
  else AngleY:=AngleY - (TempScroll3-
    ScrollBar3.Position);
  TempScroll3:=ScrollBar3.Position;
  ShowFigure;
end;

```

```

procedure TFormLab3D.Timer1Timer(Sender:
TObject);
begin
  if (AngleX=360) or (AngleY=360) then
    Timer1.Enabled:=false;
  if AutoX then
    begin
      ScrollBar2.Position:=ScrollBar2.Position + 1;
      AngleX:=AngleX + 1;
      ShowFigure;
    end;

  if AutoY then
    begin
      ScrollBar3.Position:=ScrollBar3.Position + 1;
      AngleY:=AngleY + 1;
      ShowFigure;
    end
  end;
end;

```

```

procedure TFormLab3D.BitBtn2Click(Sender:
TObject);
begin
  Timer1.Enabled:=false;
  AutoX:=false;
  AutoY:=false;
end;

```

```

procedure TFormLab3D.BitBtn3Click(Sender:
TObject);
begin
  AutoY:=true;
  Timer1.Enabled:=true;
end;

```

```

procedure TFormLab3D.FormActivate(Sender:
TObject);
begin
  // SplashForm.Close;
end;

```

```

procedure TFormLab3D.About1Click(Sender:
TObject);
begin
  AboutForm.ShowModal;
end;

```

```

procedure
TFormLab3D.SpeedButton2Click(Sender:
TObject);

```

```

begin
  OpenData1Click(Sender);
end;

procedure
TFormLab3D.SpeedButton1Click(Sender:
TObject);
begin
  OpenDataFaktual1Click(Sender);
end;

procedure TFormLab3D.LoadTextID;
var
  Data1 : TStringList;
begin
  with fDataUtama do
  begin
    Data1:=TStringList.Create;
    Data1.LoadFromFile(GetFileName
+'IDD');
    Edit1.Text:=Data1.Strings[0];
    Edit2.Text:=Data1.Strings[1];
    Edit3.Text:=Data1.Strings[2];
    Edit4.Text:=Data1.Strings[3];
    Edit5.Text:=Data1.Strings[4];
    Edit6.Text:=Data1.Strings[5];
    Edit7.Text:=Data1.Strings[6];
    Edit8.Text:=Data1.Strings[7];
    Edit9.Text:=Data1.Strings[8];
  end
end;

procedure
TFormLab3D.SpeedButton3Click(Sender:
TObject);
var
  TempName : String;
begin
  ProjectClick:=true;
  NewClick:=false;
  if OpenFileDialog2.Execute then
  begin
    // GetFileName;
    LoadTextID;
    OpenData1Click(Sender);
    OpenDataFaktual1Click(Sender);
    ProjectClick:=false;
  end
  else exit;
end;

```

```

procedure
TFormLab3D.OpenProject1Click(Sender:
TObject);
begin
  SpeedButton3Click(Sender);
end;

procedure TFormLab3D.DesainKeyPress(Sender:
TObject; var Key: Char);
begin
  if ((Key=#13) AND (NewClick=true)) then
    Desain.RowCount:=Desain.RowCount + 1;
  if Key=#27 then
    Desain.RowCount:=Desain.RowCount-1;
end;

procedure TFormLab3D.FactualKeyPress(Sender:
TObject; var Key: Char);
begin
  if ((Key=#13) AND (NewClick=true)) then
    Factual.RowCount:=Desain.RowCount + 1;
  if Key=#27 then
    Factual.RowCount:=Factual.RowCount-1;
end;

procedure
TFormLab3D.OpenDialog2Show(Sender:
TObject);
begin
  OpenFileDialog2.FileName="";
end;

end.

