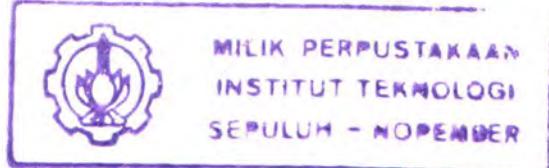


21.253/ITI/H/05



## TUGAS AKHIR (KP 1701)

### PROGRAM PERHITUNGAN PELUNCURAN END LAUNCHING KAPAL BESERTA ASPEK DINAMISNYA



RSpe  
623-83  
Arf  
P-1  
2003

Disusun oleh :

**YOSI ARFIANTO**  
NRP. 4197 100 027

PERPUSTAKAAN ITS	
Tgl. Terima	19-8-2003
Terima Dari	H/
No. Agenda Prp.	219045

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

# **PROGRAM PERHITUNGAN PELUNCURAN END LAUNCHING KAPAL BESERTA ASPEK DINAMISNYA**

## **TUGAS AKHIR**

**Diajukan Guna Memenuhi Sebagian Persyaratan Untuk  
Memperoleh Gelar Sarjana Teknik Perkapalan**

**Pada**  
**Jurusan Teknik Perkapalan**  
**Fakultas Teknologi Kelautan**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya**

**Surabaya, Agustus 2003**  
**Mengetahui / Menyetujui**

**Dosen Pembimbing I**



**I G. N. S. Buana, ST., M.Eng**

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

**PROGRAM PERHITUNGAN PELUNCURAN END  
LAUNCHING KAPAL BESERTA ASPEK  
DINAMISNYA**

**TUGAS AKHIR**

**Telah Direvisi Sesuai Dengan  
Hasil Sidang Ujian Tugas Akhir**

**Pada  
Jurusan Teknik Perkapalan  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember  
Surabaya**

**Surabaya, 4 Agustus 2003**

**Mengetahui dan Menyetujui**



**Dosen Pembimbing I**

**I G. N. S. Buana, ST., M. Eng.**

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

## ABSTRAK

Peluncuran (launching) adalah proses menurunkan kapal dari landasan peluncur ke permukaan air dengan menggunakan gaya berat atau dengan memberikan gaya dorong tambahan yang bekerja pada bidang miring. Perhitungan peluncuran ini diperlukan untuk menghindari kapal dari kondisi-kondisi kritis seperti tipping, dropping dan stern lift.

Secara tradisional, perhitungan peluncuran biasanya dibuat secara manual dan membutuhkan waktu relative lama. Oleh karena itu seiring dengan kemajuan teknologi, aplikasi computer dibuat untuk mempermudah dan mempercepat pelaksanaan perhitungan dan menampilkan diagram peluncuran. Dalam Tugas Akhir ini perhitungan peluncuran yang dilakukan adalah berdasarkan jenis peluncuran memanjang (end launching). Pembacaan kurva-kurva Bonjean menggunakan interpolasi polynomial newton. Prediksi penentuan berat kapal saat kapal diluncurkan dihitung dengan menggunakan pendekatan Lloyd Register '64. Data-data kapal dan perhitungan displacemen kapal yang dipergunakan dalam aplikasi ini adalah dari penelitian yang dilakukan sebelumnya.

Manfaat yang bias diambil dari program ini adalah effisiensi waktu, otomatisasi pembacaan kurva-kurva bonjean dan kurva-kurva Lloyd Register '64, keakurasian perhitungan dan pembuatan gambar lebih teliti, tingkat produksi gambarnya lebih cepat, tidak memerlukan pengulangan gambar, bentuk kurva lebih baik dan rapi, file input dapat disimpan, dan tidak memerlukan pengembangan besar dari diagram peluncuran yang ditampilkan.

## ABSTRACT

Launching is descending process from a slip way to water front using weight force or giving additional force that acts on a skewed plane. The calculation of this process is needed to avoid critical conditions on the ship, such as tipping, dropping and stern lift.

Traditionally, the calculation of launching process is made manually and time consuming. As the advanced development of computer and information technology, computer application is made to simplify and accelerate the calculation and to present the launching diagram. In this Final Project, the launching calculation based on end launching type. Newton polynomial interpolation method is used to read the curves of bonjean. Weight of the ship is calculated using Lloyd Register '64. Ship data and displacement calculation are provided previous research.

The advantages of this program are time efficiency, automation of bonjean and Lloyd Register '64 curves reading process, accurate calculation and better detail of images production, faster and no more redrawing of the same images, tidy and better curve shape, saveable file input, and the capability to produce many scales of the launching diagram.

## **KATA PENGANTAR**

Alhamdulillaahi Robbil ‘Alamiin. Segala puji bagi Alloh, Tuhan penguasa alam, hanya karena-Nyalah penyusunan program dan penulisan laporan dalam Tugas Akhir ini mampu penulis selesaikan. Walaupun banyak pihak telah memberikan bantuan dan bimbingan akan tetapi kesemuanya itu tidak lepas dari izin-Nya.

Dengan terselesaikannya penyusunan program dan penulisan laporan Tugas Akhir ini, penulis sampaikan terima kasih kepada :

1. Kedua orang tuaku dan adik-adikku tercinta atas seluruh jerih payah dan pengorbanannya serta do'a yang senantiasa mengiringi kehidupan penulis.
2. Bapak I G. N. S. Buana, ST., M.Eng. sebagai dosen pembimbing yang telah banyak memberikan waktu dan bimbingan selama penyusunan Tugas Akhir ini.
3. Rekan-rekan P-37 : Yusa, Umar, Basar, Danang, Andi, Helmi, Gito, Agung, Wahyu, Totok, Hasyim, Anang, Syaiful, Hanafi, serta rekan-rekan P-37 lainnya yang tidak mungkin disebutkan semuanya atas segala saran, kritikan, dan kerjasama yang terjalin selama penyelesaian Tugas Akhir ini.
4. Rekan-rekan Lantai 4 : Yusa, Umar, Basar, Ipung, Parcok, Toni, dan lainnya yang sering singgah di Lantai 4.

Penulis

## **DAFTAR ISI**

ABSTRAKSI	i
KATA PENGANTAR	iii
DAFTAR ISI	iv
DAFTAR GAMBAR	vi
DAFTAR TABEL	viii
DAFTAR NOTASI	ix

### **BAB I PENDAHULUAN**

1.1. Latar Belakang	I-1
1.2. Tujuan	I-3
1.3. Manfaat	I-3
1.4. Batasan Masalah	I-4
1.5. Metodologi Penyelesaian	I-5
1.6. Sistematika Penulisan Tugas Akhir	I-8

### **BAB II DASAR TEORI**

2.1. Pencocokan Kurva	II-1
2.2. Regresi Kuadrat Terkecil	II-2
2.3. Interpolasi	II-7
2.4. Pemilihan Metode Pencocokan Kurva	II-14
2.5. Peluncuran	II-15

### **BAB III PERHITUNGAN PELUNCURAN**

3.1. Perhitungan Berat dan Titik Berat	III-1
--	-------

3.2. Perhitungan Periode I	III-7
3.3. Perhitungan Periode II	III-8
3.4. Perhitungan Periode III	III-10
3.5. Perhitungan Periode IV	III-10
3.6. Perhitungan Aspek Dinamis	III-10

#### **BAB IV TEKNIK PEMROGRAMAN**

4.1. Teknik Pemrograman	IV-1
4.2. Panduan Pemakaian Program	IV-3

#### **BAB V ANALISIS HASIL PROGRAM**

5.1. Pembacaan Luasan Bonjean Kapal	V-2
5.2. Hasil Perhitungan Peluncuran dan	
Analisis Kondisi Kritis	V-4
5.3. Kecepatan Perolehan Hasil	V-5
5.4. Hasil Output Diagram Peluncuran	V-5

#### **BAB VI KESIMPULAN DAN SARAN**

6.1. Kesimpulan	VI-1
6.2. Saran Pengembangan	VI-2

#### **DAFTAR PUSTAKA**

<b>LAMPIRAN I Hasil Perhitungan Program</b>	Lamp I
<b>LAMPIRAN II Perhitungan Manual</b>	Lamp II
<b>LAMPIRAN III Listing Program</b>	Lamp III

## DAFTAR GAMBAR

Gambar 1.1 Alur metodologi penyelesaian_____	I-7
Gambar 2.1 Tiga usaha untuk mencocokkan sebuah kurva_____	II-2
Gambar 2.2 _____	II-4
Gambar 2.3 Indikasi bahwa parabola lebih cocok_____	II-6
Gambar 2.4 Polinomial orde pertama dan kedua_____	II-8
Gambar 2.5 Interpolasi linear_____	II-9
Gambar 2.6 Perbandingan dua interpolasi linear_____	II-10
Gambar 2.7 Interpolasi kuadratik_____	II-13
Gambar 2.8 Peluncuran <i>end launching</i> _____	II-16
Gambar 2.9 Periode I_____	II-19
Gambar 2.10 Pembebanan rata-rata Periode I_____	II-21
Gambar 2.11 Periode II_____	II-23
Gambar 3.1 Penurunan Persamaan Bonjean_____	III-9
Gambar 4.1 Tampilan awal program_____	IV-3
Gambar 4.2 Form input program_____	IV-4
Gambar 4.3 Open Database Bonjean_____	IV-5
Gambar 4.4 Tampilan data bonjean kapal_____	IV-6
Gambar 4.5 Tampilan persamaan bonjean kapal_____	IV-6
Gambar 4.6 Tampilan hasil perhitungan peluncuran_____	IV-7
Gambar 4.7 Tampilan hasil perhitungan dinamis_____	IV-7
Gambar 4.8 Tampilan diagram peluncuran pada AutoCAD_____	IV-8

Gambar 4.9 Tampilan aspek dinamis peluncuran	IV-9
Gambar 4.10 Error pada harga pembebanan	IV-10
Gambar 4.11 Error karena kapal kurang gaya dorong	IV-10
Gambar 5.1 Hasil diagram peluncuran	V-6
Gambar 5.2 Hasil aspek dinamis pada peluncuran	V-7

## **DAFTAR TABEL**

<i>Tabel 1.</i> Hasil program luasan bonjean	V-2
<i>Tabel 2.</i> Hasil manual luasan bonjean	V-2
<i>Tabel 3.</i> Hasil koreksi luasan bonjean	V-3
<i>Tabel 4.</i> Rekapitulasi Periode II program	V-4
<i>Tabel 5.</i> Rekapitulasi Periode II manual	V-4
<i>Tabel 6.</i> Koreksi rekapitulasi Periode II	V-5
<i>Tabel 7.</i> Database bonjean dalam MS. Access	Lampiran I-1
<i>Tabel 8.</i> Hasil Periode I	Lampiran I-2
<i>Tabel 9.</i> Hasil perhitungan tiap langkah Periode II	Lampiran I-2
<i>Tabel 10.</i> Hasil perhitungan dinamis kapal	Lampiran I-5
<i>Tabel 11.</i> Rekapitulasi berat dan titik berat kapal kosong	Lampiran II-1

## DAFTAR NOTASI

- P Berat total peluncuran
- G Titik berat kapal
- $\gamma V'$  Gaya apung yang bekerja pada kapal
- Q Gaya reaksi pada landasan
- a Jarak titik berat G ke ujung landasan
- b Jarak titik gaya apung  $\gamma V'$  ke ujung landasan
- c Jarak titik berat G ke ujung depan sepatu peluncur
- d Jarak titik gaya apung  $\gamma V'$  ke ujung depan sepatu peluncur
- f Jarak titik gaya apung  $\gamma V'$  ke AP
- g Jarak titik berat G ke AP
- h Jarak ujung belakang sepatu peluncur ke AP
- S Panjang sepatu peluncur
- $S'$  Panjang sepatu peluncur yang masih ada di atas landasan
- $S_o$  Panjang bagian badan kapal yang tercelup air dari AP
- $S_n$  Jarak yang ditempuh kapal dari AP sampai batas perairan
- $x'$  Jarak gaya reaksi landasan Q ke ujung landasan
- z Panjang sepatu peluncur di permukaan air

# **BAB I**

## **PENDAHULUAN**



## BAB I

### PENDAHULUAN

#### 1.1 LATAR BELAKANG

Industri bidang perkapalan dalam rangka memasuki era perdagangan bebas tentu akan semakin kompetitif. Pengadaan berbagai alternatif terhadap pemenuhan kebutuhan konsumen akan barang dan jasa dengan produk yang memiliki kualitas tinggi dengan harga yang relatif lebih murah. Penyelesaian pembangunan kapal yang tepat waktu serta efisiensi didalam proses pembangunan merupakan tuntutan pasar yang harus dilakukan oleh produsen dalam rangka pemuasan kebutuhan dari konsumennya.

Kapal sebagai sarana utama dalam menunjang kelancaran perdagangan baik nasional maupun internasional guna memindahkan barang dari suatu pulau ke pulau lain, dari suatu negara ke negara lain dan dari satu benua ke benua lain. Untuk angkutan barang seperti itu, sampai saat ini kapal masih merupakan alat transportasi yang handal karena belum terlihat sarana pengganti yang secara ekonomis lebih unggul (*Imron, 1995*).

Tugas Akhir ini pernah dibuat oleh Purwanto Budisusilo dengan judul Pembuatan Diagram Peluncuran End Launching Kapal Dibantu Komputer, yang menampilkan diagram peluncuran saja dan input data bonjean kapal dan perhitungan gaya angkat keatasnya masih manual, sehingga membutuhkan waktu yang relatif lama. Sedangkan pada Tugas Akhir ini, disamping menampilkan



## 1.2 TUJUAN

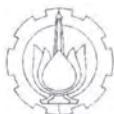
Setelah melihat latar belakang diatas, penulis bermaksud untuk membuat suatu program yang mampu secara otomatis :

1. Melakukan perhitungan peluncuran *end launching* dengan bantuan program aplikasi komputer.
2. Menampilkan diagram peluncuran *end launching* yang berhubungan dengan kondisi kritis kapal yang meliputi : P (berat kapal),  $\gamma V$  (gaya angkat ke atas),  $P * a$ ,  $\gamma V * b$ ,  $P * c$ ,  $\gamma V * d$ , Ta (sarat buritan), q (pembebatan rata-rata), qa (pembebatan belakang), qf (pembebatan depan), x (jarak gaya reaksi landasan terhadap ujung landasan).
3. Menampilkan grafik V (kecepatan kapal), a (percepatan kapal), dan S (jarak yang ditempuh) sebagai fungsi waktu.
4. Mengetahui kondisi kritis yang mungkin akan terjadi pada saat peluncuran sehingga dapat dilakukan cara penanggulangannya misalnya dengan menambahkan pelampung, dan penggereman.

## 1.3 MANFAAT

Manfaat atau kontribusi yang dapat diberikan oleh program aplikasi komputer yang dibuat ini adalah :

1. Proses perhitungan peluncuran *end launching* kapal dapat dipercepat dan lebih teliti, karena proses pembacaan grafik bonjean dilakukan dengan program komputer dengan interpolasi polinomial newton.



2. Mempermudah menganalisis kondisi kritis peluncuran dari berbagai macam variasi peluncuran sehingga dapat mempermudah dan mempercepat langkah antisipasi kondisi tersebut.

## 1.4 BATASAN MASALAH

Beberapa batasan yang dipergunakan dalam penyusunan program pada Tugas Akhir ini adalah :

1. Program peluncuran ini hanya dapat dilakukan untuk menghitung peluncuran kapal secara manajang (*end launching*).
2. Berat, LCB, dan kondisi sarat peluncuran kapal diambil dari data yang sudah ada seperti Tugas Rancangan maupun Tugas Akhir terdahulu.
3. Untuk berat dan LCB kapal yang tidak disebutkan dipakai pendekatan perhitungan dengan LR'64.
4. Landasan peluncuran dianggap rata atau lurus, tidak cembung atau tidak berchamber.
5. Data Bonjean kapal tersebut disimpan dalam bentuk database dengan MS Access 2002.
6. Pembacaan kurva bonjean dalam program dengan metode *Polinomial Interpolasi Newton*.
7. Berat dari peralatan peluncuran termasuk panjang sepatu peluncur merupakan fungsi dari berat dan panjang kapal jika tidak diketahui data sebenarnya.



8. Bahasa pemrograman yang dipakai dalam pembuatan program ini adalah Microsoft Visual Basic 6.0.
9. Output hasil programnya berupa grafik yang dapat dibuka dengan AutoCAD dengan file berekstensi \*.dwg.

## 1.5 METODOLOGI PENYELESAIAN

Metode yang digunakan dalam penyusunan Tugas Akhir ini meliputi :

- Studi Literatur

Studi literatur dilakukan dengan tujuan untuk merangkum teori-teori dasar serta acuan dalam penulisan Tugas Akhir ini. Teori yang dipelajari meliputi teori pencocokan kurva untuk mengetahui fungsi luasan bonjean terhadap garis air serta teori mengenai peluncuran kapal secara memanjang beserta aspek dinamisnya.

- Perumusan Latar Belakang

Pada tahap ini dilakukan perumusan latar belakang yang berkaitan dengan permasalahan yang ingin dipecahkan berdasarkan teori yang telah ada.

- Perumusan Masalah

Pada tahap ini dilakukan perumusan permasalahan berdasarkan langkah sebelumnya yang meliputi penentuan fungsi luasan bonjean terhadap garis air yang akan digunakan dalam perhitungan peluncuran kapal.



- Pemecahan Masalah

Pada tahap ini dilakukan pencarian solusi terhadap permasalahan-permasalahan yang telah dirumuskan sebelumnya dengan menggunakan program komputer untuk pemecahan masalah diatas.

Pada tahap ini dipilih metode yang akan digunakan dalam penentuan fungsi luasan bonjean terhadap garis air tiap station dan juga perhitungan peluncuran beserta aspek dinamisnya.

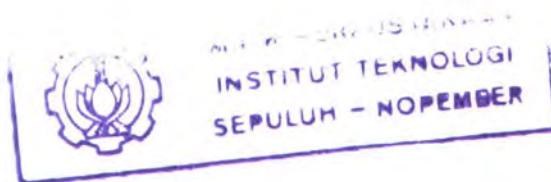
- Analisis dan Evaluasi

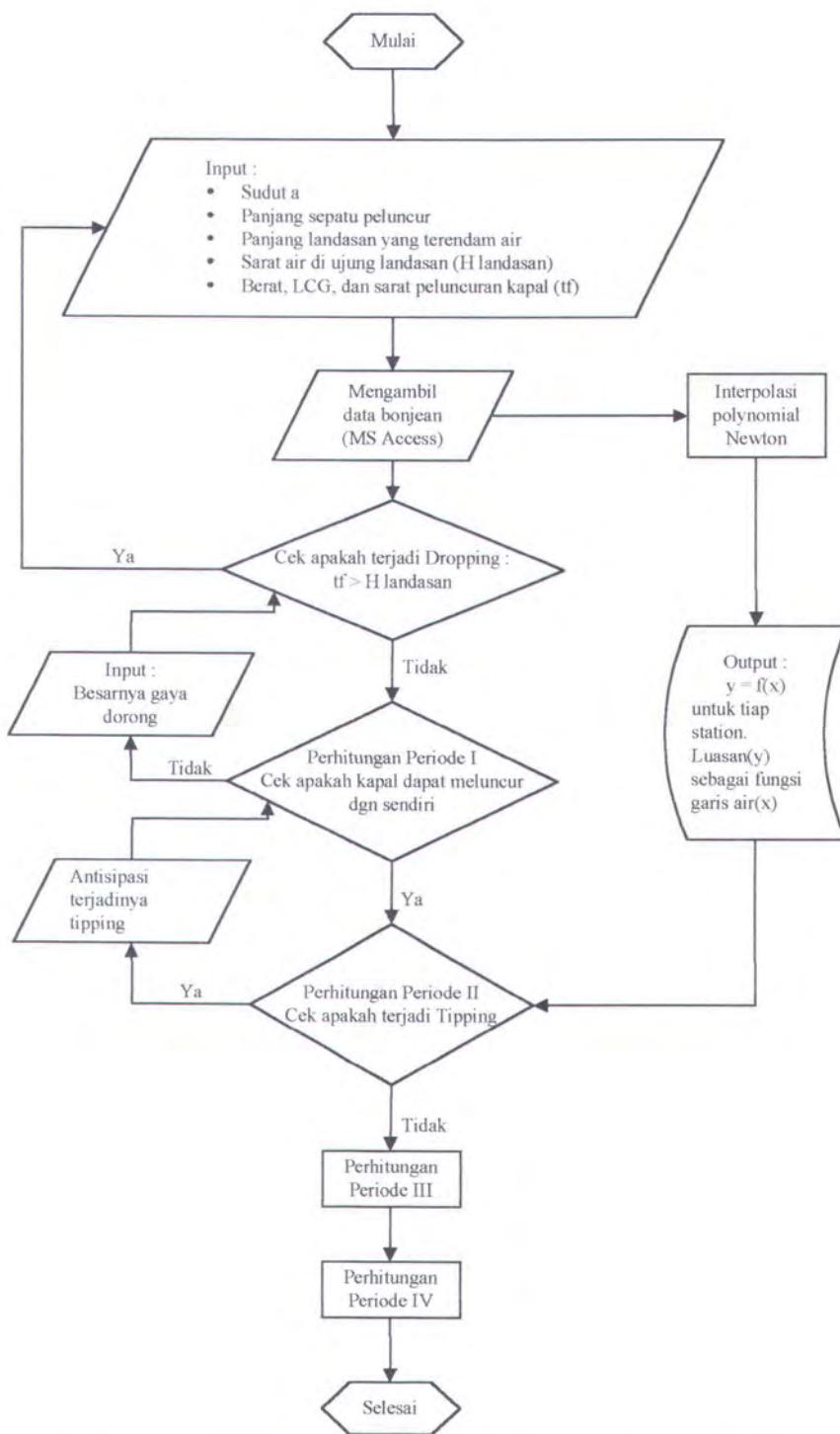
Hasil dari program komputer ini kemudian dianalisis dengan membandingkan hasil program komputer dengan hasil dari perhitungan secara manual sehingga diketahui keakuratan hasil program tersebut.

- Dokumentasi

Merupakan tahap terakhir dalam Tugas Akhir ini, yaitu tahap penulisan laporan Tugas Akhir terhadap proses penelitian yang telah dilakukan di atas.

Dalam melakukan pembuatan program perhitungan peluncuran ini dijelaskan dengan diagram alur seperti yang terlihat pada *Gambar 1.1*.