unit UDataUtama;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls;

type
  TfDataUtama = class(TForm)
    Label1: TLabel;

```

```

Label2: TLabel;
Label3: TLabel;
Label4: TLabel;
Label5: TLabel;
Label6: TLabel;
Label7: TLabel;
Label8: TLabel;
Label9: TLabel;
Label10: TLabel;
Label11: TLabel;
Label12: TLabel;
Label13: TLabel;
Label14: TLabel;
Label15: TLabel;
Label16: TLabel;
Edit1: TEdit;
Edit2: TEdit;
Edit3: TEdit;
Edit4: TEdit;
Edit5: TEdit;
Edit6: TEdit;
Edit7: TEdit;
Edit8: TEdit;
Button1: TButton;
Label17: TLabel;
Label18: TLabel;
Edit9: TEdit;
procedure Button1Click(Sender:
TObject);
private
  { Private declarations }
public
  Data1 : TStringList;
  ProjectID : TStringList;
  { Public declarations }
end;

var
  fDataUtama: TfDataUtama;

implementation

{$R *.DFM}

procedure
TfDataUtama.Button1Click(Sender:
TObject);
begin
  Data1:=TStringList.Create;
  ProjectID:=TStringList.Create;
  try

```

```

ProjectID.Add('Test Project ');
Data1.Add(Edit1.Text);
Data1.Add(Edit2.Text);
Data1.Add(Edit3.Text);
Data1.Add(Edit4.Text);
Data1.Add(Edit5.Text);
Data1.Add(Edit6.Text);
Data1.Add(Edit7.Text);
Data1.Add(Edit8.Text);
Data1.Add(Edit9.Text);
finally
  ModalResult:=1;
  Close;
end;
end;
end.

unit Report;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  QrCtrls, quickrpt, ExtCtrls;

type
  TReportForm = class(TForm)
    QuickRep1: TQuickRep;
    QRImage1: TQRImage;
    QRShape1: TQRShape;
    QRShape2: TQRShape;
    QRShape3: TQRShape;
    QRShape4: TQRShape;
    QRShape5: TQRShape;
    QRShape6: TQRShape;
    QRShape7: TQRShape;
    QRLabel1: TQRLabel;
    QRLabel2: TQRLabel;
    QRLabel3: TQRLabel;
    QRLabel4: TQRLabel;
    QRLabel5: TQRLabel;
    QRLabel6: TQRLabel;
    QRLabel7: TQRLabel;
    QRLabel8: TQRLabel;
    QRShape8: TQRShape;
    QRShape9: TQRShape;

```

```

QRLabel9: TQRLabel;
QRLabel10: TQRLabel;
QRLabel11: TQRLabel;
QRLabel12: TQRLabel;
QRMemo1: TQRMemo;
Lokasi: TQRLabel;
Jenis: TQRLabel;
QRLabel13: TQRLabel;
QRLabel14: TQRLabel;
HeatPoint: TQRMemo;
NamaKapal: TQRLabel;
DWT: TQRLabel;
private
  { Private declarations }
public
  { Public declarations }
end;

var
  ReportForm: TReportForm;

implementation

{$R *.DFM}

end.

unit Error;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  StdCtrls, ExtCtrls;

type
  TErrorDisplay = class(TForm)
    Memo1: TMemo;
    Label1: TLabel;
    Button1: TButton;
    Image1: TImage;
    Splitter1: TSplitter;
    Image2: TImage;
    Button2: TButton;
    JenisDefo: TLabel;
  // procedure FormCreate(Sender:
  TObject);

    procedure Button1Click(Sender: TObject);
    procedure FormActivate(Sender: TObject);
    procedure Button2Click(Sender: TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

  // ErrorText = TStrings;

  var
    View : boolean;
    ErrorDisplay: TErrorDisplay;
    TextError : TStringList;

  implementation

  uses Display_Tipe, Report, ScreenThreeDLab,
  UDataUtama;

  {$R *.DFM}

  {procedure TErrorDisplay.FormCreate(Sender:
  TObject);
  var
    i,j: integer;
    Text : String;
  begin
    TextError:=TStringList.Create;
    TextError.LoadFromFile('err.txt');
    j:=TextError.Count;
    //Text:=TextError.Strings[i];
    for i:=0 to j-1 do
    begin
      Text:=TextError.Strings[i];
      Memo1.Lines[i]:=Text;
      //Memo1.Lines
      //inc(i)
    end
  end;}

  procedure TErrorDisplay.Button1Click(Sender:
  TObject);
  begin
    ModalResult:=1;
    TextError.Free;
    // TextError;
  end;

```

```

procedure
TErrorDisplay.FormActivate(Sender:
TObject);
begin
  if View then ErrorDisplay.Show
  else Form1.ShowModal;
end;

procedure
TErrorDisplay.Button2Click(Sender:
TObject);
var
  Error : TStringList;
  i,j,Index : integer;
  Text: String;
begin
  View :=true;
  Error:=TStringList.Create;
  try
  with ReportForm do
  begin
  case (FileText) of
    1 :
Error.LoadFromFile('C:\NED\ERROR\ERR
RIA.TXT');
    2 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R1B.TXT');
    3 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R1C.TXT');
    4 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R2A.TXT');
    5 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R2B.TXT');
    6 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R3A.TXT');
    7 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R3B.TXT');
    8 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R4A.TXT');
    9 :
Error.LoadFromFile('C:\NED\ERROR\ERR
R4B.TXT');
    10 :
Error.LoadFromFile('C:\NED\ERROR\ERR5A.T
XT');
    11 :
Error.LoadFromFile('C:\NED\ERROR\ERR5B.T
XT');
    12 :
Error.LoadFromFile('C:\NED\ERROR\ERR7A.T
XT');
    13 :
Error.LoadFromFile('C:\NED\ERROR\ERR7B.T
XT');
  end;

  j:=Error.Count;
  Index:=FormLab3D.ComboBox1.ItemIndex;
  Lokasi.Caption :=
FormLab3D.ComboBox1.Items[Index];
  Jenis.Caption :=
ErrorDisplay.JenisDefo.Caption;
  QRImage1.Picture:=
ErrorDisplay.Image2.Picture;
  NamaKapal.Caption:=fDataUtama.Edit1.Text;
  DWT.Caption :=fDataUtama.Edit2.Text;
  for i:=0 to j-1 do
  begin
  Text:=TextError.Strings[i];
  QRMemo1.Lines[i]:=Text
  end;
  for i:=0 to ListErr.Count-1 do
  begin
  Text:=ListErr.Strings[i];
  HeatPoint.Lines[i]:=Text;
  end;
  end
  finally
  ReportForm.QuickRep1.Preview;
  Error.Free;
  end;
end;
end.

unit Display_Tipe;

interface

uses

```

Windows, Messages, SysUtils, Classes,  
Graphics, Controls, Forms, Dialogs,  
StdCtrls, ExtCtrls, Buttons;

type

```
TForm1 = class(TForm)
  Panel1: TPanel;
  Image1: TImage;
  Image2: TImage;
  Image3: TImage;
  Image4: TImage;
  Image5: TImage;
  Image6: TImage;
  Image7: TImage;
  Image8: TImage;
  Image9: TImage;
  Image10: TImage;
  Button1: TButton;
  Bevel1: TBevel;
  Bevel2: TBevel;
  Bevel3: TBevel;
  Bevel4: TBevel;
  Bevel5: TBevel;
  Bevel6: TBevel;
  Bevel7: TBevel;
  Bevel8: TBevel;
  Bevel9: TBevel;
  Bevel10: TBevel;
  procedure Button1Click(Sender:
TObject);
  procedure Image1Click(Sender:
TObject);
  procedure Image2Click(Sender:
TObject);
  procedure Image3Click(Sender:
TObject);
  procedure Image4Click(Sender:
TObject);
  procedure Image5Click(Sender:
TObject);
  procedure Image6Click(Sender:
TObject);
  procedure Image7Click(Sender:
TObject);
  procedure Image8Click(Sender:
TObject);
  procedure Image9Click(Sender:
TObject);
  procedure Image10Click(Sender:
TObject);
  procedure Dissappear;
```

```
  procedure Label1MouseMove(Sender: TObject;
Shift: TShiftState; X,
  Y: Integer);
  procedure Panel1MouseMove(Sender: TObject;
Shift: TShiftState; X,
  Y: Integer);
  procedure Image1MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image2MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image3MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image4MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image5MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image6MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image7MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image8MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image9MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);
  procedure Image10MouseMove(Sender:
TObject; Shift: TShiftState; X,
  Y: Integer);

  private
    { Private declarations }
  procedure MemoClear;
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  FileText : integer;
implementation