Gambar 1.1. Alur metodologi penyelesaian dan analisis masalah



## 1.6 SISTEMATIKA PENULISAN TUGAS AKHIR

### Bab I Pendahuluan

Bab ini menjelaskan latar belakang, tujuan, batasan masalah, metodologi penyelesaian dan sistematika penulisan dari Tugas Akhir ini.

### Bab II Dasar Teori

Bab ini menjelaskan dasar teori yang digunakan untuk mentransformasikan perhitungan yang dilakukan dengan cara manual menjadi perhitungan yang dilakukan dengan bantuan program komputer. Serta dijelaskan pula mengenai teori *Polinomial Interpolasi Newton* yang digunakan untuk pembacaan diagram bonjean kapal.

### Bab III Perhitungan Peluncuran

Bab ini berisi perhitungan yang dilakukan dengan program aplikasi komputer yang telah dibuat.

### Bab IV Teknik Pemrograman

Bab ini berisi teknik pemrograman yang dipakai untuk melakukan perhitungan peluncuran dan panduan penggunaan program peluncuran.

### Bab V Analisis Hasil Program

Bab ini berisi analisis hasil pemrograman komputer dibandingkan dengan perhitungan peluncuran secara manual.

### Bab VI Kesimpulan dan Saran

Bab ini berisi kesimpulan yang dapat diambil beserta saran pengembangan yang dapat dilakukan untuk menyempurnakan Tugas Akhir ini.

## **BAB II**

### **TEORI DASAR**



## BAB II

### DASAR TEORI

Dalam perhitungan peluncuran kapal, pembacaan grafik bonjean kapal merupakan salah satu hal yang perlu diperhatikan karena untuk menghitung besarnya gaya angkat yang diterima oleh kapal dapat dilakukan dengan mengintegralkan luasan tiap station kapal. Pada perhitungan secara manual, pembacaan grafik boonjean ini membutuhkan waktu yang lama sehingga diperlukan metode lain yaitu dengan pemrograman komputer menggunakan metode pencocokan kurva. Dari metode pencocokan kurva ini nantinya akan diperoleh luasan sebagai fungsi garis air untuk tiap station.

#### 2.1 PENCOCOKAN KURVA

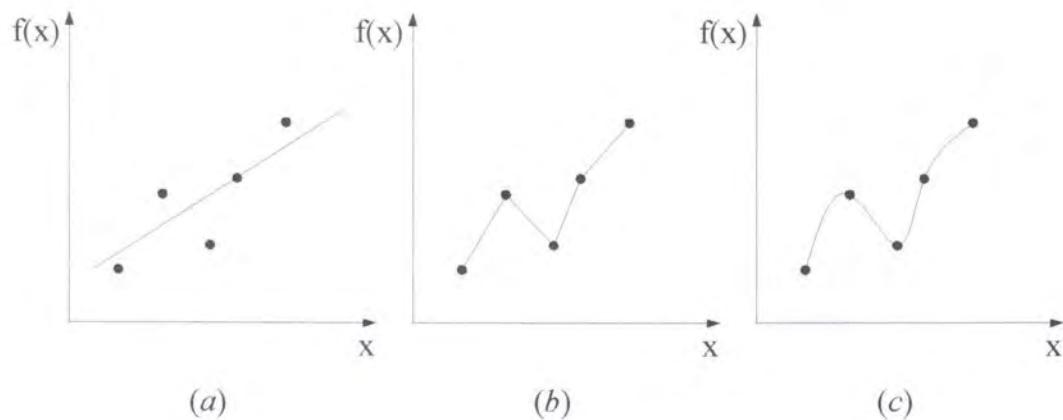
Data seringkali diberikan dalam harga-harga diskrit sepanjang suatu urutan kontinu. Tetapi dalam beberapa hal harus dicari taksiran pada titik-titik di antara harga-harga diskrit tersebut. Satu cara untuk melakukan hal ini adalah menghitung harga-harga fungsi pada sejumlah harga-harga diskrit sepanjang bentangan yang diinginkan. Kemudian sebuah fungsi yang sederhana bisa diturunkan untuk mencocokkan harga-harga ini (*Chapra dkk, 1985*).

Ada dua pendekatan umum untuk mencocokkan kurva yang dibedakan satu sama lain berdasarkan pada jumlah kesalahan sehubungan dengan data tersebut. Pertama, bilamana data menunjukkan suatu tingkat kesalahan yang



berarti (signifikan) atau *noise*, strateginya adalah menurunkan sebuah kurva tunggal yang menyatakan kecenderungan umum dari data. Sebuah pendekatan untuk keadaan ini dinamakan *regresi kuadrat terkecil (least-square regression)* (*Gambar 2.1a*).

Kedua, bilamana data telah diketahui dengan teliti, pendekatan dasar adalah mencocokkan sebuah kurva atau sederetan kurva yang melintasi setiap titik secara langsung. Contohnya adalah harga-harga luasan bonjean tiap garis air untuk masing-masing station pada kapal. Taksiran harga-harga di antara titik-titik diskrit yang telah dikenal dinamakan *interpolasi* (*Gambar 2.1b dan c*).



*Gambar 2.1.* Tiga usaha untuk mencocokkan sebuah kurva melalui 5 titik data. (a) regresi kuadrat terkecil, (b) interpolasi linear, dan (c) interpolasi kurvilinear.

## 2.2 REGRESI KUADRAT TERKECIL

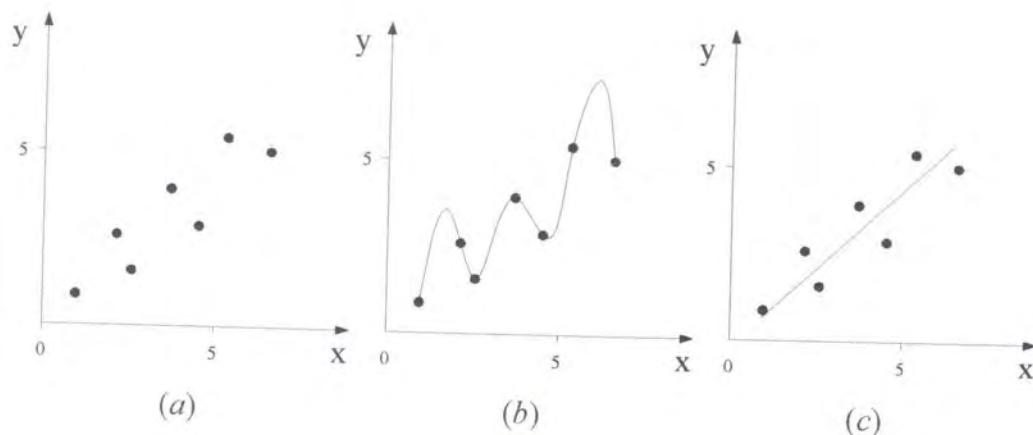
Dalam hal kesalahan penting yang tercampur dengan data, interpolasi polinomial tidak tepat dan mengandung hasil-hasil yang kurang memuaskan. Data



eksperimen seringkali termasuk jenis ini (*Chapra dkk, 1985*). Misalnya, *Gambar 2.2a* memperlihatkan tujuh titik data yang didapatkan secara eksperimen dan memperlihatkan variasi yang berarti. Pemeriksaan visual terhadap data mengusulkan suatu hubungan yang positif antara  $x$  dan  $y$ . Akhirnya, kecenderungan secara menyeluruh menunjukkan bahwa harga  $y$  yang lebih besar bersesuaian dengan harga  $x$  yang lebih besar.

Kalau sebuah polinomial interpolasi orde keenam dicocokkan terhadap data ini (*Gambar 2.2b*), ia akan tepat melewati semua titik. Tetapi karena variabilitas yang terdapat pada data, kurva itu berosilasi secara melebar dalam interval di antara titik-titik. Terutama harga-harga yang diinterpolasikan pada  $x = 1,5$  dan  $x = 6,5$  kelihatan menjadi baik di luar bentangan yang diharapkan oleh data.

Suatu strategi yang lebih pantas untuk kasus demikian ialah dengan menurunkan suatu fungsi aproksimasi yang layak atau cocok terhadap bentuk atau kecenderungan umum dari data tanpa perlu disesuaikan dengan masing-masing titik. *Gambar 2.2c* memperlihatkan bagaimana sebuah garis lurus dapat digunakan secara umum dalam mengkarakterisasi kecenderungan data tanpa melewati sembarang titik tertentu.



Gambar 2.2. (a) Data yang memperlihatkan kesalahan yang signifikan  
(b) Polinomial mencocokkan osilasi di luar bentangan data  
(c) Hasil yang lebih memuaskan dengan menggunakan pencocokan kuadrat terkecil

Suatu cara untuk menentukan garis dalam *Gambar 2.2c* adalah dengan memeriksa data yang akan diplotkan secara visual, kemudian membuat sketsa suatu garis “terbaik” melalui titik-titik tersebut. Walupun pendekatan ini masuk akal, ia tidaklah sempurna, karena selalu berubah-ubah. Artinya, kalau titik-titik tidak mendefinisikan sebuah garis lurus yang sempurna (dalam kasus dimana interpolasi adalah tepat), analisis yang berlainan akan menarik garis yang berlainan pula.

Untuk menghindari subyektivitas ini, beberapa kriteria harus memikirkan kelayakan pencocokan. Suatu cara yang dilakukan ialah dengan menurunkan sebuah kurva yang meminimalkan ketidakcocokan di antara titik-titik data dan



kurva tersebut. Teknik untuk melaksanakan tujuan ini dinamakan *regresi kuadrat terkecil (least-square regression)*.

### 2.2.1 REGRESI LINEAR

Contoh yang paling sederhana dari suatu aproksimasi kuadrat terkecil adalah mencocokkan sebuah garis lurus terhadap sekumpulan pasangan pengamatan :  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ . Pernyataan matematika untuk garis lurus itu adalah :

$$y = a_0 + a_1 x + E \quad [2.1]$$

di mana  $a_0$  dan  $a_1$  adalah koefisien-koefisien yang masing-masing menyatakan perpotongan dan kemiringan serta  $E$ , yaitu kesalahan atau selisih (residual) antara model dan pengamatan, yang dapat dinyatakan dengan mengatur kembali Persamaan 2.1 sebagai :

$$E = y - a_0 - a_1 x$$

Jadi, kesalahan atau residual adalah perbedaan antara harga  $y$  sebenarnya dan harga aproksimasi  $a_0 + a_1 x$  yang diprediksikan oleh persamaan linear itu.

Satu strategi untuk mencocokkan sebuah garis terbaik melalui data adalah dengan meminimalkan jumlah kesalahan residual, seperti dalam :

$$\sum_{i=1}^n E_i = \sum_{i=1}^n (y_i - a_0 - a_1 x_i) \quad [2.2]$$

Untuk mengatasi kelemahan pendekatan di atas ialah meminimalkan jumlah kuadrat residual,  $S_r$  yakni pada :

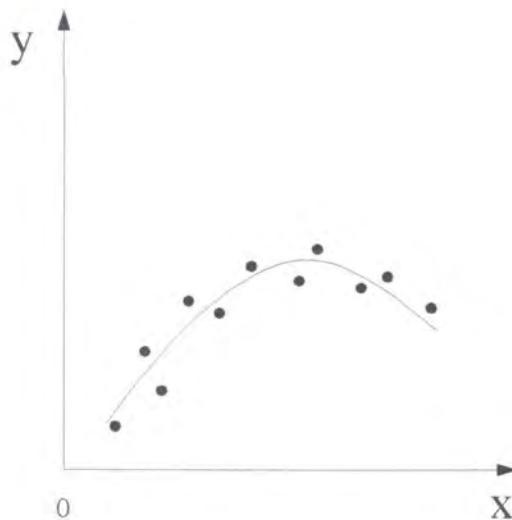
$$S_r = \sum_{i=1}^n E_i^2 = \sum_{i=1}^n (y_i - a_0 - a_1 x_i)^2 \quad [2.3]$$



Kriteria ini mempunyai sejumlah keuntungan, termasuk kenyataan bahwa ia mengandung sebuah garis yang unik bagi sekumpulan data yang diberikan.

### 2.2.2. REGRESI POLINOMIAL

Dalam pembahasan di atas, sebuah prosedur telah dikembangkan untuk menurunkan persamaan dari sebuah garis lurus dengan menggunakan criteria kuadrat terkecil. Beberapa data teknik, walaupun menunjukkan sebuah pola ditandai seperti terlihat dalam *Gambar 2.3*, kurang baik bila dinyatakan oleh sebuah garis lurus. Untuk kasus-kasus ini, kurva harus disesuaikan lebih baik agar bisa mencocokkan data. Metode yang dipakai adalah mencocokkan polinomial terhadap data dengan memakai *regresi polinomial*.



*Gambar 2.3.* Indikasi bahwa parabola lebih cocok

Prosedur kuadrat terkecil dapat diperluas untuk mencocokkan data terhadap polinomial berderajat ke- $m$ :



$$y = a_0 + a_1x + a_2x^2 + \cdots + a_mx^m \quad [2.4]$$

Untuk hal ini, jumlah kuadrat residual adalah :

$$S_r = \sum_{i=1}^n (y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_mx_i^m)^2 \quad [2.5]$$

Turunan terhadap setiap koefisien polinomial adalah :

$$\frac{\partial S_r}{\partial a_0} = -2 \sum (y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_mx_i^m)$$

$$\frac{\partial S_r}{\partial a_1} = -2 \sum x_i (y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_mx_i^m)$$

$$\frac{\partial S_r}{\partial a_2} = -2 \sum x_i^2 (y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_mx_i^m)$$

$$\frac{\partial S_r}{\partial a_m} = -2 \sum x_i^m (y_i - a_0 - a_1x_i - a_2x_i^2 - \cdots - a_mx_i^m)$$

Persamaan-persamaan ini dapat disamakan dengan nol, dan diatur kembali menjadi kumpulan persamaan normal berikut :

$$a_0n + a_1 \sum x_i + a_2 \sum x_i^2 + \cdots + a_m \sum x_i^m = \sum y_i$$

$$a_0 \sum x_i + a_1 \sum x_i^2 + a_2 \sum x_i^3 + \cdots + a_m \sum x_i^{m+1} = \sum x_i y_i$$

$$a_0 \sum x_i^2 + a_1 \sum x_i^3 + a_2 \sum x_i^4 + \cdots + a_m \sum x_i^{m+2} = \sum x_i^2 y_i$$

$$a_0 \sum x_i^m + a_1 \sum x_i^{m+1} + a_2 \sum x_i^{m+2} + \cdots + a_m \sum x_i^{2m} = \sum x_i^m y_i \quad [2.6]$$

## 2.3 INTERPOLASI

Data seringkali diberikan dalam harga-harga diskrit sepanjang suatu urutan kontinu. Tetapi dalam beberapa hal harus mencari taksiran pada titik-titik di

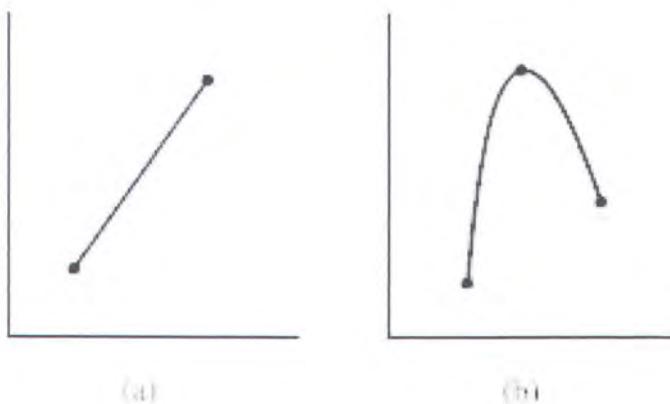


antara harga-harga diskrit tersebut. Misalnya titik data pada waterline yang tidak diketahui luasannya pada grafik bonjean. Metode yang paling sering digunakan untuk maksud ini adalah interpolasi polinomial (*Chapra dkk, 1985*).

Secara umum formula untuk sebuah polinomial orde ke- $n$  adalah :

$$f(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n \quad [2.7]$$

Untuk  $n + 1$  titik-titik data, terdapat satu dan hanya satu polinomial orde ke- $n$  atau kurang yang melewati semua titik. Misalnya, hanya terdapat satu garis lurus (yakni polinomial orde pertama) yang menghubungkan kedua titik (*Gambar 2.4a*). Dengan cara yang sama, hanya satu parabola yang menghubungkan kumpulan dari tiga buah titik (*Gambar 2.4b*). *Interpolasi polinomial* ini kemudian memberikan sebuah formula untuk menghitung nilai-nilai antara.



*Gambar 2.4. Polinomial orde pertama (a), orde kedua (b)*

Walaupun terdapat satu dan hanya satu polinomial orde ke- $n$  yang mencocokkan  $n + 1$  titik, ada berbagai format matematika dimana polinomial ini dapat dinyatakan. Dalam hal ini ada dua alternatif yang cocok untuk dilaksanakan



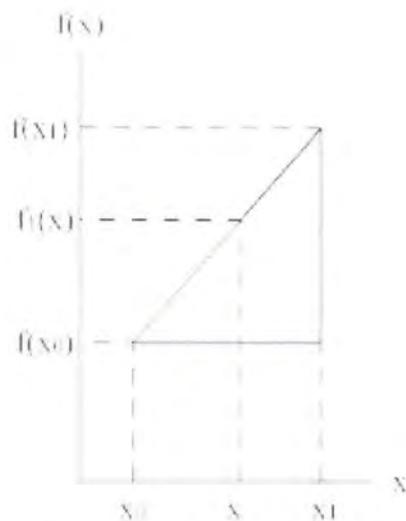
pada komputer pribadi. Kedua alternatif itu adalah *polinomial Newton* dan *polinomial Lagrange*.

### 2.2.1 POLINOMIAL INTERPOLASI DIFERENSI TERBAGI NEWTON

Seperti telah diuraikan di atas, terdapat berbagai bentuk alternatif untuk menyatakan sebuah polinomial interpolasi. *Polinomial interpolasi diferensi terbagi Newton* adalah di antara bentuk yang paling populer dan berguna. Sebelum membahas persamaan umum, sebelumnya dibahas versi orde pertama dan kedua, karena interpretasi visualnya mudah.

#### a) *Interpolasi Linear*

Bentuk yang termudah dari interpolasi ini adalah menghubungkan dua titik data dengan sebuah garis lurus. Teknik ini disebut interpolasi linear, dan dilukiskan secara grafis pada *Gambar 2.5*.



Gambar 2.5. Interpolasi linear

Dengan menggunakan segitiga sebangun :

$$\frac{f_1(x) - f(x_0)}{x - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

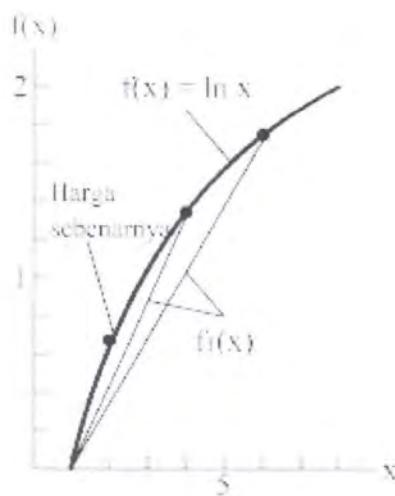


yang dapat diatur kembali supaya memenuhi :

$$f_1(x) = f(x_0) + \frac{f(x_1) - f(x_0)}{x_1 - x_0}(x - x_0) \quad [2.8]$$

adalah sebuah *formula interpolasi linear*. Notasi  $f_1(x)$  menandakan bahwa ia adalah sebuah polinomial interpolasi orde pertama. Perhatikan bahwa disamping menyatakan *slope* dari garis yang menghubungkan titik-titik, suku  $[f(x_1) - f(x_0)] / (x_1 - x_0)$  adalah sebuah aproksimasi diferensi terbagi hingga dari turunan pertama.

Umumnya, semakin kecil interval di antara titik-titik data, aproksimasinya semakin baik. Karakteristik ini ditunjukkan pada *Gambar 2.6* dimana semakin kecil interval di antara titik-titik data, aproksimasinya semakin baik.



*Gambar 2.6.* Perbandingan dua interpolasi linear

### b) Interpolasi Kuadratik

Kesalahan pada interpolasi linear di atas disebabkan oleh kenyataan bahwa dalam penyelesaian sebuah kurva didekati dengan sebuah garis lurus. Konsekuensinya, sebuah strategi untuk memperbaiki taksiran adalah dengan memperkenalkan beberapa lengkungan ke dalam garis yang menghubungkan titik-



titik. Jika tiga titik data telah tersedia, ini dapat dilakukan dengan sebuah polinomial orde kedua (disebut juga sebuah polinomial kuadratik atau sebuah parabola). Suatu bentuk yang terutama sangat baik untuk tujuan ini adalah :

$$f_2(x) = b_0 + b_1(x - x_0) + b_2(x - x_0)(x - x_1) \quad [2.9]$$

Walaupun Persamaan [2.9] kelihatannya berbeda dari polinomial umum (Persamaan [2.7]), kedua persamaan tersebut adalah ekuivalen. Ini dapat dilihat dengan mengalikan suku-suku dalam Persamaan [2.9], sehingga :

$$f_2(x) = b_0 + b_1x - b_1x_0 + b_2x^2 + b_2x_0x_1 - b_2xx_0 - b_2xx_1$$

atau dengan mengumpulkan suku-suku :

$$f_2(x) = a_0 + a_1x + a_2x^2$$

dimana :

$$a_0 = b_0 - b_1x_0 + b_2x_0x_1$$

$$a_1 = b_1 - b_2x_0 - b_2x_1$$

$$a_2 = b_2$$

Jadi, Persamaan [2.7] dan [2.9] adalah alternatif formulasi ekuivalen lainnya dari polinomial orde kedua yang unik menghubungkan tiga titik.

Suatu prosedur sederhana dapat dipakai untuk menentukan harga-harga koefisien. Untuk  $b_0$ , Persamaan [2.9] dengan  $x = x_0$  dapat digunakan untuk menghitung :

$$b_0 = f(x_0) \quad [2.10]$$

Persamaan [2.10] dapat disubstitusikan ke dalam Persamaan [2.9] dan dapat dievaluasikan pada  $x = x_1$  untuk :



$$b_1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad [2.11]$$

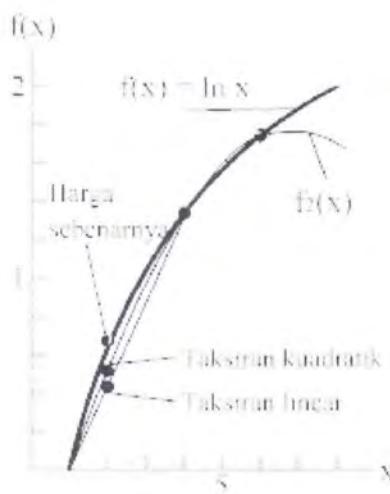
Akhirnya, Persamaan [2.10] dan [2.11] dapat disubstitusikan ke dalam Persamaan [2.9] dan dapat dievaluasikan pada  $x = x_2$  serta diselesaikan (setelah beberapa manipulasi aljabar) untuk :

$$b_2 = \frac{\frac{f(x_2) - f(x_1)}{x_2 - x_1} - \frac{f(x_1) - f(x_0)}{x_1 - x_0}}{x_2 - x_0} \quad [2.12]$$

Hal yang perlu diperhatikan, jika kasusnya dengan interpolasi linear,  $b_1$  masih menyatakan slope garis yang menghubungkan titik  $x_0$  dan  $x_1$ . Jadi, dua suku pertama dari Persamaan [2.9] adalah ekuivalen terhadap interpolasi linear dari  $x_0$  ke  $x_1$ , seperti dispesifikasikan sebelumnya dalam Persamaan [2.8]. Suku terakhir,  $b_2(x - x_0)(x - x_1)$ , memberikan lengkungan orde kedua ke dalam formula.

Sebelum mengilustrasikan bagaimana menggunakan Persamaan [2.9], terlebih dahulu diperiksa bentuk koefisien  $b_2$ . Ia sangat mirip dengan aproksimasi terbagi hingga dari turunan kedua yang telah diperkenalkan. Jadi, Persamaan [2.9] adalah titik tolak untuk memanifestasikan suatu struktur yang sangat mirip dengan perluasan Deret Taylor.

Jadi, kelengkungan yang dihasilkan oleh formula kuadratik (*Gambar 2.7*) memperbaiki interpolasi, dibandingkan dengan hasil yang diperoleh dengan menggunakan garis lurus pada *Gambar 2.6*.



Gambar 2.7. Interpolasi kuadratik

### 2.3.2 Bentuk Umum Polinomial Interpolasi Newton

Analisis sebelumnya dapat dibuat untuk mencocokkan polinomial orde ke- $n$  untuk  $n + 1$  titik data. Polinomial orde ke- $n$  adalah :

$$f_n(x) = b_0 + b_1(x - x_0) + \cdots + b_n(x - x_0)(x - x_1)\cdots(x - x_{n-1}) \quad [2.13]$$

Seperti telah dilakukan sebelumnya dengan interpolasi linear dan kuadratik, titik-titik data yang digunakan untuk mengevaluasikan koefisien  $b_0, b_1, \dots, b_n$ . Untuk suatu polinomial orde ke- $n$ , diperlukan  $n + 1$  titik data :  $x_0, x_1, x_2, \dots, x_n$ . Dengan menggunakan titik-titik data ini, persamaan berikut dipakai untuk mengevaluasikan koefisien-koefisien :

$$b_0 = f(x_0) \quad [2.14]$$

$$b_1 = f[x_1, x_0] \quad [2.15]$$

$$b_2 = f[x_2, x_1, x_0] \quad [2.16]$$

.

.

$$b_n = f[x_n, x_{n-1}, \dots, x_1, x_0] \quad [2.17]$$



dimana evaluasi dari fungsi yang berakolade adalah diferensi terbagi hingga.

Misalnya, *diferensi terbagi hingga pertama* dinyatakan sebagai :

$$f[x_i, x_j] = \frac{f(x_i) - f(x_j)}{x_i - x_j} \quad [2.18]$$

*Diferensi terbagi hingga kedua*, yang menyatakan perbedaan dari dua diferensi terbagi hingga pertama umumnya dinyatakan sebagai :

$$f[x_i, x_j, x_k] = \frac{f[x_i, x_j] - f[x_j, x_k]}{x_i - x_k} \quad [2.19]$$

Dengan cara yang serupa, *diferensi terbagi hingga ke-n* adalah :

$$f[x_n, x_{n-1}, \dots, x_1, x_0] = \frac{f[x_n, x_{n-1}, \dots, x_1] - f[x_{n-1}, x_{n-2}, \dots, x_0]}{x_n - x_0} \quad [2.20]$$

Diferensi-diferensi ini dapat dipakai untuk mengevaluasikan koefisien-koefisien dalam Persamaan [2.14] sampai [2.17] yang kemudian dimasukkan ke dalam Persamaan [2.13] untuk memenuhi interpolasi polinomial :

$$\begin{aligned} f_n(x) = & f(x_0) + (x - x_0)f[x_1, x_0] + (x - x_0)(x - x_1)f[x_2, x_1, x_0] \\ & + \dots + (x - x_0)(x - x_1) \dots (x - x_{n-1})f[x_n, x_{n-1}, \dots, x_0] \end{aligned} \quad [2.21]$$

## 2.4 PEMILIHAN METODE PENCOCOKAN KURVA

Metode pencocokan kurva yang dipakai pada Tugas Akhir ini adalah Metode Interpolasi Polinomial Newton, karena hasil fungsi persamaannya melalui tepat pada titik data yang diberikan sehingga koreksi kesalahannya relatif lebih kecil daripada metode regresi. Disamping itu fungsi luasan bonjean adalah *kurvilinear* (berbentuk kurva yang halus / *smooth*) sehingga lebih cocok dipakai metode ini.



## 2.5 PELUNCURAN

Peluncuran (*launching*) adalah menurunkan kapal dari landasan peluncur menggunakan gaya berat kapal atau dengan memberikan gaya dorong tambahan yang bekerja pada bidang miring kapal. Perhitungan peluncuran ini dipergunakan untuk menghindari kapal dari bahaya-bahaya yang tidak dikehendaki, seperti kapal tenggelam ketika diluncurkan, *dropping*, *tipping* dan *lifting* (Bakri dkk, 1983).

Proses perhitungan dan pembuatan diagram peluncuran yang memakan waktu cukup lama, sehingga timbul keinginan untuk mempermudah dan mempercepat perhitungan dan pembuatan diagram peluncuran ini. Dengan membuat sebuah perangkat lunak (program aplikasi komputer) yang dapat melakukan perhitungan dan pembuatan diagram peluncuran kapal, maka diperoleh efisiensi waktu serta mempermudah proses perhitungan dan pembuatan diagram peluncuran kapal.

Peluncuran kapal pada dasarnya dibedakan menjadi dua macam, yaitu :

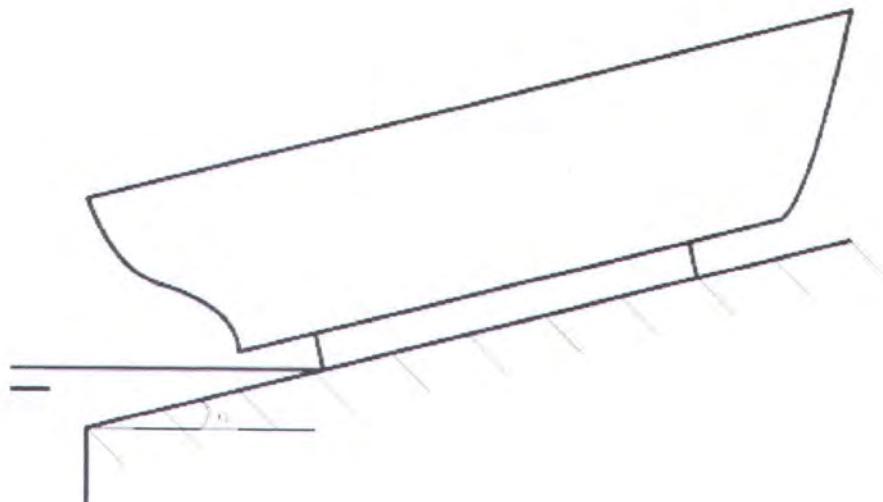
1. Peluncuran memanjang (*end launching*)
2. Peluncuran melintang (*side launching*)

Di dalam peluncuran kapal, pada umumnya digunakan peluncuran memanjang (*end launching*), sehingga di dalam Tugas Akhir ini dipilih perhitungan peluncuran memanjang (*end launching*) untuk mempercepat proses perhitungannya. Sedangkan cara peluncuran melintang (*side launching*) digunakan hanya apabila keadaannya memaksa, misalnya bila permukaan air (*water front*) di depan landasannya sempit. Hal ini biasanya pada perairan sungai.



Pada peluncuran memanjang, buritan kapal diarahkan ke air sehingga buritan akan terkena air terlebih dahulu (*Gambar 2.8*). Hal ini dilakukan dengan tujuan supaya :

1. Linggi buritan tidak terbentur pada landasan
2. Pada waktu kapal masuk ke air, maka dapat mengurangi laju kecepatan meluncurnya kapal
3. Menambah gaya angkat ke atas pada waktu kapal diluncurkan



*Gambar 2.8. Peluncuran end launching dengan sudut kemiringan  $\alpha$*

Di dalam proses peluncurannya untuk mengurangi terjadinya gesekan antara sepatu peluncur dengan landasan diberikan bahan pelumas yang terdiri dari campuran kapur, gemuk, dan parafin. Besarnya tahanan yang disebabkan oleh gesekan ini tergantung dari :

1. Macam bahan pelumas
2. Tekanan rata-rata dari sepatu peluncur terhadap landasan



3. Suhu udara pada waktu peluncuran dilaksanakan
4. Kecepatan peluncuran

Sebagai rata-rata dari koefisien gesekan,  $f_d$  selama peluncuran tersebut adalah :

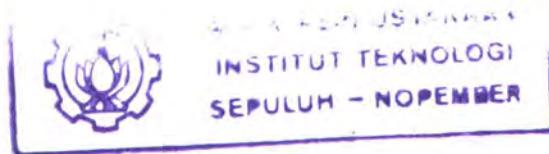
- $f_d = 0.024$  berlaku untuk tekanan rata-rata pada landasan sebesar  $30 \text{ Ton/m}^2$
- $f_d = 0.032$  berlaku untuk tekanan rata-rata pada landasan sebesar  $20 \text{ Ton/m}^2$
- $f_d = 0.040$  berlaku untuk tekanan rata-rata pada landasan sebesar  $10 \text{ Ton/m}^2$

Koefisien gesekan pada waktu kapal berhenti  $f_s$ , untuk semua tekanan rata-rata antara 0.040 s/d 0.080.

Sudut kemiringan peluncuran,  $\alpha$  dalam hal ini dianggap sama dengan sudut kemiringan landasan terhadap permukaan air. Sedangkan perincian diperkirakan kurang lebih sebagai berikut :

- Untuk kapal-kapal yang besar       $\operatorname{tg} \alpha = \frac{1}{22} \sim \frac{1}{26}$
- Untuk kapal-kapal yang sedang       $\operatorname{tg} \alpha = \frac{1}{18} \sim \frac{1}{20}$
- Untuk kapal-kapal yang kecil       $\operatorname{tg} \alpha = \frac{1}{12} \sim \frac{1}{14}$

Harga-harga ini bukan merupakan harga yang pasti sebab masih tergantung dari faktor-faktor lain, misalnya bentuk badan kapal dan sebagainya.





Untuk perhitungan peluncuran memanjang dibagi atas tiga bagian :

- Periode I : Dimulai pada waktu kapal dilepaskan dan berakhir pada waktu kapal menyinggung permukaan air.
- Periode II : Dimulai pada akhir Periode I dan berakhir pada waktu bagian belakang (*stern*) kapal mulai terangkat (*stern lift*).
- Periode III : Dimulai pada akhir Periode II dan berakhir pada saat kapal mulai terapung seluruhnya (*free floating*).
- Periode IV : Dimulai pada akhir Periode III dan berakhir pada saat kapal berhenti.

Langkah pertama pada perhitungan peluncuran adalah perhitungan berat dan titik berat terhadap penampang tengah (*midship*) kapal pada saat kapal diluncurkan. Karena kapal pada waktu diluncurkan biasanya dalam keadaan belum selesai pembangunannya, maka berat kapal kurang dari berat sesungguhnya. Hal ini dapat dilakukan dengan rumus pendekatan yaitu :

$$P = C(Pb + Pa) \quad [2.22]$$

dimana : P = berat kapal pada saat diluncurkan

P<sub>b</sub> = berat badan kapal

P<sub>a</sub> = berat mesin kapal

C = koefisien yang harganya : 0.85 untuk kapal barang

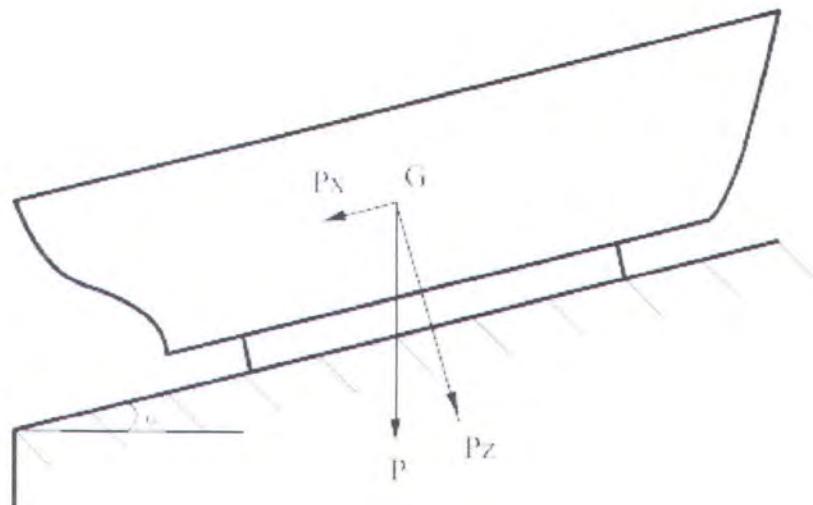
0.65 untuk kapal penumpang

Disamping rumus [2.22] di atas, dapat dipakai juga pendekatan perhitungan berat dan titik berat kapal dengan *Lloyd Register 1964*.



### 2.5.1 PERIODE I

Periode ini dimulai pada waktu kapal dilepaskan dan berakhir pada waktu kapal menyentuh permukaan air. Faktor yang memberikan pengaruh penting pada perhitungan peluncuran adalah sudut kemiringan landasan,  $\alpha$  dan tinggi pasang surut air. Secara umum dapat dikatakan bila sudut kemiringan landasan diperkecil, kemungkinan terjadinya *tipping* diperbesar, tetapi kemungkinan tekanan pada landasan pada saat *stern lift* diperkecil.



Gambar 2.9. Periode I

Dari Gambar 2.9, P menunjukkan berat dari kapal yang akan diluncurkan termasuk berat peralatan peluncuran, sudut  $\alpha$  adalah sudut kemiringan landasan. Komponen berat P dapat diuraikan menjadi gaya Px dan Pz.

$$Px = P \sin \alpha \text{ yang sejajar dengan landasan}$$

$$Px = P \sin \alpha \text{ yang tegak lurus landasan}$$

$$G = \text{Titik berat kapal beserta peralatan peluncurannya}$$



Syarat agar kapal dapat bergerak dengan sendirinya adalah memenuhi beberapa kondisi yaitu :

$$1. \quad P_x > W$$

Dimana  $W =$  tahanan gesek

$$2. \quad P \sin \alpha > P \cos \alpha \times f_s \quad [2.23]$$

Jika  $\alpha$  kecil, maka  $\sin \alpha \approx a$  dan  $\cos \alpha \approx 1$

Maka  $\sin \alpha > \cos \alpha \times f_s$

$$\alpha > f_s \text{ atau } \tan \alpha > f_s \quad [2.24]$$

Dengan perkataan lain, supaya kapal dapat bergerak dengan sendirinya maka sudut kemiringan landasan harus lebih besar dari  $f_s$ . Jika syarat ini tidak dipenuhi, maka untuk menggerakkan kapal perlu adanya gaya dorong sebesar :

$$P_d = P (f_s - \alpha) \quad [2.25]$$

Persamaan gerakan kapal pada landasan pada Periode I oleh pengaruh gaya beratnya sendiri adalah (*Semyonov, 1957*) :

$$F = m \cdot a$$

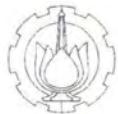
$$P \sin \alpha - W = \frac{P}{G} \cdot \frac{dv}{dt}$$

$$P \sin \alpha - f_d \cdot P \cos \alpha = \frac{P}{G} \cdot \frac{dv}{dt}$$

$$a = g(\sin \alpha - f_d \cdot \cos \alpha) \quad [2.26]$$

$$\frac{dv}{dt} = g(\sin \alpha - f_d \cdot \cos \alpha) \quad [2.27]$$

bila  $\alpha$  kecil :



$$\frac{dv}{dt} = g(\alpha - fd)$$

Kecepatan :

$$Vx = g(\alpha - fd)t + V_0 \quad [2.28]$$

Jarak buritan :

$$Sx = 0.5g(\alpha - fd)t^2 + V_0 \cdot t + S_0 \quad [2.29]$$

Bila  $V_0 = 0$  dan  $S_0 = 0$

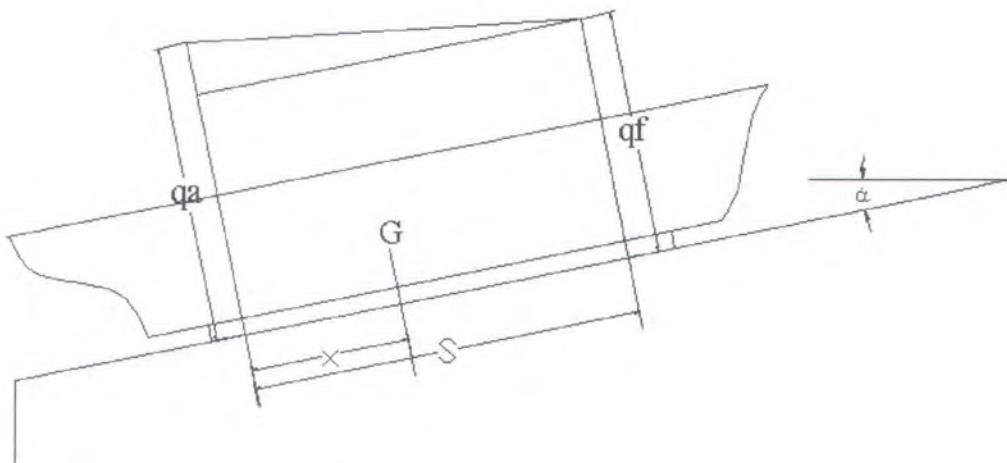
Maka :

$$\begin{aligned} Vx &= g(\alpha - fd) \cdot t \\ Sx &= 0.5g(\alpha - fd) \cdot t^2 \end{aligned} \quad [2.30]$$

Pembebanan rata-rata yang bekerja pada landasan untuk tiap meter diambil :

$$q = \frac{P}{S} \text{ [ton/m]} \quad [2.31]$$

Pembebanan pada landasan ini dapat digambarkan sebagai sebuah trapesium seperti yang ditunjukkan *Gambar 2.10* berikut ini.



*Gambar 2.10.* Pembebanan rata-rata Periode I



Keterangan *Gambar 2.10* :

$q_f$  = pembebanan depan atau haluan

$q_a$  = pembebanan belakang atau buritan

$x$  = jarak titik berat bidang beban terhadap ujung belakang bidang beban

Berat peluncuran       $P$  = luas bidang beban

$$P = 0.5 S (q_f + q_a)$$

Karena  $q = P/S$

maka,  $P = q \cdot S$

$$q \cdot S = 0.5 S (q_f + q_a)$$

$$2 q = q_f + q_a$$

$$q_f = 2q - q_a$$

$$q_a = 2q - q_f$$

[2.32]

Dengan membagi bidang beban itu menjadi sebuah segitiga siku-siku dan sebuah persegi panjang, maka :

$$P = q_f \cdot S + (q_a - q_f) 0.5 S$$

Dan momen terhadap ujung belakang adalah :

$$P \cdot x = q_f \cdot S \cdot 0.5 \cdot S + (q_a - q_f) \cdot 0.5 \cdot S \cdot 1/6 \cdot S$$

$$q \cdot S \cdot x = 2 \cdot q_f \cdot S^2 + 1/6 \cdot S^2 (q_a - q_f)$$

$$= 1/6 \cdot S^2 (q_a + 2 \cdot q_f)$$

$$q \cdot x = 1/6 \cdot S (q_a + 2 \cdot q_f)$$

$$q_a \cdot S = 6(q \cdot x) - 2 \cdot q_f \cdot S$$

$$2 \cdot q_f \cdot S = 6(q \cdot x) - q_a \cdot S$$

[2.33]

Dari Pers. [2.26] dan [2.27] didapat :



$$\begin{aligned} qa &= \frac{2q(2S - 3x)}{S} \\ qf &= \frac{2q(3x - S)}{S} \end{aligned} \quad [2.34]$$

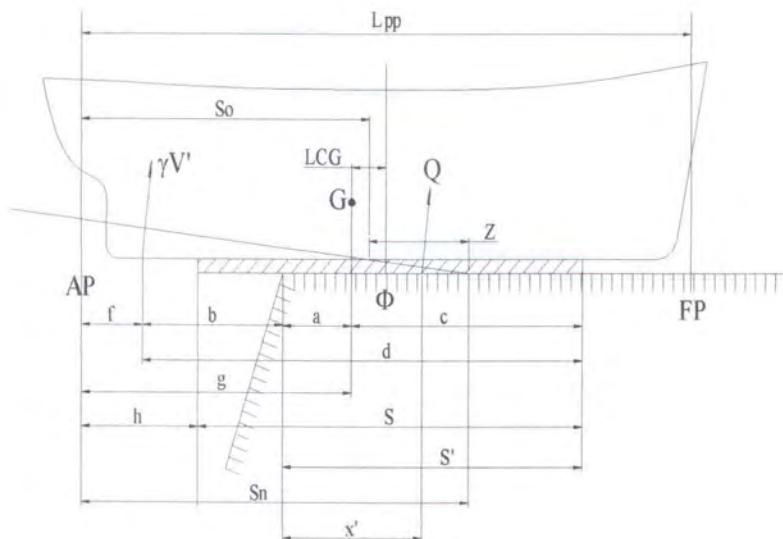
Sehingga jika :

$$x = 0.5 S \text{ maka } q = qa = qf$$

$$x < 0.5 S \text{ maka } qf < qa$$

## 2.5.2 PERIODE II

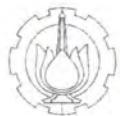
Periode ini dimulai pada akhir Periode I dan berakhir pada waktu bagian belakang (*stern*) kapal mulai terangkat (*stern lift*) seperti pada gambar 2.11.