{$R *.DFM}
```

```

uses
  Error,
  ScreenThreeDLab;

procedure TForm1.MemoClear;
var
  i : integer;
begin
  for i:=1 to 14 do
    ErrorDisplay.Memo1.Lines[i-1]:=' '
  end;

procedure TForm1.Button1Click(Sender:
TObject);
begin
  ModalResult:=1;
  ErrorDisplay.Button2.Enabled:=true;
  Close;
end;

procedure TForm1.Dissapear;
begin
  Bevel1.Visible:=false;
  Bevel2.Visible:=false;
  Bevel3.Visible:=false;
  Bevel4.Visible:=false;
  Bevel5.Visible:=false;
  Bevel6.Visible:=false;
  Bevel7.Visible:=false;
  Bevel8.Visible:=false;
  Bevel9.Visible:=false;
  Bevel10.Visible:=false;
end;

procedure TForm1.Image1Click(Sender:
TObject);
var
  i,j : integer;
begin
  ErrorDisplay.Image1.Picture:=Image1.Pict
ure;
  Dissapear;
  Bevel1.Visible:=true;
  with ErrorDisplay do
  begin
    JenisDefo.Caption:='Deformasi Pada
Pelat Antara Dua Gading';
    TextError:=TStringList.Create;
    TebalPelat:=StrToInt(FormLab3D.Edit1.Text);
    if (TebalPelat>10) OR (TebalPelat=10) then
      begin
        TextError.LoadFromFile('C:\NED\ERROR\ERR1
A.TXT');
        FileText:=1;
      end;
    if (TebalPelat>6) AND (TebalPelat<10) then
      begin
        TextError.LoadFromFile('C:\NED\ERROR\ERR1
B.TXT');
        FileText:=2;
      end
    else
      if TebalPelat<=5 then
        TextError.LoadFromFile('C:\NED\ERROR\ERR1
C.TXT');
        FileText:=3;
        j:=TextError.Count;
        MemoClear;
        for i:=0 to j-1 do
          begin
            Text:=TextError.Strings[i];
            Memo1.Lines[i]:=Text;
            //Memo1.Lines
          end
        end
      end;

procedure TForm1.Image2Click(Sender:
TObject);
var
  i,j : integer;
begin
  ErrorDisplay.Image1.Picture:=Image2.Picture;
  Dissapear;
  Bevel2.Visible:=true;
  with ErrorDisplay do
  begin
    JenisDefo.Caption:='Deformasi Pada Pelat
Sambungan Las Tumpul';
    TextError:=TStringList.Create;
    TebalPelat:=StrToInt(FormLab3D.Edit1.Text);
    if (TebalPelat>10) OR (TebalPelat=10) then
      begin
        TextError.LoadFromFile('C:\NED\ERROR\ERR2
A.TXT');

```

```

    FileText:=4;
  end
else
  TextError.LoadFromFile('C:\NED\ERROR
\ERR2B.TXT');
  j:=TextError.Count;
  FileText:=5;
  MemoClear;
  for i:=0 to j-1 do
  begin
    Text:=TextError.Strings[i];
    Memo1.Lines[i]:=Text;
  end
end
end;

```

```

procedure TForm1.Image3Click(Sender:
TObject);
var
  i,j : integer;
begin

```

```

  ErrorDisplay.Image1.Picture:=Image3.Pict
ure;
  Dissapear;
  with ErrorDisplay do
  begin
    JenisDefo.Caption:='Deformasi Pada
Face Plate Ke Arah Bawah';
    TextError:=TStringList.Create;

```

```

  TebalPelat:=StrToInt(FormLab3D.Edit1.T
ext);
  if (TebalPelat>10) OR (TebalPelat=10)
then
  begin

```

```

    TextError.LoadFromFile('C:\NED\ERROR
\ERR3A.TXT');
    FileText:=6;
  end
  else
    TextError.LoadFromFile('C:\NED\ERROR
\ERR3B.TXT');

```

```

    FileText:=7;
    j:=TextError.Count;
    MemoClear;
    for i:=0 to j-1 do
    begin
      Text:=TextError.Strings[i];

```