Gambar 2.11. Periode II

Keterangan Gambar 2.11 :

- a = Jarak titik berat G ke ujung landasan
- b = Jarak titik gaya apung  $\gamma V'$  ke ujung landasan



c = Jarak titik berat G ke ujung depan sepatu peluncur

d = Jarak titik gaya apung  $\gamma V'$  ke ujung depan sepatu peluncur

f = Jarak titik gaya apung  $\gamma V'$  ke AP

g = Jarak titik berat G ke AP

h = Jarak ujung belakang sepatu peluncur ke AP

S = Panjang sepatu peluncur

S' = Panjang sepatu peluncur yang masih ada di atas landasan

So = Panjang bagian badan kapal yang tercelup air dari AP

Sn = Jarak yang ditempuh kapal dari AP sampai batas perairan

x' = Jarak gaya reaksi landasan Q ke ujung landasan

z = Panjang sepatu peluncur di permukaan air

P = Berat total peluncuran

Q = Gaya reaksi pada landasan

Pada periode ini hal yang diperhatikan adalah :

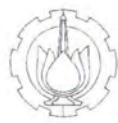
3. Kondisi *stern lift* (bagian buritan kapal mulai terangkat)

4. Momen anti *tipping*

Kondisi stern lift terjadi jika momen yang ditimbulkan oleh berat kapal terhadap ujung belakang sepatu peluncur dan momen yang ditimbulkan oleh gaya angkat ke atas terhadap ujung belakang sepatu peluncur sama.

$$P \cdot c = \gamma V \cdot d \quad [2.35]$$

Sedangkan *tipping* terjadi bila momen yang ditimbulkan oleh berat kapal terhadap ujung landasan lebih besar daripada momen yang ditimbulkan oleh gaya angkat ke atas terhadap ujung landasan. Sehingga momen anti *tipping* (MAT)-nya adalah :



$$\text{MAT} = \gamma V \cdot b - P \cdot a \quad [2.36]$$

Aspek dinamis ditinjau dari persamaan gerak pada periode ini adalah sebagai berikut (Semyonov, 1957) :

$$\Sigma F = m \cdot a$$

$$(P - \gamma V) \cdot (\alpha - \mu d) - R = \frac{P}{g} \cdot a, \text{ dimana } R = \text{tahanan air}$$

$$a = \frac{dv}{dt} = \frac{g}{P} (P - \gamma V) \cdot (\alpha - \mu d) - R \quad [2.37]$$

dimana menurut I.G. Bubnova :

$$R = 6,5(W + 9,25W^1)V_x^2 \quad [2.38]$$

dimana :

$W$  : luas penampang melintang terbesar bagian kapal yang masuk dalam air

$W^1$  : luas penampang melintang terbesar alat peluncur

$V_x$  : kecepatan kapal

Sehingga didapatkan persamaan :

$$V_x^2 = \frac{2g(\alpha - \mu d) \cdot s_I \left( 1 - \frac{c}{3} \cdot s_I^2 \right) + v_{xi}^2}{K \cdot s_I^2} \quad [2.39]$$

dimana :  $v_{xi}$  : kecepatan di akhir periode I

$s_I$  : jarak yang ditempuh sampai akhir periode I

K dan c merupakan konstanta yang besarnya tergantung dari :

$$c = \frac{1 - \frac{P - \gamma V}{P} \cdot 1^{k \cdot s_I^2}}{s_I^2} \quad [2.40]$$



$$K = \frac{g}{P} \cdot \frac{6,5(W + 9,25W^1)}{S_1} \quad [2.41]$$

### 2.5.3 PERIODE III

Pada periode ini yang perlu diperhatikan adalah pada saat kapal mulai mengapung sepenuhnya,

$$P = \gamma V \quad [2.42]$$

ataupun ujung haluan kapal meninggalkan landasan peluncuran. Sehingga perlu diperhatikan pula akan kemungkinan terjadinya *dropping*, dimana sarat air ujung belakang sepatu peluncur lebih besar dari pada kedalaman perairan.

$$H > T_f \quad [2.43]$$

### 2.5.4 PERIODE IV

Pada periode ini yang perlu diperhatikan adalah pada saat kapal berhenti total baik dengan sendirinya maupun dengan penggereman. Aspek dinamis dari persamaan geraknya juga menggunakan persamaan pada periode II. Pada periode ini dihitung sampai dengan kapal berhenti (kecepatan kapal, v sama dengan nol).

## **BAB III**

### **PERHITUNGAN PELUNCURAN**



## BAB III

### PERHITUNGAN PELUNCURAN

Perhitungan peluncuran kapal secara memanjang meliputi :

- Perhitungan berat dan titik berat total peluncuran
- Perhitungan periode I
- Perhitungan periode II
- Perhitungan periode III
- Perhitungan periode IV

#### 3.1 PERHITUNGAN BERAT DAN TITIK BERAT

Dalam melakukan perhitungan berat dan titik berat peluncuran kapal, dilakukan dengan metode pendekatan berdasarkan *Lloyd Register 1964*.

Pelaksanaan perhitungan dibagi menjadi dua, yaitu :

1. Perhitungan berat dan titik berat kapal sampai *upper deck*
2. Perhitungan berat dan titik berat konstruksi pada bagian-bagian khusus

Sedangkan perhitungan berat dan titik berat peralatan peluncuran ditentukan sendiri dengan ketentuan berat peralatannya sebesar 7% sampai dengan 16% dari berat kapal saat diluncurkan (*Semyonov, 1957* ).



### 3.1.1 PERHITUNGAN BERAT DAN TITIK BERAT KAPAL SAMPAI DENGAN UPPER DECK

Berat kapal sampai dengan *upper deck* dibagi menjadi dua, yaitu :

- a. Berat konstruksi badan kapal sampai dengan *upper deck* dari AP hingga FP
- b. Berat konstruksi badan kapal di luar AP dan FP.

#### a. Berat konstruksi badan kapal sampai dengan *upper deck* dari AP hingga FP

Perhitungan berat dan titik berat kapal ini dimulai dengan menghitung besarnya konstanta m, sebagai berikut :

$$m = 43.4 \cdot H \cdot \frac{Lpp}{10^4} \text{ [ton/m]} \quad [3.1]$$

dengan harga H dicari berdasarkan formula berikut :

$$H = C_1 \cdot B + C_2 \cdot D + C_3 \cdot d + C_4 \cdot W_0 \cdot N_T + C_5 + 2.5 \cdot h \quad [3.2]$$

dimana :

B = lebar kapal

D = tinggi kapal

d = sarat kapal

h = tinggi dasar ganda

$N_T$  = jumlah sekat melintang

Sedangkan nilai dari  $C_1$ ,  $C_2$ ,  $C_3$ ,  $C_4$ ,  $C_5$ , dan  $W_0$  dibaca dari grafik LR'64 dengan menggunakan *polinomial interpolasi beda hingga newton*, sehingga harga H dapat diketahui.



Harga H ini kemudian dimasukkan ke dalam formula mencari harga konstanta m di atas. Kemudian harga m tersebut dipakai sebagai konstanta pengali untuk memperoleh harga distribusi berat tiap station :

$$q = m \cdot a \text{ [ton/m]} \quad [3.3]$$

dimana harga a juga diperoleh dari pembacaan grafik LR'64 dengan menggunakan *polinomial interpolasi beda hingga newton*. Kemudian harga q untuk tiap station ini kemudian disimpson untuk mendapatkan berat dan titik berat konstruksi badan kapal sampai *upper deck* dari AP hingga FP.

**b. Berat konstruksi badan kapal sampai dengan upper deck di luar AP dan FP**

*Konstruksi di belakang AP :*

Panjang pembebanan konstruksi di belakang AP adalah 0,05 Lpp, sedangkan beratnya adalah :

$$W = 0.01133 \cdot m \cdot Lpp \text{ [ton]} \quad [3.4]$$

Kemudian dihitung titik berat konstruksi dari AP sebesar  $1/3 \times 0,05 \times Lpp$ .

*Konstruksi di belakang AP :*

Panjang pembebanannya adalah 0,025 Lpp, sedangkan berat konstruksinya adalah :

$$W = 0.00175 \cdot m \cdot Lpp \text{ [ton]} \quad [3.5]$$

Kemudian dihitung titik berat konstruksi dari FP sebesar  $1/3 \times 0,025 \times Lpp$ .



### 3.1.2 PERHITUNGAN BERAT DAN TITIK BERAT KONSTRUKSI PADA BAGIAN-BAGIAN KHUSUS

#### a. Ruangan Poop

Berat ruangan *Poop* adalah :

$$W = 0.1292 \cdot V \text{ [ton]} \quad [3.6]$$

dimana  $V$  adalah volume *Poop* ( $\text{m}^3$ ). Berat ini didistribusikan secara linier sepanjang *Poop*. Kemudian dihitung titik beratnya terhadap *midship* kapal.

#### b. Ruangan Forecastle

Berat ruangan *Forecastle* adalah :

$$W = 0.0897 \cdot V \text{ [ton]} \quad [3.7]$$

dimana  $V$  adalah volume *Forecastle* ( $\text{m}^3$ ). Berat ini didistribusikan berbentuk segitiga sepanjang *Forecastle*. Kemudian dihitung titik beratnya terhadap *midship* kapal.

#### c. Anjungan Kapal

Berat anjungan kapal adalah :

$$W = 0.1292 \cdot V \text{ [ton]} \quad [3.8]$$

dimana  $V$  adalah volume anjungan kapal ( $\text{m}^3$ ). Berat ini didistribusikan merata sepanjang anjungan. Kemudian dihitung titik beratnya terhadap *midship* kapal.

#### d. Rumah Geladak dan Mast House

Berat rumah geladak dan *Mast House* adalah :

$$W = 0.1185 \cdot V \text{ [ton]} \quad [3.9]$$



dimana  $V$  adalah volume *House* ( $\text{m}^3$ ). Berat ini didistribusikan secara merata sepanjang rumah geladak atau *Mast House*. Kemudian dihitung titik beratnya terhadap *midship* kapal.

*e. Ruang Akomodasi di bawah Upper Deck*

Berat ruang akomodasi adalah :

$$W = 0.0338 \cdot V \text{ [ton]} \quad [3.10]$$

dimana  $V$  adalah volume ruang akomodasi ( $\text{m}^3$ ). Berat ini didistribusikan secara merata sepanjang ruang akomodasi. Kemudian dihitung titik beratnya terhadap *midship* kapal.

*f. Cargo Gear*

Berat *Cargo gear* ini termasuk derek, jangkar, tali dan lain-lain adalah :

$$W = \left[ 0.08 \cdot \Sigma(\text{SWL})^2 + 5 \cdot nb \right] + 10 \text{ [ton]} \quad [3.11]$$

dimana :

$\Sigma(\text{SWL})^2$  : jumlah kuadrat beban kerja yang aman dari masing-masing derek pada posisinya.

nb : jumlah derek pada posisinya

Berat ini didistribusikan merata sepanjang 0,02 L pada posisinya. Kemudian dihitung titik beratnya terhadap *midship* kapal.

*g. Deep dan Peak tank*

Berat konstruksi *deep tank* dan *peak tank* tanpa menggunakan *cofferdam* adalah :

$$W = 0.0538 \cdot V \text{ [ton]} \quad [3.12]$$

Sedangkan untuk konstruksi yang memakai *cofferdam* adalah :



$$W = 0.10005 \cdot V \text{ [ton]} \quad [3.13]$$

dimana  $V$  adalah volume tangki ( $\text{m}^3$ ). Berat ini didistribusikan merata sepanjang tangki. Kemudian dihitung titik beratnya terhadap *midship* kapal.

#### ***h. Kamar Mesin***

Sebelum menghitung berat dari kamar mesin ini, terlebih dulu ditentukan apakah kapal yang akan diluncurkan ini untuk muatan berat (*heavy cargo*) atau tidak. Setelah itu baru perhitungan beratnya dilakukan, yaitu :

$$W = W_m + (0.011 \cdot L + 0.73) \cdot l_0 \text{ [ton]} \text{ untuk muatan berat} \quad [3.14]$$

$$W = W_m + (0.044 \cdot L + 1.47) \cdot l_0 \text{ [ton]} \text{ untuk muatan lain} \quad [3.15]$$

dimana :

$W_m$  : berat mesin utama + mesin bantu

$l_0$  : panjang kamar mesin

Berat ini didistribusikan merata sepanjang kamar mesin. Kemudian dihitung titik beratnya terhadap *midship* kapal.

#### ***i. Propeller dan Poros***

Jika berat baling-baling tidak diketahui, maka dibaca terlebih dahulu harga  $S$  yang merupakan fungsi  $ls/L$  pada grafik LR'64. Kemudian harga  $S$  ini dimasukkan ke dalam perhitungan berat, yaitu :

$$W = ls \cdot (0.0164 \cdot Lpp + S) \text{ untuk } single \text{ screw ship} \quad [3.16]$$

$$W = 1.5 \cdot ls \cdot (0.0164 \cdot Lpp + S) \text{ untuk } twin \text{ screw ship} \quad [3.17]$$

Sedangkan apabila berat baling-baling diketahui, maka perhitungan beratnya adalah :

$$W = W_p + 0.67 \cdot ls \text{ untuk } single \text{ screw ship} \quad [3.18]$$



$$W = W_p + 1.0 \cdot l_s \text{ untuk twin screw ship}$$

[3.19]

dimana :

$W_p$  : berat propeller (dari suplier)

$l_s$  : panjang terowongan poros dari tengah baling-baling sampai sekat belakang kamar mesin

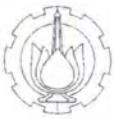
Berat ini didistribusikan merata sepanjang terowongan poros. Kemudian dihitung titik beratnya terhadap *midship* kapal.

### 3.2 PERHITUNGAN PERIODE I

Sebelum menginjak ke langkah periode I, terlebih dahulu dilakukan perhitungan kondisi sepatu peluncur maupun landasan peluncuran. Dimensi dari sepatu peluncur dimasukkan dalam program sebagai input program yaitu panjang sepatu peluncur ( $S$ ), jumlah sepatu peluncur, tekanan terhadap landasan berdasarkan fungsi panjang kapal (*Bakri dkk, 1983*), lebar dan tinggi sepatu peluncur ( $hspt$ ). Pada landasan peluncuran diperlukan input data berupa sudut kemiringan landasan terhadap permukaan air ( $\alpha$ ), panjang landasan yang tercelup air ( $\lambda$ ), dan sarat air di ujung landasan (Hujung).

Pemeriksaan awal sebelum kapal diluncurkan adalah kondisi sarat haluan kapal setelah diluncurkan harus memenuhi atau lebih kecil dari sarat air di ujung landasan agar tidak terjadi *jumping* atau *dropping*.

Setelah itu kemudian dicek apakah kapal mempunyai gaya dorong sendiri karena berat kapal atau harus memberi gaya dorong dari luar, seperti pada pers. [2.23]. Begitu juga dengan pembebanan rata-rata pada periode ini, seperti pada Pers.



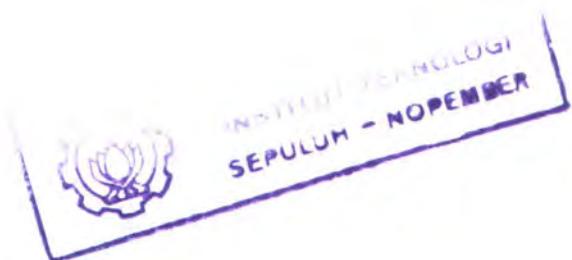
[2.31] dan [2.34]. Kemudian dilakukan perhitungan persamaan geraknya yang terdiri dari kecepatan, percepatan, dan jarak yang ditempuh sebagai fungsi waktu, seperti pada Pers. [2.30].

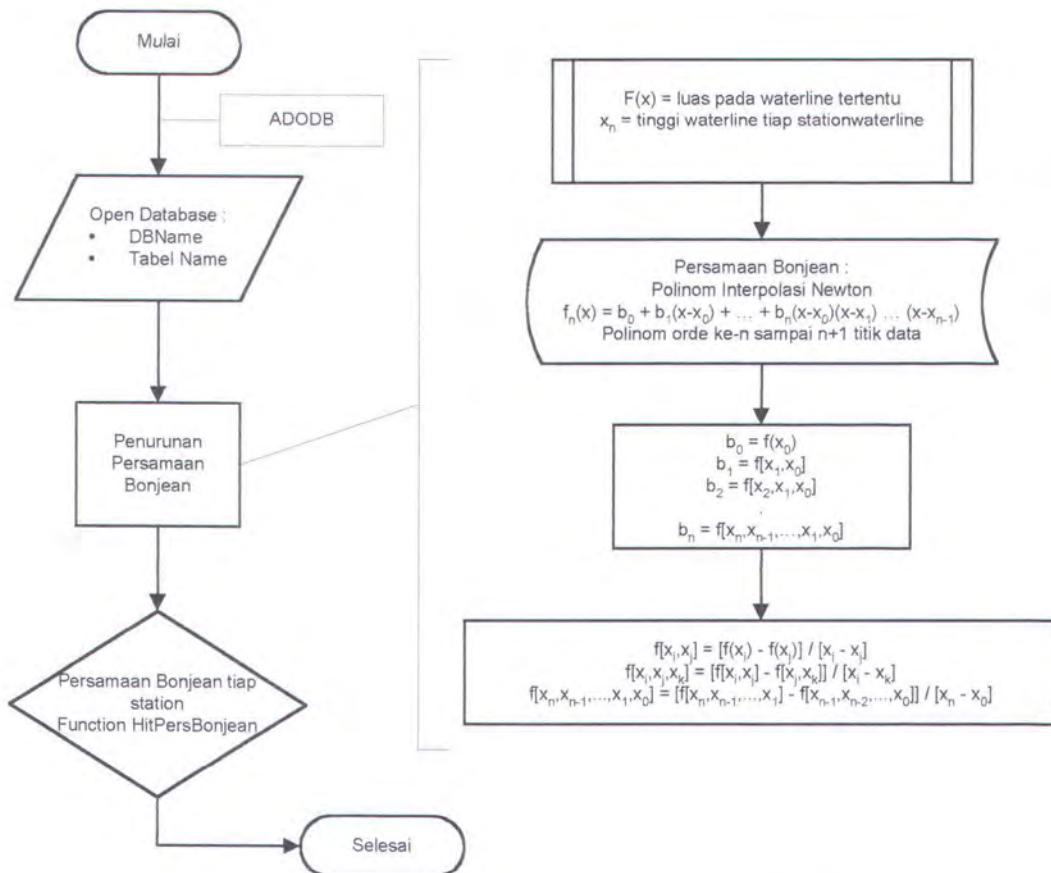
### 3.3 PERHITUNGAN PERIODE II

Sebelum menginjak pada perhitungan periode II, terlebih dahulu *user* atau pengguna harus melakukan input data dari bonjean kapal. Data bonjean kapal ini kemudian dicari persamaannya untuk tiap *station* dengan menggunakan polinomial interpolasi newton (*Gambar 3.1*). Persamaan ini nantinya dipakai untuk menghitung gaya angkat ke atas kapal pada tiap langkahnya.

Perhitungan pada periode II ini dibagi menjadi beberapa langkah yang menunjukkan jarak meluncurnya kapal sampai kapal pada kondisi *stern lift*. Dimana tiap langkahnya diambil dua jarak *station* ( $L_{pp}/20 \times 2$ ).

Untuk menghitung volume *displacemen* ( $\gamma V'$ ) dan LCB kapal tiap langkahnya diambil dari persamaan bonjean tiap station kapal lalu dicari volume gaya angkat ke atasnya. Sehingga harga dari volume displacemen ini akan berubah atau bertambah terus untuk setiap langkahnya.

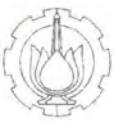




Gambar 3.1. Penurunan Persamaan Bonjean

Selanjutnya dilakukan perhitungan untuk tiap langkahnya sampai dengan lengkap momen antara  $\gamma V' x d$  sama dengan  $P_{total} x c$  sesuai pada Gambar 2.9 atau kondisi *stern lift* terjadi seperti pada Pers. [2.35].

Selain itu pada periode ini juga dicek harga dari Momen Anti Tipping (MAT) -nya. Dimana harga  $\gamma V' x b$  harus selalu lebih besar dari  $P x a$  seperti pada Pers. [2.36]. Jadi dalam alur programnya pada setiap langkah dicek apakah MAT-nya positif atau negatif. Jika kondisi ini tidak terpenuhi maka program akan memberikan *error message* dan kembali ke permulaan perhitungan.



### 3.4 PERHITUNGAN PERIODE III

Perhitungan pada periode ini melanjutkan perhitungan periode II sampai dengan kapal mengalami *free floating* dimana harga  $\gamma V'$  sama dengan  $P_{total}$  atau kapal sudah meninggalkan ujung landasan peluncuran seperti pada Pers. [2.42].

Tetapi untuk tiap langkah dilakukan perhitungan tiap tiga sarat air yang berbeda. Dari hasil perhitungan ketiga kondisi sarat tersebut dapat diketahui sarat sebenarnya dan volume *displacement* sebenarnya untuk tiap langkah. Sehingga didapat harga  $\gamma V'$  sampai dengan memenuhi syarat *free floating*.

### 3.5 PERHITUNGAN PERIODE IV

Perhitungan pada periode ini melanjutkan perhitungan periode III sampai dengan kapal berhenti dengan sendirinya karena adanya tahanan badan kapal terhadap air seperti pada Pers. [2.50].

### 3.6 PERHITUNGAN ASPEK DINAMIS

Perhitungan aspek dinamis kapal yang meliputi hubungan antara S (jarak yang ditempuh), v (kecepatan kapal), dan a (percepatan kapal) terhadap fungsi waktu, dibagi menjadi dua bagian yaitu :

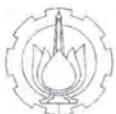
- ▶ Percepatan konstan : yang dimulai dari kapal mulai bergerak sampai dengan kapal menyentuh permukaan air
- ▶ Percepatan tidak konstan : yang terjadi pada saat kapal mulai menyentuh permukaan air, dimana terjadi tahanan air yang menimbulkan percepatan kapal berkurang.



Perhitungan untuk bagian pertama dimana percepatan konstan, menggunakan Persamaan [2.27] sampai dengan Persamaan [2.29]. Sedangkan untuk perhitungan bagian kedua dimana percepatan berubah-ubah, menggunakan Persamaan [2.37] sampai dengan Persamaan [2.41].

## **BAB IV**

### **TEKNIK PEMROGRAMAN**



## BAB IV

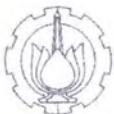
### TEKNIK PEMROGRAMAN

#### 4.1 TEKNIK PEMROGRAMAN

Penyusunan program dalam Tugas Akhir ini menggunakan bahasa pemrograman *Microsoft Visual Basic 6.0*. Dimana bahasa pemrograman ini sangat umum dipakai dalam aplikasi program komputer di kehidupan sehari-hari. Disamping itu, program aplikasi peluncuran kapal ini juga melibatkan atau menggunakan *software* lain yang dibutuhkan dalam aplikasi ini yaitu *Microsoft Access 2000* yang dipakai untuk menyimpan data bonjean kapal dan *AutoCAD 2000* yang dipakai sebagai output program berupa diagram peluncuran.

Dalam kenyataan seringkali program harus ditulis cukup panjang, sehingga kesalahan yang mungkin dibuat oleh seorang programer semakin besar. Untuk mengatasi masalah tersebut dapat dilakukan dengan memecah program menjadi bagian-bagian kecil (rutan). Rutan-rutan kecil tersebut akan membuat penelusuran dan perawatan program menjadi lebih mudah dan terstruktur. Rutin-rutan kecil ini disebut dengan nama prosedur. Dalam Visual Basic terdapat 2 macam prosedur yaitu :

- Prosedur umum (*general purpose procedure*) yaitu prosedur yang ditemukan di dalam daftar *drop down* pada jendela kode.
- Prosedur kejadian (*event procedure*) yaitu prosedur yang berisi kode yang dijalankan ketika suatu kejadian dari kontrol diaktifkan.



Adapun prosedur-prosedur yang ditambahkan dalam sebuah program tersebut disebut *subprogram*. Ada 2 jenis subprogram dalam Visual Basic yaitu :

- Prosedur subrutin (*subroutine procedures*)
- Prosedur fungsi (*function procedures*)

Dalam program peluncuran ini terdapat beberapa *form* beserta *source code*-nya. *Form* adalah *window* yang merupakan komponen dasar dari aplikasi yang dibuat dan *source code*-nya bisa berupa tipe data, konstanta, variabel, prosedur atau fungsi. Program utama berisi program proyek yang dibuat dan disimpan dalam bentuk file berekstensi \*.vbp yaitu *Peluncuran.vbp*. Sedangkan form-nya disimpan dalam file berekstensi \*.frm.

Ada dua form dan satu modul yang dibuat dalam penyusunan program ini, yaitu :

- Form 1 (frmMain.frm) ; merupakan window utama program.
- Form 2 (frmSplash.frm) ; digunakan sebagai tampilan awal ketika program dijalankan.
- Modul 1 (modAwal.bas) ;

Adapun rutin-rutin utama yang ada pada *source code* program adalah :

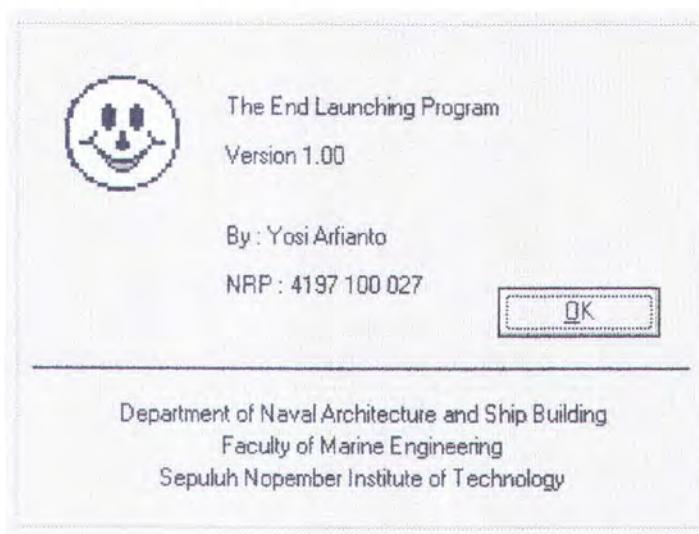
- Prosedur OpenDatabase ; berisi rutin-rutin untuk memanggil data bonjean dari MS.Access berupa file berekstensi \*.mdb.
- Prosedur recordDatabase ; berisi rutin-rutin yang menyimpan informasi dari database bonjean untuk kemudian diproses.



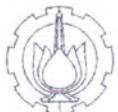
- Fungsi hitPersBonjean ; berisi rutin-rutin yang digunakan untuk menghitung persamaan bonjean untuk tiap station dengan menggunakan metode interpolasi polinomial newton.
- Prosedur PERIODE1, PERIODE2, PERIODE3 ; berisi rutin-rutin yang digunakan untuk melakukan perhitungan peluncuran pada tiap periode peluncuran.
- Prosedur yDrawing ; berisi rutin-rutin yang digunakan untuk menghubungkan program peluncuran dengan aplikasi AutoCAD untuk kemudian menampilkan diagram peluncuran dalam aplikasi tersebut.

## 4.2 PANDUAN PEMAKAIAN PROGRAM

Ketika program peluncuran ini dijalankan maka secara otomatis tampilan awal program (frmSplash.frm) dipanggil (*Gambar 4.1*) yang menggambarkan informasi tentang program peluncuran ini..



*Gambar 4.1.* Tampilan awal program (frmSplash.frm)



Kemudian selanjutnya program akan menampilkan form utama yaitu *frmMain.frm* yang berisi beberapa panel dan yang aktif adalah panel input (*Gambar 4.2*). Panel input ini berisi textbox yang harus diisi sebagai input program yaitu : nama kapal yang digunakan untuk mengambil data bonjean, Lpp, berat dan LCG kapal, berat dan LCG peralatan, panjang, lebar, tinggi dan jumlah sepatu peluncur, panjang landasan, panjang landasan dibawah air, sudut kemiringan landasan, koeffisien gesek statis dan dinamis, sarat haluan dan buritan, serta jarak ujung sepatu terhadap permukaan air.

The screenshot shows a Windows application window titled "The End Launching Program". The window has a menu bar with "File" and several toolbar icons. Below the toolbar is a tab bar with "Input" selected, followed by "Data Kepayang", "Pengukuran", "Format", and "Bantuan".

The "Input" tab contains two main sections: "Data Kapal" and "Landasan Peluncuran".

**Data Kapal:**

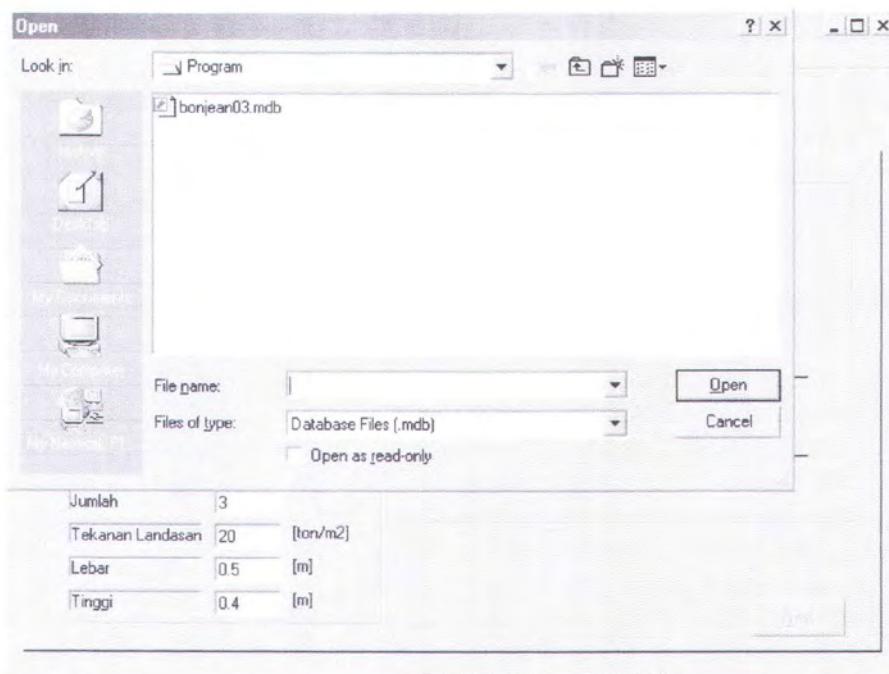
Nama Kapal	Arcapada
Lpp	101 [m]
Berat Kapal	1720.37 [ton]
LCG Kapal	-7.96 [m]
Berat Alat	10 [2] 172.037 [ton]
LCG Alat	0 [m]

**Landasan Peluncuran:**

Panjang landasan	100 [m]
Panjang di bawah air	33.667 [m]
Sudut kemiringan	1/22 = 2.603 °
Koeff. gesek statis	0.032 [m]
Koeff. gesek dinamis	0.02 [m]
Sarat buritan	3.18 [m]
Sarat haluan	0.52 [m]
Jarak ujung spt. thd permukaan air	10 [m]

*Gambar 4.2.* Form input program

Setelah input program diisi, selanjutnya dipanggil data bonjean kapal berupa file MS.Access \*.mdb dari menubar File → Open Database Bonjean. Kemudian akan ditampilkan kotak dialog Open File (*Gambar 4.3*).



Gambar 4.3. Open Database Bonjean

Setelah file database diambil, maka tombol Next pada form input akan aktif. Selanjutnya data bonjean kapal akan ditampilkan dalam panel Data Bonjean (Gambar 4.4).

WaterLine	Station0	Station1	Station2	Station3	Station4	Station5	Station6	S
0	0	0	0	0	0	0	0	0
1	0	0.65	1.76	4.36	7.77	10.53	13	1
2	0	1.85	5.34	11.29	18.27	23.47	27.9	3
3	0	3.54	10.16	19.85	30.13	37.43	43.45	4
4	0	5.84	16.19	29.64	42.91	51.98	59.14	6
6.61	3.05	20.61	39.75	60	78.87	91.31	100.73	11

Gambar 4.4. Tampilan data bonjean kapal



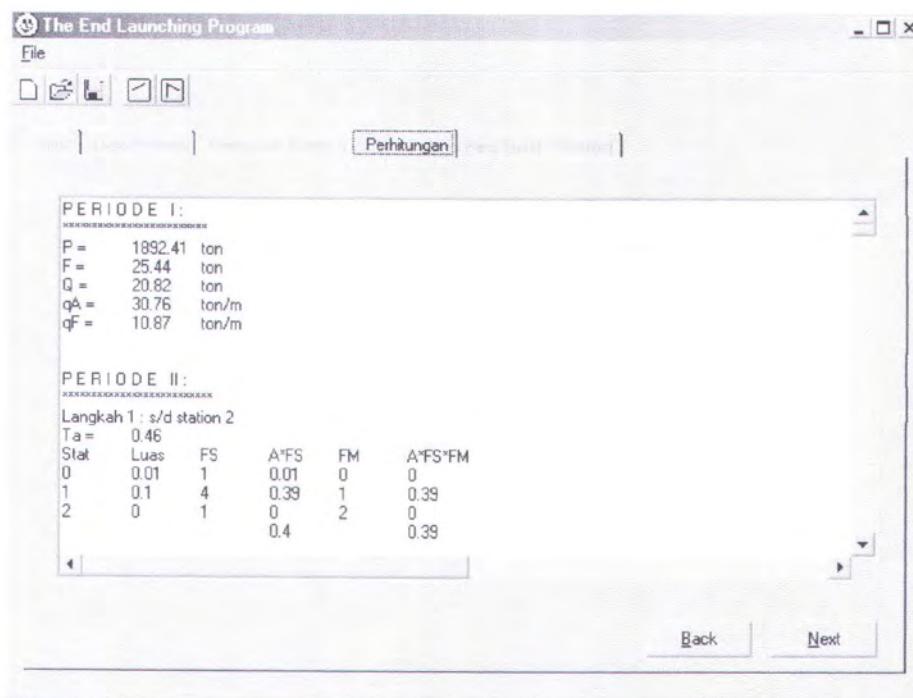
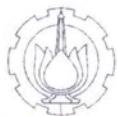
Kemudian dapat juga dilihat persamaan bonjean kapal untuk tiap *station* pada panel berikutnya (*Gambar 4.5*).

The screenshot shows a software window titled "The End Launching Program". The main area displays a list of polynomial equations for ship stations, starting from station 0 and ending at station 17. The equations are listed in a scrollable text area. At the bottom of the window, there are "Back" and "Next" buttons.

```
Pers. station 0 = 0 + (x - 0) (0) + (x - 0)(x - 1) (0) + (x - 0)(x - 1)(x - 2) (0) + [x - 0](x - 1)(x - 2)(x - 3) (0) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (0)
Pers. station 1 = 0 + (x - 0) (0.65) + (x - 0)(x - 1) (0.275) + (x - 0)(x - 1)(x - 2) (-1.000000000000001E-02) + (x - 0)(x - 1)(x - 2)(x - 3) (-9.66666666666666E-02) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.228333333333333)
Pers. station 2 = 0 + (x - 0) (1.76) + (x - 0)(x - 1) (0.91) + (x - 0)(x - 1)(x - 2) (-9.66666666666666E-02) + (x - 0)(x - 1)(x - 2)(x - 3) (-0.1566666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.2316666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.208333333333332)
Pers. station 3 = 0 + (x - 0) (4.36) + (x - 0)(x - 1) (1.285) + (x - 0)(x - 1)(x - 2) (-0.228333333333333) + (x - 0)(x - 1)(x - 2)(x - 3) (-0.1566666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.2316666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.208333333333332)
Pers. station 4 = 0 + (x - 0) (7.77) + (x - 0)(x - 1) (1.365) + (x - 0)(x - 1)(x - 2) (-0.228333333333333) + (x - 0)(x - 1)(x - 2)(x - 3) (-0.1566666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.2316666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.208333333333332)
Pers. station 5 = 0 + (x - 0) (10.53) + (x - 0)(x - 1) (1.205) + (x - 0)(x - 1)(x - 2) (-0.2316666666666666) + (x - 0)(x - 1)(x - 2)(x - 3) (-0.1566666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.2316666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.208333333333332)
Pers. station 6 = 0 + (x - 0) (13) + (x - 0)(x - 1) (0.949999999999999) + (x - 0)(x - 1)(x - 2) (-0.208333333333332) + (x - 0)(x - 1)(x - 2)(x - 3) (-0.1566666666666666) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-0.208333333333332)
Pers. station 7 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 8 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 9 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 10 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 11 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 12 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 13 = 0 + (x - 0) (15.83) + (x - 0)(x - 1) (0.260000000000001) + (x - 0)(x - 1)(x - 2) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 14 = 0 + (x - 0) (14.05) + (x - 0)(x - 1) (0.565) + (x - 0)(x - 1)(x - 2) (-0.12) + (x - 0)(x - 1)(x - 2)(x - 3) (2.45E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 15 = 0 + (x - 0) (11.9) + (x - 0)(x - 1) (0.785) + (x - 0)(x - 1)(x - 2) (-0.168333333333334) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 16 = 0 + (x - 0) (9.32) + (x - 0)(x - 1) (0.779999999999999) + (x - 0)(x - 1)(x - 2) (-9.16666666666656E-01) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
Pers. station 17 = 0 + (x - 0) (6.41) + (x - 0)(x - 1) (0.734999999999999) + (x - 0)(x - 1)(x - 2) (-0.104999999999999) + (x - 0)(x - 1)(x - 2)(x - 3) (-8.66666666666669E-01) + (x - 0)(x - 1)(x - 2)(x - 3)(x - 4) (-8.66666666666669E-01)
```

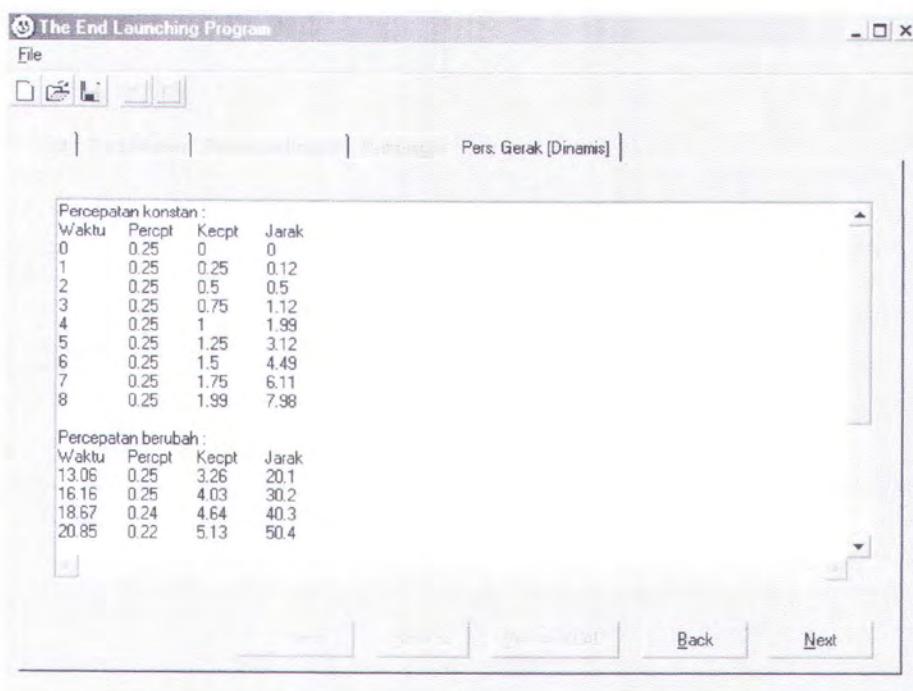
*Gambar 4.5.* Tampilan persamaan bonjean kapal

Selanjutnya dapat dilihat hasil perhitungan peluncuran mulai dari Periode I sampai Periode III (*Gambar 4.6*).



Gambar 4.6. Tampilan hasil perhitungan peluncuran

Demikian juga dengan hasil perhitungan persamaan gerak (dinamis) kapal dapat dilihat pada Gambar 4.7.



Gambar 4.7. Tampilan hasil perhitungan dinamis

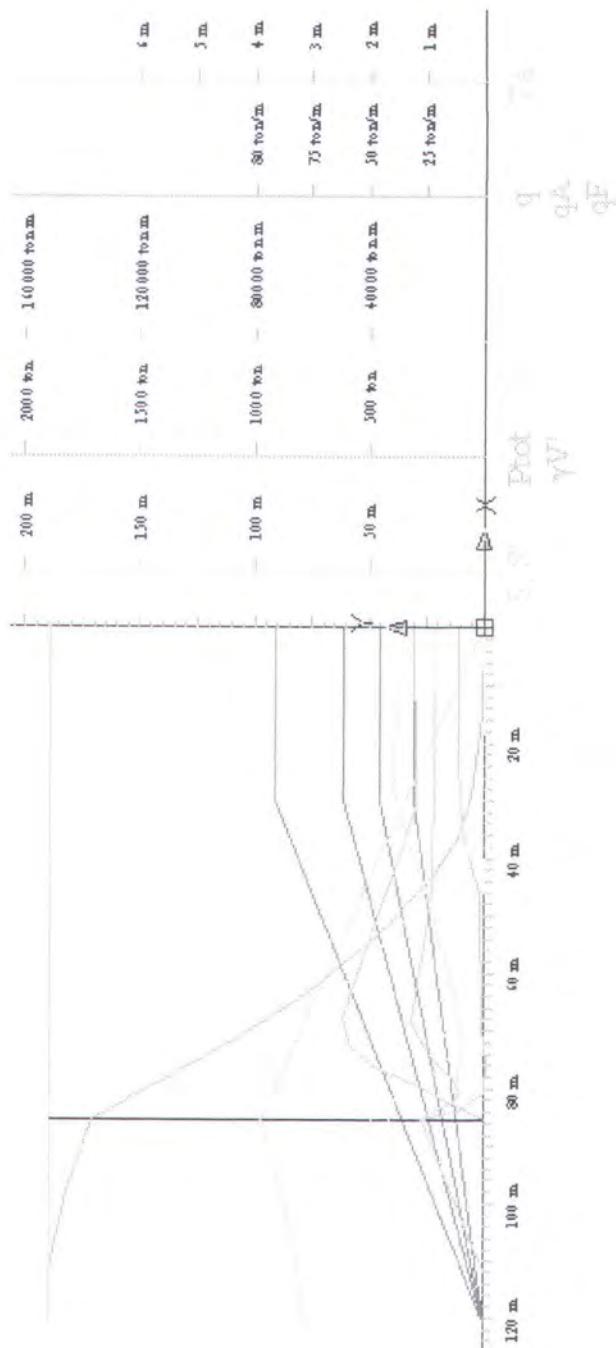


Kemudian hasil output diagram peluncuran dapat dilihat dengan menekan tombol

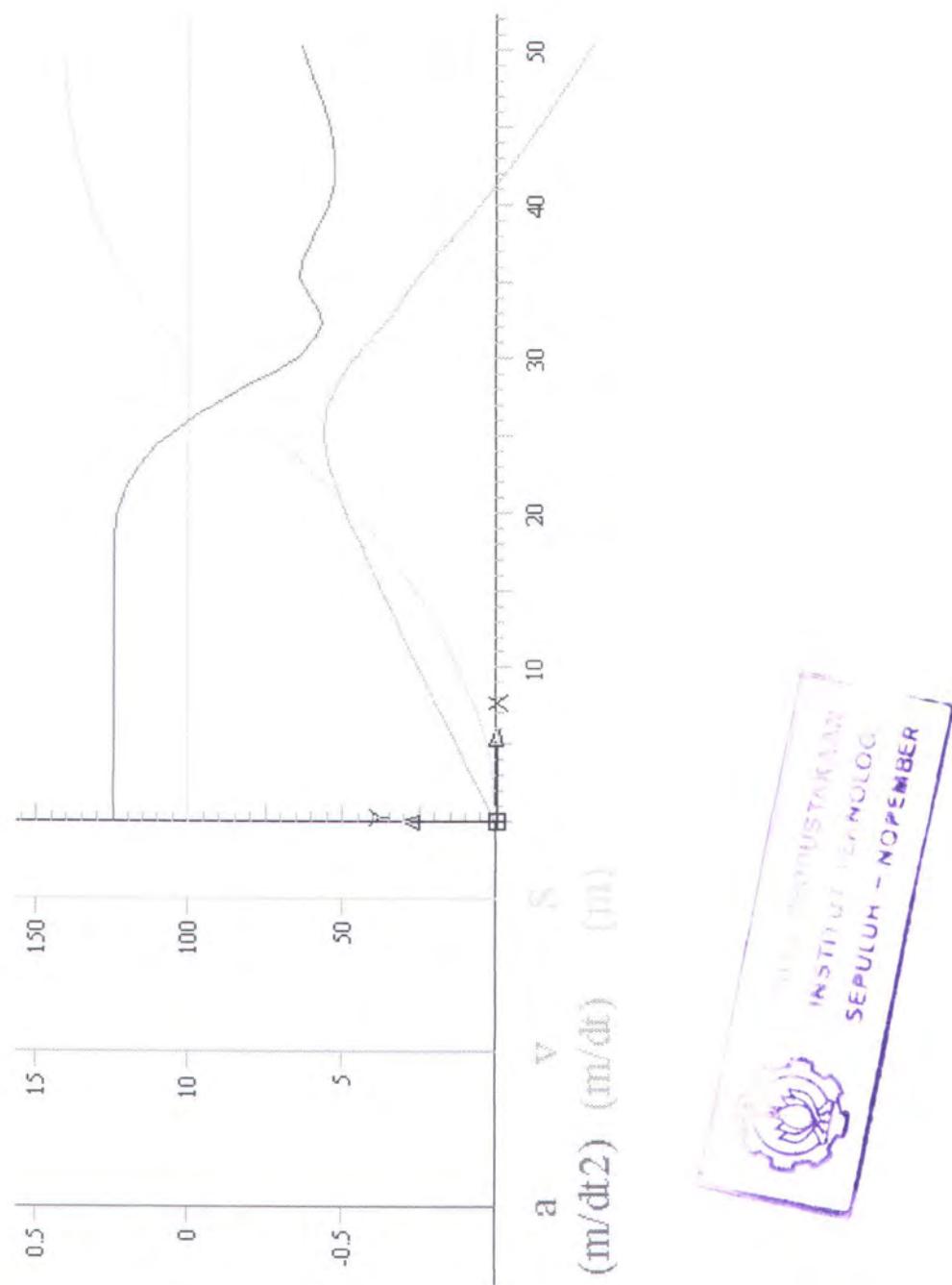
View AutoCAD yang akan langsung menghubungkan program dengan aplikasi

AutoCAD dan menampilkan diagram peluncuran beserta aspek dinamisnya

(Gambar 4.8 dan 4.9).