```

      Memo1.Lines[i]:=Text;
    end
  end
end;

```

```

procedure TForm1.Image4Click(Sender:
TObject);

```

```

var
  i,j : integer;
begin
  ErrorDisplay.Image1.Picture:=Image4.Picture;
  Dissapear;
  with ErrorDisplay do
  begin
    JenisDefo.Caption:='Deformasi Pada Face Plate
Ke Arah Bawah';
    TextError:=TStringList.Create;
    TebalPelat:=StrToInt(FormLab3D.Edit1.Text);
    if (TebalPelat>10) OR (TebalPelat=10) then
    begin

```

```

      TextError.LoadFromFile('C:\NED\ERROR\ERR4
A.TXT');
      FileText:=8;
    end
  else
    TextError.LoadFromFile('C:\NED\ERROR\ERR4
B.TXT');

```

```

      FileText:=9;
      j:=TextError.Count;
      MemoClear;
      for i:=0 to j-1 do
      begin
        Text:=TextError.Strings[i];
        Memo1.Lines[i]:=Text;
      end
    end
  end
end;

```

```

end;

procedure TForm1.Image5Click(Sender:
TObject);

```

```

var
  i,j : integer;
begin
  ErrorDisplay.Image1.Picture:=Image5.Picture;
  Dissapear;
  with ErrorDisplay do
  begin

```

```

    JenisDefo.Caption:= 'Deformasi Pada
Face Plate II';
    TextError:=TStringList.Create;

TebalPelat:=StrToInt(FormLab3D.Edit1.T
ext);
    if (TebalPelat>10) OR (TebalPelat=10)
then
    begin

TextError.LoadFromFile('C:\NED\ERROR
\ERR5A.TXT');
    FileText:=10;
    end
    else
TextError.LoadFromFile('C:\NED\ERROR
\ERR5B.TXT');

    FileText:=11;
    j:=TextError.Count;
    MemoClear;
    for i:=0 to j-1 do
    begin
    Text:=TextError.Strings[i];
    Memo1.Lines[i]:=Text;
    end
    end

end;

procedure TForm1.Image6Click(Sender:
TObject);
begin
    Image4Click(Sender);
    Dissappear;

ErrorDisplay.JenisDefo.Caption:='Deform
asi Pada Penguat II';

ErrorDisplay.Image1.Picture:=Image6.Pict
ure;;
end;

procedure TForm1.Image7Click(Sender:
TObject);
var
    i,j : integer;
begin

ErrorDisplay.Image1.Picture:=Image7.Pict
ure;;

```

```

Dissappear;
with ErrorDisplay do
begin
    JenisDefo.Caption:='Deformasi Shrinkage';
    TextError:=TStringList.Create;
    TebalPelat:=StrToInt(FormLab3D.Edit1.Text);
    if (TebalPelat>10) OR (TebalPelat=10) then
    begin

TextError.LoadFromFile('C:\NED\ERROR\ERR7
A.TXT');
    FileText:=12;
    end
    else
TextError.LoadFromFile('C:\NED\ERROR\ERR7
B.TXT');

    FileText:=13;
    j:=TextError.Count;
    MemoClear;
    for i:=0 to j-1 do
    begin
    Text:=TextError.Strings[i];
    Memo1.Lines[i]:=Text;
    end
    end

end;

procedure TForm1.Image8Click(Sender:
TObject);
begin
    Image4Click(Sender);
    Dissappear;
    ErrorDisplay.Image1.Picture:=Image8.Picture;;
    ErrorDisplay.JenisDefo.Caption:='Deformasi
Sudut Pada Profile';
end;

procedure TForm1.Image9Click(Sender:
TObject);
begin
    Image4Click(Sender);
    Dissappear;
    ErrorDisplay.Image1.Picture:=Image9.Picture;
    ErrorDisplay.JenisDefo.Caption:='Deformasi
Pada Flange';

end;

```

```

procedure TForm1.Image10Click(Sender:
TObject);
begin
  Image4Click(Sender);
  Dissapear;

```

```

ErrorDisplay.Image1.Picture:=Image10.Pic
ture;;

```

```

ErrorDisplay.JenisDefo.Caption:='Deform
asi Pada Flange II';
end;

```

```

procedure
TForm1.Label1MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  // BitBtn1.Visible:=true;
end;

```

```

procedure
TForm1.Panel1MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
end;

```

```

procedure
TForm1.Image1MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Bevel1.Visible:=true;
end;

```