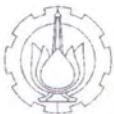


Gambar 4.8. Tampilan diagram peluncuran pada AutoCAD

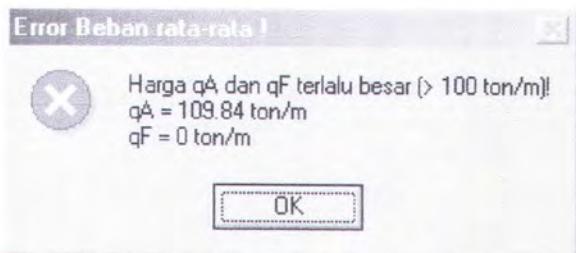


Gambar 4.9. Tampilan aspek dinamis peluncuran pada AutoCAD

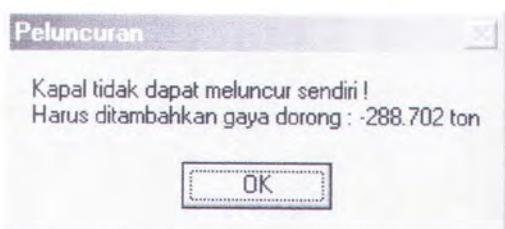
Contoh *error message* yang akan ditampilkan jika terjadi kesalahan perhitungan, misalnya kapal tidak dapat bergerak dengan sendirinya atau terjadi



pembebanan terhadap landasan yang besar ( $> 100 \text{ ton/m}$ ) seperti yang ditunjukkan pada Gambar 4.10 dan 4.11.



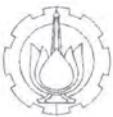
Gambar 4.10. Error pada harga pembebanan



Gambar 4.11. Error karena kapal kurang gaya dorong

## **BAB V**

### **ANALISIS HASIL PROGRAM**



## BAB V

### ANALISIS HASIL PROGRAM

Hasil dari pemrograman komputer ini harus diperhatikan keakurasiannya dengan membandingkannya dengan perhitungan secara manual. Validasi dari hasil pemrograman dibandingkan dengan perhitungan manual (MS. Excel) seperti pada penggerjaan Tugas Rancang Peluncuran. Hal ini perlu dilakukan untuk mengecek apakah hasil dari program yang telah dibuat dapat menggantikan perhitungan manual dengan lebih cepat dan teliti.

Adapun data kapal yang akan dipakai adalah sebagai berikut :

Lpp	= 101 m	T	= 6.61 m
B	= 16.35 m	Tipe	= General Cargo
H	= 8.2 m		

Hasil program peluncuran kapal yang dianalisis meliputi :

- Pembacaan luasan bonjean kapal (diambil pada salah satu langkah perhitungan Periode II).
- Hasil perhitungan peluncuran beserta analisis kondisi kritisnya.

Data bonjean kapal (dalam bentuk database MS. Access) dapat dilihat pada *Tabel 1*, Lampiran I.



## 5.1 PEMBACAAN LUASAN BONJEAN KAPAL

Pembacaan kurva luasan bonjean kapal menggunakan metode interpolasi polinomial newton. Data bonjean kapal diambil dari database bonjean yang sudah ada (MS Access). Hasil program komputer dapat dilihat pada Lampiran I. Sedangkan perhitungan manual dapat dilihat pada Lampiran II. Dari kedua perhitungan tersebut dihitung selisih penyimpangan dari hasil program terhadap hasil perhitungan manual, dimana faktor koreksinya diusahakan harus  $< 5\%$ .

Dari hasil perhitungan program dapat dibandingkan hasil simpson luas bonjean  $\Sigma_1$  dan  $\Sigma_2$  dari masing-masing cara perhitungan sebagai berikut :

Program komputer :

Tabel 1. Hasil program untuk luasan bonjean

Langkah	$\Sigma_1$	$\Sigma_2$
1	0.4	0.39
2	5.36	11.98
3	31.39	107.58
4	99.09	440.17
5	218.1	1151.1
6	387.46	2326.04
7	606.28	4051.25
8	867.89	6325.74
9	1159.95	9031.17

Perhitungan manual :

Tabel 2. Hasil manual untuk luasan bonjean

Langkah	$\Sigma_1$	$\Sigma_2$
1	0.4	0.4
2	5.56	12.72
3	31.76	109.52
4	98.88	434.08
5	216.16	1131.6
6	382.96	2292
7	599.24	3999.04
8	858.96	6255.36
9	1147.96	8927.12



Koreksi antara perhitungan program dan manual :

$$\text{Koreksi } \Sigma_1 = \frac{\Sigma_1 \text{program} - \Sigma_1 \text{manual}}{\Sigma_1 \text{manual}} \times 100\%$$

$$\text{Koreksi } \Sigma_2 = \frac{\Sigma_2 \text{program} - \Sigma_2 \text{manual}}{\Sigma_2 \text{manual}} \times 100\%$$

Diperoleh hasil koreksinya seperti yang ditunjukkan oleh *Tabel 3*.

*Tabel 3.* Hasil koreksi untuk luasan bonjean

Langkah	Koreksi $\Sigma_1$ %	Koreksi $\Sigma_2$ %
1	0.000	-2.500
2	-3.597	-5.818
3	-1.165	-1.771
4	0.212	1.403
5	0.897	1.723
6	1.175	1.485
7	1.175	1.306
8	1.040	1.125
9	1.044	1.166

Dari koreksi di atas dapat diketahui bahwa selisih antara perhitungan manual dan hasil output program peluncuran tidak terlalu besar, sekitar  $< \pm 5\%$ . Jadi masih dapat dikatakan bahwa hasil output program masih layak.

Hal ini dapat dimaklumi karena dalam metode *interpolasi polinomial newton*, jumlah atau *range* antar data input untuk membuat persamaan sangat berpengaruh terhadap keakuratan fungsi persamaan yang akan dihasilkan. Semakin banyak atau semakin sempit *range* data input yang dimasukkan maka semakin akurat hasil persamaan yang akan dihasilkan, begitu juga sebaliknya.



## 5.2 HASIL PERHITUNGAN PELUNCURAN DAN ANALISIS KONDISI KRITIS

Pada rekapitulasi perhitungan Periode II, poin-poin yang dianalisis meliputi gaya apung  $\gamma V'$  dan momen gaya apung terhadap ujung sepatu  $\gamma V' \times d$ . Sedangkan poin-poin lainnya tidak dibahas karena tidak berhubungan dengan persyaratan *stern lift* kapal.

Dari hasil *running* program peluncuran diperoleh rekapitulasi perhitungan Periode II seperti yang ditunjukkan oleh *Tabel 4*.

*Tabel 4.* Rekapitulasi Periode II program

REKAPITULASI PERHITUNGAN PERIODE II :

Notasi	Satuan	1	2	3	4	5
$gV'$	ton	0.69	9.29	54.38	171.66	377.82
$gV'.d$	ton m	62.45	786.89	4276.93	12619.79	26182.06

Notasi	Satuan	6	7	8	9
$gV'$	ton	671.21	1050.27	1503.46	2009.4
$gV'.d$	ton m	44053.91	65332.13	88918.5	113795.3

Dari perhitungan manual didapat rekapitulasi periode II seperti yang ditunjukkan oleh *Tabel 5*.

*Tabel 5.* Rekapitulasi Periode II manual

REKAPITULASI PERHITUNGAN PERIODE II :

Notasi	Satuan	1	2	3	4	5
$gV'$	ton	0.69	9.63	55.02	171.29	374.46
$gV'.d$	ton m	62.99	812.88	4320.91	12638.01	26029.77

Notasi	Satuan	6	7	8	9
$gV'$	ton	663.41	1038.07	1487.99	1988.63
$gV'.d$	ton m	43603.17	64618.81	88049.66	112713



Koreksi untuk tiap-tiap langkah seperti yang ditunjukkan oleh *Tabel 6*.

*Tabel 6.* Koreksi rekapitulasi Periode II

Notasi	Satuan	1	2	3	4	5
$gV'$	ton	-0.42%	-3.55%	-1.16%	0.22%	0.90%
$gV'.d$	ton m	-0.85%	-3.20%	-1.02%	-0.14%	0.59%

Notasi	Satuan	6	7	8	9
$gV'$	ton	1.18%	1.17%	1.04%	1.04%
$gV'.d$	ton m	1.03%	1.10%	0.99%	0.96%

Dari koreksi di atas dapat diketahui bahwa selisih antara perhitungan manual dan hasil output program peluncuran tidak terlalu besar, sekitar  $\pm 5\%$ . Disamping itu juga diketahui bahwa kapal mengalami *stern lift* antara langkah 8 dan 9. Dan kapal mengalami *free floating* pada langkah antara 11 dan 12.

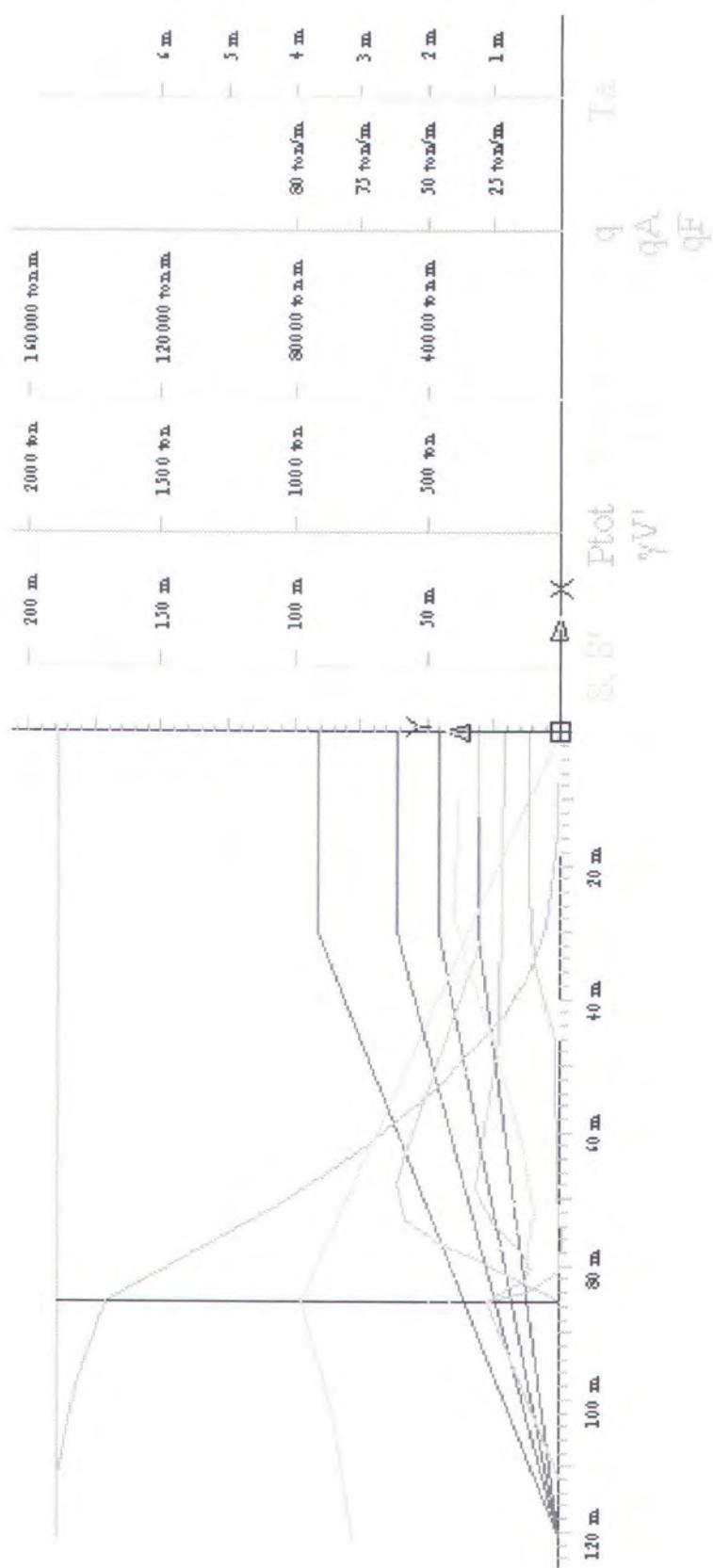
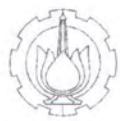
### 5.3 KECEPATAN PEROLEHAN HASIL

Kecepatan dalam memperoleh hasil perhitungan dengan menggunakan program ini hanya membutuhkan waktu kurang dari 1 menit dengan data yang dibutuhkan diantaranya data bonjean kapal, data berat dan titik berat kapal, data landasan peluncuran dan peralatannya sudah tersedia.

Sedangkan jika menggunakan perhitungan secara manual dengan data yang sama membutuhkan waktu pengerjaan lebih dari 5 jam, dan yang paling membutuhkan waktu relatif lama adalah pada saat membaca kurva bonjean kapal.

### 5.4 HASIL OUTPUT DIAGRAM PELUNCURAN

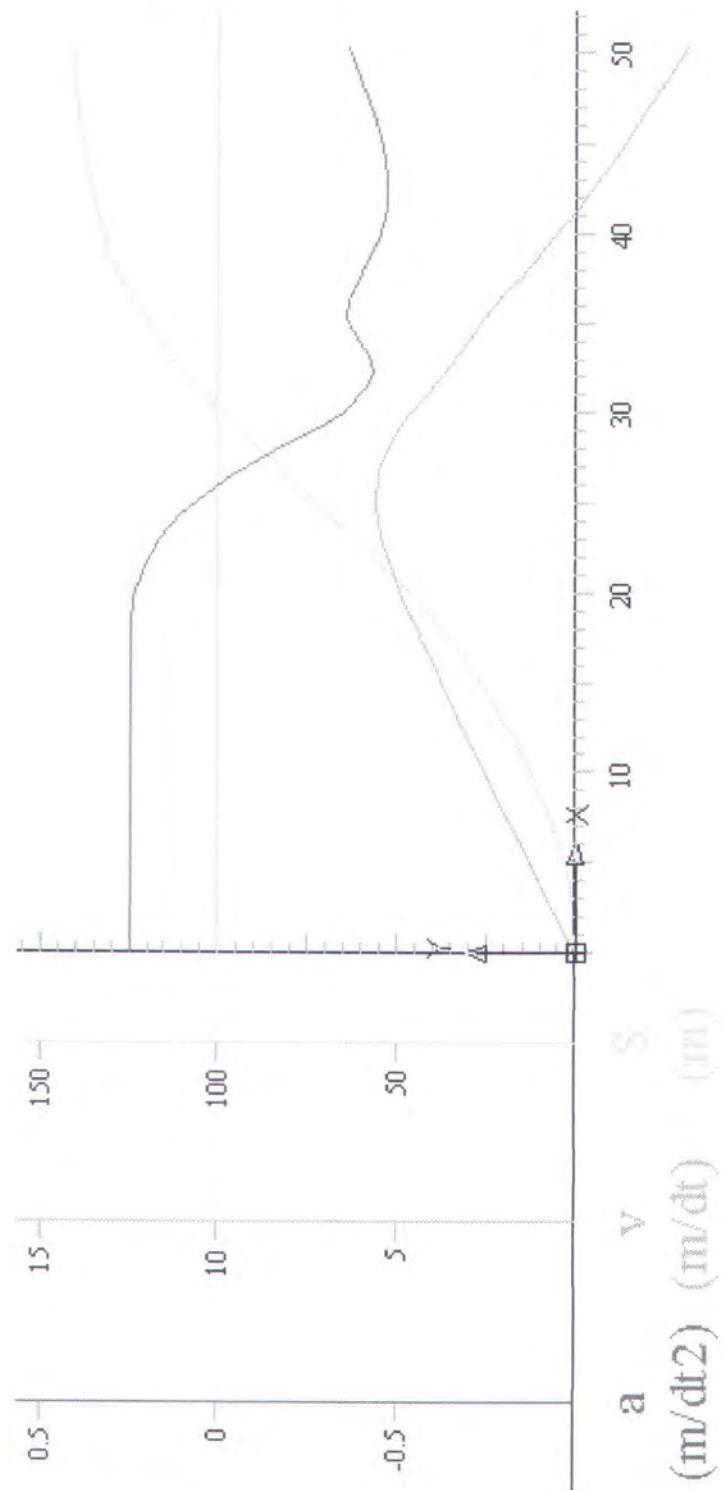
Hasil diagram peluncuran yang dalam bentuk file AutoCAD (\*.dwg) seperti yang ditunjukkan oleh *Gambar 5.1*.



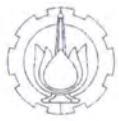
Gambar 5.1. Hasil diagram peluncuran



Hasil dari grafik dinamis kapal pada saat diluncurkan sebagai fungsi waktu dapat dilihat pada *Gambar 5.2*.



*Gambar 5.2.* Hasil aspek dinamis pada peluncuran



Dari hasil output diagram peluncuran di atas dapat diketahui bahwa kapal mengalami *stern lift* setelah kapal berjalan sejauh 85.25 m dari ujung permukaan air. Dan kapal mengalami *free floating* setelah menempuh jarak 111.49 m dari ujung permukaan air.

Sedangkan untuk aspek dinamisnya yang merupakan grafik S (jarak yang ditempuh dari kapal berhenti), v (kecepatan kapal), dan a (percepatan kapal) terhadap fungsi t (waktu) diperoleh hasil bahwa kapal akan berhenti setelah menempuh jarak 136.05 m dari awal mulanya kapal bergerak.

Percepatan kapal mulai berkurang pada saat kapal memasuki Periode II atau kapal mulai mendapatkan tahan dari air laut walaupun pada awalnya masih terlalu kecil. Sehingga sampai dengan percepatan belum negatif, kecepatan kapal masih bertambah. Dari grafik pada *Gambar 5.2*, diketahui bahwa percepatan kapal mulai negatif pada saat kapal telah menempuh jarak 78.36 m.

## **BAB VI**

### **KESIMPULAN DAN SARAN**



## BAB VI

### KESIMPULAN DAN SARAN

Proses perhitungan dan pembuatan diagram peluncuran *end launching* kapal yang memakan waktu yang relatif lama dapat dipercepat pelaksanaannya dengan menggunakan program komputer. Sehingga diperoleh effisiensi waktu serta mempermudah proses perhitungannya. Dengan demikian bahaya-bahaya yang tidak dikehendaki seperti kapal tenggelam, *dropping*, *tipping* dan lainnya dapat diketahui lebih cepat sehingga dapat dilakukan langkah-langkah antisipasinya.

#### 5.1. KESIMPULAN

1. Dari hasil analisis mengenai kecepatan perolehan hasil diperoleh bahwa dengan menggunakan program ini dapat menghemat waktu perhitungan.
2. Hasil perbandingan antara perhitungan manual dan program menunjukkan tingkat keakuriasan yang tinggi dengan rata-rata kesalahan di bawah 5 %.
3. Dengan menggunakan program peluncuran ini, *user* dapat melakukan perhitungan secara cepat dan dapat mengetahui hal-hal yang tidak diinginkan seperti kemungkinan terjadinya *tipping*, *dropping* ataupun *jumping* sedini mungkin.
4. Perbedaan hasil antara program komputer dengan perhitungan manual dikarenakan pada perhitungan manual pembacaan lengkung bonjean



dilakukan secara manual sedangkan pada program ini lengkung bonjean diwakili oleh sebuah fungsi dengan metode interpolasi polinomial newton.

5. Dengan menggunakan *interpolasi polinomial newton* maka semakin banyak data input atau semakin sempit *range waterline* untuk lengkung bonjean maka fungsi persamaan yang dihasilkan juga akan semakin mendekati keadaan sebenarnya.

## 5.2. SARAN PENGEMBANGAN

1. Pengembangan program ini dapat dilakukan dengan menambah penampilan simulasi 3 dimensi proses peluncuran kapal.
2. Untuk penyempurnaan program, jika terjadi pesan error misalnya pada saat tiiping atau dropping bisa ditambah dengan pesan alternatif yang paling aman dilakukan untuk mengatasi masalah tersebut.
3. Pada program ini pembacaan tinggi sarat peluncuran pada lengkung hidrostatik masih dilakukan secara manual sehingga diperlukan hubungan atau *link* antara program ini dengan program lain yang menghitung hidrostatik kapal.

## **DAFTAR PUSTAKA**

## DAFTAR PUSTAKA

1. Bakri, M., Santosa, I G.M., dan Sudjono, "*Teori Bangunan Kapal 3*", Departemen Pendidikan dan Kebudayaan, Direktorat Jendral Pendidikan Dasar dan Menengah, 1983.
2. Chapra, S.C., dan Canale, R.P., "*Numerical Methods for Engineers*", Mc Graw-Hill Book Company, 1985.
3. Andrews, Harrison B., "Launching", "*Principles of Naval Architecture*", SNAME, 1967.
4. Semyonov, Tyan, Shansky, V., "*Statics and Dynamics of the Ship*", Peace Publisher, Moscow, (date-unknown).
5. Taggart, Robert, "*Ship Design and Construction*", (date-unknown).
6. Budisusilo, Purwanto, "*Pembuatan Diagram Peluncuran End Launching Kapal Dibantu Komputer*", Tugas Akhir, Jurusan Teknik Perkapalan FTK-ITS, Surabaya, 2000.
7. QUE, "*Using Visual Basic 6*", MacMillan Computer Publishing, 1998.
8. Wibisono, Waskito, "*Visual Basic Lanjut*", PIKTI – ITS, 2002.
9. <http://www.freevbcodem.com> dan <http://www.vbcodem.com>

## LAMPIRAN



## LAMPIRAN I

### HASIL PERHITUNGAN PROGRAM

Data bonjean kapal yang diambil dari database MS. Access ditunjukkan oleh *Tabel 7*.

*Tabel 7. Database bonjean dalam MS. Access*

WaterLine	Station0	Station1	Station2	Station3	Station4	Station5
0	0	0	0	0	0	0
1	0	0.65	1.76	4.36	7.77	10.53
2	0	1.85	5.34	11.29	18.27	23.47
3	0	3.54	10.16	19.85	30.13	37.43
4	0	5.84	16.19	29.64	42.91	51.98
6.61	3.05	20.61	39.75	60	78.87	91.31

WaterLine	Station6	Station7	Station8	Station9	Station10
0	0	0	0	0	0
1	13	15.83	15.83	15.83	15.83
2	27.9	32.18	32.18	32.18	32.18
3	43.45	48.53	48.53	48.53	48.53
4	59.14	64.88	64.88	64.88	64.88
6.61	100.73	107.55	107.55	107.55	107.55

WaterLine	Station11	Station12	Station13	Station14	Station15
0	0	0	0	0	0
1	15.83	15.83	15.83	14.05	11.9
2	32.18	32.18	32.18	29.23	25.37
3	48.53	48.53	48.53	44.82	39.4
4	64.88	64.88	64.88	60.69	53.84
6.61	107.55	107.55	107.55	102.4	92.47

WaterLine	Station16	Station17	Station18	Station19	Station20
0	0	0	0	0	0
1	9.32	6.41	3.42	0.71	0
2	20.2	14.29	7.92	2.25	0
3	32.09	23.01	13.13	4.41	0
4	44.56	32.39	19.1	7.1	0
6.61	78.44	58.58	16.06	5.13	0



Hasil perhitungan program komputer ditunjukkan oleh *Tabel 8* dan *Tabel 9*.

*Tabel 8.* Hasil Periode I

**PERIODE I:**

\*\*\*\*\*

P =	1892.41	ton
F =	25.44	ton
Q =	20.82	ton
qA =	30.76	ton/m
qF =	10.87	ton/m

*Tabel 9.* Hasil perhitungan tiap langkah Periode II

**PERIODE II:**

\*\*\*\*\*

Langkah 1 : s/d station 2

Ta = 0.46

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0.01	1	0.01	0	0
1	0.1	4	0.39	1	0.39
2	0	1	0	2	0
			0.4		0.39

Langkah 2 : s/d station 4

Ta = 0.92

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0	1	0	0	0
1	0.39	4	1.54	1	1.54
2	0.52	2	1.03	2	2.06
3	0.7	4	2.79	3	8.37
4	0	1	0	4	0
			5.36		11.98

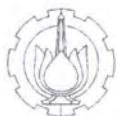
Langkah 3 : s/d station 6

Ta = 1.38

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0	1	0	0	0
1	0.79	4	3.18	1	3.18
2	1.53	2	3.07	2	6.14
3	2.66	4	10.65	3	31.96
4	3.09	2	6.17	4	24.69
5	2.08	4	8.32	5	41.61
6	0	1	0	6	0
			31.39		107.58

Langkah 4 : s/d station 8





Ta = 1.84

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0	1	0	0	0
1	1.32	4	5.27	1	5.27
2	2.94	2	5.89	2	11.77
3	5.25	4	21.01	3	63.03
4	7	2	14.01	4	56.03
5	6.89	4	27.56	5	137.82
6	5.61	2	11.23	6	67.37
7	3.53	4	14.12	7	98.87
8	0	1	0	8	0
			99.09		440.17

Langkah 5 : s/d station 10

Ta = 2.3

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0	1	0	0	0
1	1.95	4	7.79	1	7.79
2	4.66	2	9.32	2	18.65
3	8.33	4	33.33	3	100
4	11.51	2	23.02	4	92.09
5	12.34	4	49.35	5	246.76
6	11.84	2	23.68	6	142.1
7	10.8	4	43.2	7	302.4
8	7.14	2	14.27	8	114.2
9	3.53	4	14.12	9	127.12
10	0	1	0	10	0
			218.1		1151.1

Langkah 6 : s/d station 12

Ta = 2.75

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0	1	0	0	0
1	2.68	4	10.7	1	10.7
2	6.64	2	13.29	2	26.58
3	11.81	4	47.24	3	141.73
4	16.44	2	32.87	4	131.49
5	18.21	4	72.85	5	364.23
6	18.49	2	36.97	6	221.82
7	18.23	4	72.93	7	510.54
8	14.5	2	29.01	8	232.05
9	10.8	4	43.2	9	388.79
10	7.14	2	14.27	10	142.75
11	3.53	4	14.12	11	155.37
12	0	1	0	12	0
			387.46		2326.04

Langkah 7 : s/d station 14

Ta = 3.21



Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0	1	0	0	0
1	3.51	4	14.04	1	14.04
2	8.87	2	17.74	2	35.48
3	15.62	4	62.46	3	187.39
4	21.66	2	43.33	4	173.32
5	24.37	4	97.47	5	487.34
6	25.39	2	50.79	6	304.74
7	25.74	4	102.95	7	720.68
8	21.98	2	43.96	8	351.7
9	18.23	4	72.93	9	656.4
10	14.5	2	29.01	10	290.06
11	10.8	4	43.2	11	475.19
12	7.14	2	14.27	12	171.3
13	3.53	4	14.12	13	183.62
14	0	1	0	14	0
			606.28		4051.25

Langkah 8 : s/d station 16

Ta = 3.67

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	-0.01	1	-0.01	0	0
1	4.46	4	17.86	1	17.86
2	11.34	2	22.68	2	45.36
3	19.7	4	78.81	3	236.44
4	27.13	2	54.25	4	217.02
5	30.71	4	122.86	5	614.29
6	32.46	2	64.92	6	389.53
7	33.26	4	133.04	7	931.27
8	29.5	2	59	8	471.99
9	25.74	4	102.95	9	926.58
10	21.98	2	43.96	10	439.63
11	18.23	4	72.93	11	802.27
12	14.5	2	29.01	12	348.07
13	10.8	4	43.2	13	561.59
14	6.23	2	12.45	14	174.36
15	2.49	4	9.97	15	149.49
16	0	1	0	16	0
			867.89		6325.74

Langkah 9 : s/d station 18

Ta = 4.13

Stat	Luas	FS	A*FS	FM	A*FS*FM
0	0.01	1	0.01	0	0
1	5.58	4	22.31	1	22.31
2	14.07	2	28.14	2	56.28
3	24.05	4	96.19	3	288.57
4	32.79	2	65.58	4	262.3
5	37.2	4	148.81	5	744.06
6	39.61	2	79.22	6	475.33



7	40.77	4	163.09	7	1141.6
8	37.02	2	74.03	8	592.27
9	33.26	4	133.04	9	1197.34
10	29.5	2	59	10	589.98
11	25.74	4	102.95	11	1132.49
12	21.98	2	43.96	12	527.55
13	18.23	4	72.93	13	948.14
14	12.84	2	25.68	14	359.58
15	7.94	4	31.78	15	476.66
16	4.05	2	8.09	16	129.48
17	1.28	4	5.13	17	87.23
18	0	1	0	18	0
			1159.95		9031.17

Hasil perhitungan aspek dinamis kapal ditunjukkan oleh *Tabel 10.*

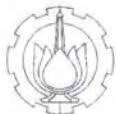
*Tabel 10.* Hasil perhitungan dinamis kapal

Percepatan konstan :

Waktu	Percpt	Kecpt	Jarak
0	0.25	0	0
1	0.25	0.25	0.12
2	0.25	0.5	0.5
3	0.25	0.75	1.12
4	0.25	1	1.99
5	0.25	1.25	3.12
6	0.25	1.5	4.49
7	0.25	1.75	6.11
8	0.25	1.99	7.98

Percepatan berubah :

Waktu	Percpt	Kecpt	Jarak
13.06	0.25	3.26	20.1
16.16	0.25	4.03	30.2
18.67	0.24	4.64	40.3
20.85	0.22	5.13	50.4
22.82	0.17	5.47	60.5
24.66	0.09	5.64	70.6
26.45	-0.03	5.58	80.7
28.26	-0.18	5.25	90.8
30.19	-0.36	4.56	100.9
32.4	-0.43	3.6	111
35.21	-0.36	2.59	111.1
39.11	-0.43	0.9	121.2
50.31	-0.36	-3.13	131.3



## LAMPIRAN II

### PERHITUNGAN MANUAL

Perhitungan manual peluncuran kapal ini sesuai dengan perhitungan pada Tugas Rancang III Peluncuran. Dimana untuk perhitungan berat kapal menggunakan pendekatan *Lloyd Register 1964*.

Rekapitulasi berat kapal kosong dan titik beratnya dapat dilihat pada *Tabel 11*.

*Tabel 11.* Rekapitulasi berat dan titik berat kapal kosong

Nomor	Bagian	Berat (I)	LCG (II)	I x II
1	Baja AP-FP	1280.78	-0.49	-627.72
2	Baja AP-buritan	17.98	-52.37	-941.49
3	Baja FP-haluan	2.78	51.07	141.81
4	Forecastle deck	7.59	47.03	356.82
5	Poop deck	78.51	-41.22	-3236.32
6	Equipment fore	44.63	46.97	2096.02
7	Equipment after	11.16	-50.50	-563.57
8	Machinery space	266.37	-39.03	-10395.14
9	Propeller dan poros	10.57	-48.825	-516.27
$\Sigma_1 =$		<b>1720.37</b>	$\Sigma_2 =$	<b>-13685.88</b>

Berat kapal yang akan diluncurkan :

$$P = \Sigma_1 = \mathbf{1720.37 \text{ ton}}$$

Titik berat kapal yang diluncurkan :

$$\begin{aligned} LCG &= \Sigma_2 / \Sigma_1 = -13685.88 / 1720.37 \\ LCG &= \mathbf{-7.96 \text{ m}} \quad (\text{di belakang midship}) \end{aligned}$$



## Rekapitulasi perhitungan Periode II :

No	Notasi	Satuan			
			1	2	3
1	$\Sigma_1$	m <sup>2</sup>	0.40	5.56	31.76
2	$\Sigma_2$	m <sup>2</sup>	0.40	12.72	109.52
3	hs = Lpp / 20	m	5.05	5.05	5.05
4	So = 2 x n x hs	m	10.10	20.20	30.30
5	$z = hs \tan \alpha$	m	8.80	8.80	8.80
6	Sn = So + z	m	18.90	29.00	39.10
7	Ta = So x tg $\alpha$	m	0.46	0.92	1.38
8	$V' = 1/3 x hs x \Sigma_1$	m <sup>3</sup>	0.67	9.36	53.46
9	$\gamma V' = 1.004 x 1.025 x V'$	ton	0.69	9.63	55.02
10	$LCB' = \Sigma_2 / \Sigma_1 x hs$	m	5.05	11.55	17.41
11	f = LCB'	m	5.05	11.55	17.41
12	S =	m	90.90	90.90	90.90
13	$h = (Lpp - S) / 2$	m	5.05	5.05	5.05
14	$g = Lpp/2 + LCGp$	m	43.26	43.26	43.26
15	$\lambda = Lpp / 3$	m	33.67	33.67	33.67
16	$d = (S + h) - f$	m	90.90	84.40	78.54
17	$c = (S + h) - g$	m	52.69	52.69	52.69
18	P	ton	1892.41	1892.41	1892.41
19	$\gamma V' x d$	ton m	62.99	812.88	4320.91
20	P x c	ton m	99704.04	99704.04	99704.04
21	(P x c) - ( $\gamma V' x d$ )	ton m	99641.06	98891.16	95383.13
22	a = (Sn - $\lambda$ ) - g	m	-58.03	-47.93	-37.83
23	b = (Sn - $\lambda$ ) - f	m	-19.82	-16.22	-11.98
24	P x a	ton m	-109817.58	-90704.27	-71590.96
25	$\gamma V' x b$	ton m	-13.73	-156.23	-659.19
26	Q = P - $\gamma V'$	ton	1891.71	1882.78	1837.39
27	$x' = ((\gamma V' x b) - (P x a)) / Q$	m	58.04	48.09	38.60
28	$S' = S$	m	90.90	90.90	
	$S' = (S + h) - f - b$	m			90.52
29	$1/2 x S'$	m	45.45	45.45	45.26
30	$1/3 x S'$	m	30.30	30.30	30.17
31	$2/3 x S'$	m	60.60	60.60	60.34
32	$1/3 x S' < x' < 2/3 x S'$		Trapesium	Trapesium	Trapesium
	$x' < 1/3 x S'$ or $x' > 2/3 x S'$				
33	Jika Trapesium :		38.23	38.38	
	$q = Q/S'$	ton/m	20.81	20.71	20.30
	$qa = 2q (2S' - 3x') / S'$	ton/m	30.73	30.38	29.25
	$qf = 2q (3x' - S') / S'$	ton/m	10.89	11.04	11.35
34	Jika Segitiga :				
	$q = Q/3x'$	ton/m			
	$qa = 2 x q$	ton/m			
	$qf = 0$	ton/m			



4	5	6	7	8	9
98.88	216.16	382.96	599.24	858.96	1147.96
434.08	1131.60	2292.00	3999.04	6255.36	8927.12
5.05	5.05	5.05	5.05	5.05	5.05
40.40	50.50	60.60	70.70	80.80	90.90
8.80	8.80	8.80	8.80	8.80	8.80
49.20	59.30	69.40	79.50	89.60	99.70
1.84	2.30	2.75	3.21	3.67	4.13
166.45	363.87	644.65	1008.72	1445.92	1932.40
171.29	374.46	663.41	1038.07	1487.99	1988.63
22.17	26.44	30.22	33.70	36.78	39.27
22.17	26.44	30.22	33.70	36.78	39.27
90.90	90.90	90.90	90.90	90.90	90.90
5.05	5.05	5.05	5.05	5.05	5.05
43.26	43.26	43.26	43.26	43.26	43.26
33.67	33.67	33.67	33.67	33.67	33.67
73.78	69.51	65.73	62.25	59.17	56.68
52.69	52.69	52.69	52.69	52.69	52.69
1892.41	1892.41	1892.41	1892.41	1892.41	1892.41
12638.01	26029.77	43603.17	64618.81	88049.66	112712.96
99704.04	99704.04	99704.04	99704.04	99704.04	99704.04
87066.03	73674.28	56100.88	35085.23	11654.38	-13008.91
-27.73	-17.63	-7.53	2.57	12.67	22.77
-6.64	-0.80	5.51	12.13	19.16	26.76
-52477.65	-33364.34	-14251.03	4862.28	23975.59	43088.90
-1136.75	-300.99	3654.69	12593.63	28504.68	53219.05
1721.12	1517.95	1229.00	854.33	404.41	-96.23
29.83	21.78	14.57	9.05	11.20	-105.28
80.42	70.32	60.22	50.12	40.02	29.92
40.21	35.16	30.11	25.06	20.01	14.96
26.81	23.44	20.07	16.71	13.34	9.97
53.61	46.88	40.14	33.41	26.68	19.94
Trapesium					
	Segitiga	Segitiga	Segitiga	Segitiga	Segitiga
21.40					
37.98					
4.83					
	23.23	28.12	31.47	12.04	0.30
	46.46	56.24	62.94	24.07	0.61
	0.00	0.00	0.00	0.00	0.00

Tabel 2.2. Rekapitulasi perhitungan Periode II



Rekapitulasi perhitungan Periode III :

No	Notasi	Satuan	Langkah 1		
			1a	1b	1c
1	$\Sigma_1$		609.73	802.47	1001.40
2	$\Sigma_2$		4710.46	6126.26	7562.56
3	Ta	m	2.50	3.20	3.90
4	$V' = 1/3 \times hstat \times \Sigma_1$	$m^3$	1026.38	1350.82	1685.69
5	$\gamma V' = 1.004 \times 1.025 \times V'$	ton	1056.25	1390.13	1734.74
6	$f = \Sigma_2/\Sigma_1 \times hstat$	m	39.01	38.55	38.14
7	$d = (S + h) - f$	m	56.94	57.40	57.81
8	$c = (S + h) - g$	m	52.69	52.69	52.69
9	P	ton	1892.41	1892.41	1892.41
10	P x c	ton m	99704.04	99704.04	99704.04
11	$\gamma V' \times d$	ton m	60138.77	79789.51	100289.80

No	Notasi	Satuan	Langkah 4		
			4a	4b	4c
1	$\Sigma_1$		929.76	1270.40	1456.84
2	$\Sigma_2$		7791.04	10429.92	11833.68
3	Ta	m	3.00	4.05	4.59
4	$V' = 1/3 \times hstat \times \Sigma_1$	$m^3$	1565.10	2138.51	2452.35
5	$\gamma V' = 1.004 \times 1.025 \times V'$	ton	1610.64	2200.74	2523.71
6	$f = \Sigma_2/\Sigma_1 \times hstat$	m	42.32	41.46	41.02
7	$d = (S + h) - f$	m	53.63	54.49	54.93
8	$c = (S + h) - g$	m	52.69	52.69	52.69
9	P	ton	1892.41	1892.41	1892.41
10	P x c	ton m	99704.04	99704.04	99704.04
11	$\gamma V' \times d$	ton m	86383.30	119917.63	138626.55



Langkah 2			Langkah 3		
2a	2b	2c	3a	3b	3c
784.36	963.24	1147.96	855.70	1123.46	1299.90
6135.36	7517.04	8927.12	6943.92	9016.12	10362.46
3.00	3.57	4.13	3.00	3.83	4.36
1320.34	1621.45	1932.40	1440.43	1891.16	2188.17
1358.76	1668.64	1988.63	1482.34	1946.19	2251.84
39.50	39.41	39.27	40.98	40.53	40.26
56.45	56.54	56.68	54.97	55.42	55.69
52.69	52.69	52.69	52.69	52.69	52.69
1892.41	1892.41	1892.41	1892.41	1892.41	1892.41
99704.04	99704.04	99704.04	99704.04	99704.04	99704.04
76699.74	94345.22	112712.96	81484.13	107862.08	125411.15

Langkah 5			Langkah 6		
5a	5b	5c	6a	6b	6c
843.52	1347.44	1617.00	920.44	1322.60	1781.56
7377.36	11316.96	13336.40	8269.60	11495.04	14894.00
2.50	4.05	4.82	2.50	3.78	5.05
1419.93	2268.19	2721.95	1549.41	2226.38	2998.96
1461.25	2334.20	2801.16	1594.50	2291.16	3086.23
44.17	42.41	41.65	45.37	43.89	42.22
51.78	53.54	54.30	50.58	52.06	53.73
52.69	52.69	52.69	52.69	52.69	52.69
1892.41	1892.41	1892.41	1892.41	1892.41	1892.41
99704.04	99704.04	99704.04	99704.04	99704.04	99704.04
75667.80	124962.90	152101.59	80647.63	119276.21	165827.87

Tabel 2.3. Rekapitulasi perhitungan Periode III



## LAMPIRAN III

### LISTING PROGRAM

Listing program untuk frmSplash.frm :

Option Explicit

```
Private Sub Command1_Click()
Unload Me
frmMain.Show
End Sub
```

```
Private Sub Form_Activate()
Frame1.Top = frmSplash.Height / 2 - Frame1.Height / 2
Frame1.Left = frmSplash.Width / 2 - Frame1.Width / 2
End Sub
```

Listing program untuk modAwal.bas :

Option Explicit

```
Private Sub main()
'display splash screen
frmSplash.Show vbModeless
End Sub
```

Listing program untuk frmMain.frm :

Option Explicit

```
Dim strDBName As String
Dim strRSName As String
Dim cnnDataBonjean As New ADODB.Connection
Dim rsTable As ADODB.Recordset
Dim jumField As Integer
Dim jumRecord As Integer
Dim WL(100) As Double
Dim stat(100, 100) As Double
Dim ordeB(100, 100, 100) As Double
Dim q(100) As Double
```



Dim qA(100), qF(100) As Double  
Dim X(100) As Double  
Dim Ta(100) As Double  
Dim So(100) As Double  
Dim z As Double  
Dim Sn(100) As Double  
Dim Vaks(100) As Double  
Dim gVaks(100) As Double  
Dim f(100) As Double  
Dim d(100) As Double  
Dim gVaks\_d(100) As Double  
Dim A(100) As Double  
Dim b(100) As Double  
Dim Ptotal\_a(100) As Double  
Dim gVaks\_b(100) As Double  
Dim Qbesar(100) As Double  
Dim Saks(100) As Double  
Dim Saks23(100) As Double  
Dim Saks12(100) As Double  
Dim Saks13(100) As Double  
Dim xAks(100) As Double

Dim Lpp As Double  
Dim hStat As Double  
Dim Ptotal As Double  
Dim LCGtotal As Double  
Dim BsepatuMin As Double  
Dim sudutA As Double  
Dim S As Double  
Dim hSpt, bSpt, Aspt, nSpt As Double  
Dim Lamda As Double  
Dim h As Double  
Dim g As Double  
Dim c As Double  
Dim jSpt As Double  
Dim mUs, mUd As Double  
Dim Ptotal\_c As Double  
Const Pi = 3.14159265358979  
Const Grav = 9.80665  
Const EXP = 2.718281828  
Dim strPerhitungan As String  
Dim awalP3, awalP4 As Integer