```

{procedure
TForm1.Image1MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel1.Visible:=true;
end;
}

```

```

procedure TForm1.Image2MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel2.Visible:=true;
end;

```

```

procedure TForm1.Image3MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel3.Visible:=true;
end;

```

```

procedure TForm1.Image4MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel4.Visible:=true;
end;

```

```

procedure TForm1.Image5MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel5.Visible:=true;
end;

```

```

procedure TForm1.Image6MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel6.Visible:=true;
end;

```

```

procedure TForm1.Image7MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);
begin
  Dissapear;
  Bevel7.Visible:=true;
end;

```

```

procedure TForm1.Image8MouseMove(Sender:
TObject; Shift: TShiftState; X,
Y: Integer);

```

```

begin
  Dissappear;
  Bevel8.Visible:=true;
end;

procedure
 TForm1.Image9MouseMove(Sender:
 TObject; Shift: TShiftState; X,
 Y: Integer);
begin
  Dissappear;
  Bevel9.Visible:=true;
end;

procedure
 TForm1.Image10MouseMove(Sender:
 TObject; Shift: TShiftState; X,
 Y: Integer);
begin
  Dissappear;
  Bevel10.Visible:=true;
end;
end.

unit About;

interface

uses
  Windows, Messages, SysUtils, Classes,
  Graphics, Controls, Forms, Dialogs,
  ExtCtrls, StdCtrls;

type
  TAboutForm = class(TForm)
    Image1: TImage;
    Label1: TLabel;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    Button1: TButton;
    Bevel1: TBevel;
    procedure Button1Click(Sender:
 TObject);
  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  AboutForm: TAboutForm;

implementation

{$R *.DFM}

procedure TAboutForm.Button1Click(Sender:
 TObject);
begin
  ModalResult:=1;
  AboutForm.Close;
end;

end.

```



# JURUSAN TEKNIK PERKAPALAN

## FAKULTAS TEKNOLOGI KELAUTAN ITS

### SURAT KEPUTUSAN TUGAS AKHIR (NA 1701)

No. : 166 /PT12.FTK2/M/199 8

Nama Mahasiswa : Tri Yulianto  
Nomor Pokok : 4193100006  
Tanggal diberikan tugas : 25 September 1998  
Tanggal selesai tugas : 25 Februari 1999  
Dosen Pembimbing : 1. Ir. Sjariof Widjaja, Ph.D  
2. ....

#### Uraian / judul tugas akhir yang diberikan :

#PENERAPAN GRAFIK DATA BASE UNTUK STANDARISASI FAIRING WORK PENYIMPANGAN BENTUK-KOMPONEN KAPAL#

sOn

Surabaya, 28 September 1998  
Jurusan Teknik Perkapalan FTK-ITS  
Ketua,

#### Tembusan :

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

Ir. Koestowo Sastro Wiyono.  
NIP. 130 687 430.



**FAKULTAS TEKNOLOGI KELAUTAN ITS**  
**JURUSAN TEKNIK PERKAPALAN**

DAFTAR KEMAJUAN TUGAS AKHIR (NA 1701)

Nama mahasiswa : Tri Yulianto .....  
N.R.P. : 4193100006 .....  
Tugas diberikan : Semester Gasal 19 98 / 19 99 .....  
Tanggal mulai tugas : 25 September 1998 .....  
Tanggal selesai tugas : 25 Februari 1999 .....  
Dosen Pembimbing : 1. Ir. Sjariof Widjaja, Ph.D .....  
2. ....

Tanggal	Uraian Kemajuan Tugas	Tanda Tangan
20/09 98	Pembahasan tentang tema tugas akhir	
26/09 98	Diskusi tentang penyimpangan bentuk dan pembuatan data base	
10/10 98	Pembuatan graphic dengan menggunakan sistem vutek	
24/10 98	Jenis-jenis penyimpangan akibat deformasi dan perbaikan menggunakan line Heating	
31/10 98	Pembuatan program penggambaran Graphis secara tiga dimensi	
12/12 98	Pembuatan program database graphics dan teks	
26/12 98	Pengujian program dan Running Program.	
3/2 99	Pengcekan program	

lihat halaman berikutnya .....