Dim gV3d(100, 3) As Double  
Dim Ta3(100, 3) As Double  
Dim gV3(100, 3) As Double



```
Dim grafA(100) As Double
Dim grafS(100) As Double
Dim grafV(100) As Double
Dim grafT(100) As Double
Dim strDinamis As String
Dim ordeQ(100, 100) As Double
Dim V1g As Double
Dim gLuas(100) As Double
Dim ordeGV(100, 100) As Double

Dim acadApp As AcadApplication
Dim acadDoc As AcadDocument
Dim layerObj As AcadLayer
Dim plineObj As AcadPolyline
Dim splineObj As AcadSpline

Dim strDwg As String

Private Sub cmdBack1_Click()
With SSTab1
    .TabEnabled(1) = False
    .TabEnabled(0) = True
    .Tab = 0
End With
End Sub

Private Sub cmdBack2_Click()
With SSTab1
    .TabEnabled(2) = False
    .TabEnabled(1) = True
    .Tab = 1
End With
End Sub

Private Sub cmdBack3_Click()
With SSTab1
    .TabEnabled(3) = False
    .TabEnabled(2) = True
    .Tab = 2
End With
End Sub

Private Sub cmdBack4_Click()
With SSTab1
    .TabEnabled(4) = False

```



```
.TabEnabled(3) = True
.Tab = 3
End With
End Sub

Private Sub cmdNext0_Click()

hitungDataKapal
With SSTab1
    .TabEnabled(0) = False
    .TabEnabled(1) = True
    .Tab = 1
End With

lblNamaBonjean.Caption = strRSName
tampilBonjean

S = Val(txtS.Text)
Lpp = Val(txtLpp.Text)
hStat = Lpp / (jumField - 2)
hSpt = Val(txtHsepatu.Text)
bSpt = Val(txtBsepatu.Text)
nSpt = Val(txtNsepatu.Text)
Aspt = hSpt * bSpt * nSpt
Lamda = Val(txtPanjBwhAir.Text)
h = (Lpp - S) / 2
g = Lpp / 2 + LCGtotal
c = (S + h) - g
Ptotal_c = Ptotal * c
z = hSpt / Tan(sudutA * Pi / 180)
jSpt = Val(txtJarakSpt.Text)
mUs = Val(txtMuS.Text)
mUd = Val(txtMuD.Text)

cmdNext1.SetFocus

End Sub

Private Sub cmdNext1_Click()

Dim j, k As Integer
Dim strTemp As String
Dim strTemp2(100) As String

With SSTab1
    .TabEnabled(1) = False
```



```
.TabEnabled(2) = True
.Tab = 2
End With

hitungPers

strTemp2(0) = ""

For j = 1 To jumRecord - 1 Step 1
    strTemp2(j) = strTemp2(j - 1) & "(x - " & WL(j - 1) & ")"
Next j

For k = 0 To 20 Step 1
    strTemp = strTemp & "Pers. station " & k & " = " & vbTab
    strTemp = strTemp & ordeB(k, 0, 0) & "+"
    For j = 1 To jumRecord - 1 Step 1
        strTemp = strTemp & strTemp2(j) & "(" & ordeB(k, j, 1) & ")" + "
    Next j
    strTemp = strTemp & vbCrLf
Next k

txtPers.Text = strTemp
cmdNext2.SetFocus

End Sub

Private Sub cmdNext2_Click()

With SSTab1
    .TabEnabled(2) = False
    .TabEnabled(3) = True
    .Tab = 3
End With

PERIODE1
PERIODE2
PER3

txtPerhitungan.Text = ""
txtPerhitungan.Text = strPerhitungan

End Sub

Private Sub cmdNext3_Click()

With SSTab1
```



```
.TabEnabled(3) = False  
.TabEnabled(4) = True  
.Tab = 4
```

```
End With
```

```
Gerak1  
iterNolQ  
iterSatuQ  
iterDuaQ  
Gerak2
```

```
txtDinamis.Text = strDinamis
```

```
End Sub
```

```
Private Sub cmdNext4_Click()
```

```
yDrawing  
OpenDrawing  
cmdView.Enabled = True  
cmdNext4.Enabled = False  
Dim i As Integer
```

```
'Saks
```

```
Set layerObj = acadDoc.Layers.Add("Saks")  
acadDoc.ActiveLayer = layerObj  
layerObj.Color = acBlue  
Dim pointsSaks(0 To 8) As Double  
pointsSaks(0) = 0 * 5: pointsSaks(1) = S * 2: pointsSaks(2) = 0  
pointsSaks(3) = -(Lamda - z + h) * 5: pointsSaks(4) = S * 2: pointsSaks(5) = 0  
pointsSaks(6) = -(Lamda - z + h + S) * 5: pointsSaks(7) = 0 * 2: pointsSaks(8) = 0  
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsSaks)  
Dim pointsSaks1(0 To 8) As Double  
pointsSaks1(0) = 0 * 5: pointsSaks1(1) = (S / 3) * 2: pointsSaks1(2) = 0  
pointsSaks1(3) = -(Lamda - z + h) * 5: pointsSaks1(4) = (S / 3) * 2:  
pointsSaks1(5) = 0  
pointsSaks1(6) = -(Lamda - z + h + S) * 5: pointsSaks1(7) = 0 * 2: pointsSaks1(8)  
= 0
```

```
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsSaks1)
```

```
Dim pointsSaks2(0 To 8) As Double
```

```
pointsSaks2(0) = 0 * 5: pointsSaks2(1) = (S / 2) * 2: pointsSaks2(2) = 0  
pointsSaks2(3) = -(Lamda - z + h) * 5: pointsSaks2(4) = (S / 2) * 2:  
pointsSaks2(5) = 0  
pointsSaks2(6) = -(Lamda - z + h + S) * 5: pointsSaks2(7) = 0 * 2: pointsSaks2(8)  
= 0
```

```
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsSaks2)
```

```
Dim pointsSaks3(0 To 8) As Double
```

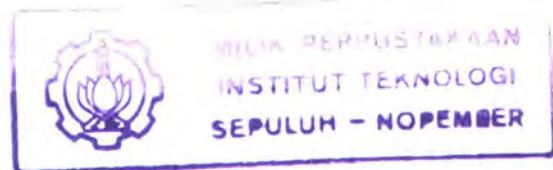


```
pointsSaks3(0) = 0 * 5: pointsSaks3(1) = (S * 2 / 3) * 2: pointsSaks3(2) = 0
pointsSaks3(3) = -(Lamda - z + h) * 5: pointsSaks3(4) = (S * 2 / 3) * 2:
pointsSaks3(5) = 0
pointsSaks3(6) = -(Lamda - z + h + S) * 5: pointsSaks3(7) = 0 * 2: pointsSaks3(8)
= 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsSaks3)

'Ptotal
Set layerObj = acadDoc.Layers.Add("Ptot")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acRed
Dim pointsP(0 To 5) As Double
pointsP(0) = 0 * 5: pointsP(1) = Ptotal / 5: pointsP(2) = 0
pointsP(3) = -(Lamda - z + h + S) * 5: pointsP(4) = Ptotal / 5: pointsP(5) = 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsP)

'Px c
Set layerObj = acadDoc.Layers.Add("Pxc")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acYellow
Dim pointsPxc(0 To 5) As Double
pointsPxc(0) = 0 * 5: pointsPxc(1) = Ptotal_c / 400: pointsPxc(2) = 0
pointsPxc(3) = -(Lamda - z + h + S) * 5: pointsPxc(4) = Ptotal_c / 400:
pointsPxc(5) = 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsPxc)

'gV x d
Set layerObj = acadDoc.Layers.Add("gVxd")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acYellow
Dim b0, b1, b2, gVdSt, gVst As Double
b0 = So(awalP3 - 2)
b1 = (So(awalP3 - 1) - So(awalP3 - 2)) _
/ (gVaks_d(awalP3 - 1) - gVaks_d(awalP3 - 2))
b2 = (So(awalP3) - So(awalP3 - 1)) _
/ (gVaks_d(awalP3) - gVaks_d(awalP3 - 1))
b2 = (b2 - b1) / (gVaks_d(awalP3) - gVaks_d(awalP3 - 2))
gVdSt = b0 + b1 * (Ptotal_c - gVaks_d(awalP3 - 2)) _
- b2 * (Ptotal_c - gVaks_d(awalP3 - 2)) * (Ptotal_c - gVaks_d(awalP3 - 1))
Dim pointsGVd() As Double
Dim startGV(0 To 2) As Double
Dim endGV(0 To 2) As Double
ReDim pointsGVd(0 To awalP3 * 3 + 2) As Double
For i = 0 To awalP3 - 1
    pointsGVd(i * 3) = -(So(i) * 5): pointsGVd(i * 3 + 1) = gVaks_d(i) / 400:
    pointsGVd(i * 3 + 2) = 0
```





```
Next i
pointsGVd(awalP3 * 3) = -(gVdSt * 5): pointsGVd(awalP3 * 3 + 1) = Ptotal_c / 400: pointsGVd(awalP3 * 3 + 2) = 0
Set splineObj = acadDoc.ModelSpace.AddSpline(pointsGVd, startGV, endGV)

'stern lift
Set layerObj = acadDoc.Layers.Add("SternLift")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acWhite
Dim pointsLift(0 To 5) As Double
pointsLift(0) = -(gVdSt * 5): pointsLift(1) = 0: pointsLift(2) = 0
pointsLift(3) = -(gVdSt * 5): pointsLift(4) = Ptotal / 5: pointsLift(5) = 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsLift)

'gV
Set layerObj = acadDoc.Layers.Add("GV")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acRed
b0 = gVaks(awalP3 - 2)
b1 = (gVaks(awalP3 - 1) - gVaks(awalP3 - 2))_ / (So(awalP3 - 1) - So(awalP3 - 2))
b2 = (gVaks(awalP3) - gVaks(awalP3 - 1))_ / (So(awalP3) - So(awalP3 - 1))
b2 = (b2 - b1) / (So(awalP3) - So(awalP3 - 2))
gVSt = b0 + b1 * (gVdSt - So(awalP3 - 2))_ - b2 * (gVdSt - So(awalP3 - 2)) * (gVdSt - So(awalP3 - 1))
'linear
'gVSt = (gVdSt - So(awalP3 - 2)) / (So(awalP3 - 1) - So(awalP3 - 2))
'gVSt = gVSt * (gVaks(awalP3 - 1) - gVaks(awalP3 - 2)) + gVaks(awalP3 - 2)
Dim pointsGV() As Double
ReDim pointsGV(0 To awalP3 * 3 + 2) As Double
For i = 0 To awalP3 - 1
    pointsGV(i * 3) = -(So(i) * 5): pointsGV(i * 3 + 1) = gVaks(i) / 5: pointsGV(i * 3 + 2) = 0
Next i
pointsGV(awalP3 * 3) = -(gVdSt * 5): pointsGV(awalP3 * 3 + 1) = gVSt / 5: pointsGV(awalP3 * 3 + 2) = 0
Set splineObj = acadDoc.ModelSpace.AddSpline(pointsGV, startGV, endGV)

Dim pointsGV2() As Double
Dim gVst2 As Double
b0 = So(awalP4 - 2)
b1 = (So(awalP4 - 1) - So(awalP4 - 2))_ / (gVaks(awalP4 - 1) - gVaks(awalP4 - 2))
b2 = (So(awalP4) - So(awalP4 - 1))_ / (gVaks(awalP4) - gVaks(awalP4 - 1))
```



```
b2 = (b2 - b1) / (gVaks(awalP4) - gVaks(awalP4 - 2))
gVst2 = b0 + b1 * (Ptotal - gVaks(awalP4 - 2)) -
         - b2 * (Ptotal - gVaks(awalP4 - 2)) * (Ptotal - gVaks(awalP4 - 1))
ReDim pointsGV2(0 To (awalP4 - awalP3 + 1) * 3 + 2) As Double
pointsGV2(0) = -(gVdSt * 5): pointsGV2(1) = gVSt / 5: pointsGV2(2) = 0
For i = 1 To (awalP4 - awalP3)
    pointsGV2(i * 3) = -(So(i + awalP3 - 1) * 5): pointsGV2(i * 3 + 1) = gVaks(i +
        awalP3 - 1) / 5: pointsGV2(i * 3 + 2) = 0
    Next i
    pointsGV2((awalP4 - awalP3 + 1) * 3) = -(gVst2 * 5): pointsGV2((awalP4 -
        awalP3 + 1) * 3 + 1) = Ptotal / 5: pointsGV2((awalP4 - awalP3 + 1) * 3 + 2) = 0
    Set splineObj = acadDoc.ModelSpace.AddSpline(pointsGV2, startGV, endGV)
```

```
'P x a
Set layerObj = acadDoc.Layers.Add("Pxa")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acYellow
Dim yPa As Double
yPa = ((Lamda - z + h + S) - So(1)) * (-Ptotal_a(1))
yPa = yPa / (Ptotal_c - Ptotal_a(1))
yPa = yPa + So(1)
Dim pointsPxa(0 To 5) As Double
pointsPxa(0) = -yPa * 5: pointsPxa(1) = 0 / 400: pointsPxa(2) = 0
pointsPxa(3) = -(Lamda - z + h + S) * 5: pointsPxa(4) = Ptotal_c / 400:
pointsPxa(5) = 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsPxa)
```

```
'gV x b
Set layerObj = acadDoc.Layers.Add("gVxb")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acYellow
Dim pointsGVb() As Double
ReDim pointsGVb(0 To awalP3 * 3 + 2) As Double
For i = 0 To awalP3
    pointsGVb(i * 3) = -(So(i) * 5): pointsGVb(i * 3 + 1) = gVaks_b(i) / 400:
    pointsGVb(i * 3 + 2) = 0
    Next i
    Set splineObj = acadDoc.ModelSpace.AddSpline(pointsGVb, startGV, endGV)
```

```
'Ta
Set layerObj = acadDoc.Layers.Add("Ta")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acGreen
Dim yTa As Double
yTa = (gVdSt - So(1)) * (Ta(2) - Ta(1))
yTa = yTa / (So(2) - So(1))
```



```
yTa = yTa + Ta(1)
Dim pointsTa(0 To 5) As Double
pointsTa(0) = -0 * 5: pointsTa(1) = 0 * 50: pointsTa(2) = 0
pointsTa(3) = -(gVdSt * 5): pointsTa(4) = yTa * 50: pointsTa(5) = 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsTa)
Dim pointsTa2(0 To 11) As Double
pointsTa2(0) = -(gVdSt * 5): pointsTa2(1) = yTa * 50: pointsTa2(2) = 0
pointsTa2(3) = -(So(10) * 5): pointsTa2(4) = Ta(10) * 50: pointsTa2(5) = 0
pointsTa2(6) = -(So(11) * 5): pointsTa2(7) = Ta(11) * 50: pointsTa2(8) = 0
pointsTa2(9) = -(So(12) * 5): pointsTa2(10) = Ta(12) * 50: pointsTa2(11) = 0
Set plineObj = acadDoc.ModelSpace.AddPolyline(pointsTa2)

'q
Set layerObj = acadDoc.Layers.Add("q")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acMagenta
Dim qSkala As Integer
qSkala = 2
Dim gQ As Double
b0 = q(awalP4 - 2)
b1 = (q(awalP4 - 1) - q(awalP4 - 2))_
 / (So(awalP4 - 1) - So(awalP4 - 2))
b2 = (q(awalP4) - q(awalP4 - 1))_
 / (So(awalP4) - So(awalP4 - 1))
b2 = (b2 - b1) / (So(awalP4) - So(awalP4 - 2))
gQ = b0 + b1 * (gVdSt - So(awalP4 - 2))_
 - b2 * (gVdSt - So(awalP4 - 2)) * (gVdSt - So(awalP4 - 1))
Dim pointsq() As Double
ReDim pointsq(0 To awalP3 * 3 + 2) As Double
For i = 0 To awalP3 - 1
    pointsq(i * 3) = -(So(i) * 5): pointsq(i * 3 + 1) = q(i) * qSkala: pointsq(i * 3 +
2) = 0
Next i
pointsq(awalP3 * 3) = -(gVdSt * 5): pointsq(awalP3 * 3 + 1) = gQ * qSkala:
pointsq(awalP3 * 3 + 2) = 0
Set splineObj = acadDoc.ModelSpace.AddSpline(pointsq, startGV, endGV)

Dim pointsq2(0 To 11) As Double
pointsq2(0) = -(gVdSt * 5): pointsq2(1) = q(9) * qSkala: pointsq2(2) = 0
pointsq2(3) = -(So(10) * 5): pointsq2(4) = q(10) * qSkala: pointsq2(5) = 0
pointsq2(6) = -(So(11) * 5): pointsq2(7) = q(11) * qSkala: pointsq2(8) = 0
pointsq2(9) = -(gVst2 * 5): pointsq2(10) = 0 * qSkala: pointsq2(11) = 0
Set splineObj = acadDoc.ModelSpace.AddSpline(pointsq2, startGV, endGV)

'Dim pointsq2() As Double
'ReDim pointsq2(0 To (awalP4 - awalP3 + 2) * 3 + 2) As Double
```



```
'pointsq2(0) = -(gVdSt * 5): pointsq2(1) = gQ * qSkala: pointsq2(2) = 0
'For i = 0 To (awalP4 - awalP3 - 2)
    pointsq2(i * 3) = -(So(i + awalP3) * 5): pointsq2(i * 3 + 1) = q(i + awalP3) *
    qSkala: pointsq2(i * 3 + 2) = 0
'Next i
'pointsq2((awalP4 - awalP3) * 3) = -(gVst2 * 5): pointsq2((awalP4 - awalP3) * 3
+ 1) = 0 * qSkala: pointsq2((awalP4 - awalP3) * 3 + 2) = 0
'Set splineObj = acadDoc.ModelSpace.AddSpline(pointsq2, startGV, endGV)

'qa
Set layerObj = acadDoc.Layers.Add("qa")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acMagenta
Dim gQa As Double
b0 = q(awalP3 - 2)
b1 = (q(awalP3 - 1) - q(awalP3 - 2)) _
    / (So(awalP3 - 1) - So(awalP3 - 2))
b2 = (q(awalP3) - q(awalP3 - 1)) _
    / (So(awalP3) - So(awalP3 - 1))
b2 = (b2 - b1) / (So(awalP3) - So(awalP3 - 2))
gQa = b0 + b1 * (gVdSt - So(awalP3 - 2)) _
    - b2 * (gVdSt - So(awalP3 - 2)) * (gVdSt - So(awalP3 - 1))
Dim pointsqA(0 To 29) As Double
pointsqA(0) = 0 * 5: pointsqA(1) = qA(0) * qSkala: pointsqA(2) = 0
pointsqA(3) = -(So(1) * 5): pointsqA(4) = qA(1) * qSkala: pointsqA(5) = 0
pointsqA(6) = -(So(2) * 5): pointsqA(7) = qA(2) * qSkala: pointsqA(8) = 0
pointsqA(9) = -(So(3) * 5): pointsqA(10) = qA(3) * qSkala: pointsqA(11) = 0
pointsqA(12) = -(So(4) * 5): pointsqA(13) = qA(4) * qSkala: pointsqA(14) = 0
pointsqA(15) = -(So(5) * 5): pointsqA(16) = qA(5) * qSkala: pointsqA(17) = 0
pointsqA(18) = -(So(6) * 5): pointsqA(19) = qA(6) * qSkala: pointsqA(20) = 0
pointsqA(21) = -(So(7) * 5): pointsqA(22) = qA(7) * qSkala: pointsqA(23) = 0
pointsqA(24) = -(So(8) * 5): pointsqA(25) = qA(8) * qSkala: pointsqA(26) = 0
pointsqA(27) = -(gVdSt * 5): pointsqA(28) = qA(9) * qSkala: pointsqA(29) = 0
Set splineObj = acadDoc.ModelSpace.AddSpline(pointsqA, startGV, endGV)

'qf
Set layerObj = acadDoc.Layers.Add("qf")
acadDoc.ActiveLayer = layerObj
layerObj.Color = acMagenta
Dim gQf As Double
b0 = q(awalP3 - 2)
b1 = (q(awalP3 - 1) - q(awalP3 - 2)) _
    / (So(awalP3 - 1) - So(awalP3 - 2))
b2 = (q(awalP3) - q(awalP3 - 1)) _
    / (So(awalP3) - So(awalP3 - 1))
b2 = (b2 - b1) / (So(awalP3) - So(awalP3 - 2))
```



```
gQf = b0 + b1 * (gVdSt - So(awalP3 - 2))  
    - b2 * (gVdSt - So(awalP3 - 2)) * (gVdSt - So(awalP3 - 1))  
Dim pointsqF(0 To 29) As Double  
pointsqF(0) = 0 * 5: pointsqF(1) = qF(0) * qSkala: pointsqF(2) = 0  
pointsqF(3) = -(So(1) * 5): pointsqF(4) = qF(1) * qSkala: pointsqF(5) = 0  
pointsqF(6) = -(So(2) * 5): pointsqF(7) = qF(2) * qSkala: pointsqF(8) = 0  
pointsqF(9) = -(So(3) * 5): pointsqF(10) = qF(3) * qSkala: pointsqF(11) = 0  
pointsqF(12) = -(So(4) * 5): pointsqF(13) = qF(4) * qSkala: pointsqF(14) = 0  
pointsqF(15) = -(So(5) * 5): pointsqF(16) = qF(5) * qSkala: pointsqF(17) = 0  
pointsqF(18) = -(So(6) * 5): pointsqF(19) = qF(6) * qSkala: pointsqF(20) = 0  
pointsqF(21) = -(So(7) * 5): pointsqF(22) = qF(7) * qSkala: pointsqF(23) = 0  
pointsqF(24) = -(So(8) * 5): pointsqF(25) = qF(8) * qSkala: pointsqF(26) = 0  
pointsqF(27) = -(gVdSt * 5): pointsqF(28) = qF(9) * qSkala: pointsqF(29) = 0  
Set splineObj = acadDoc.ModelSpace.AddSpline(pointsqF, startGV, endGV)
```

End Sub

```
Private Sub cmdSaveAs_Click()  
strDwg = InputBox("Drawing save as ...", "Save as")  
acadDoc.SaveAs strDwg & ".dwg"
```

End Sub

```
Private Sub cmdView_Click()  
yDrawing  
cmdSaveAs.Enabled = True  
acadApp.Visible = True  
cmdView.Enabled = False
```

End Sub

```
Private Sub comboPalat_Change()  
hitPalat  
End Sub
```

```
Private Sub comboPalat_Click()  
hitPalat  
End Sub
```

```
Private Sub Form_Activate()
```

Dim iRate As Single

```
Set cnnDataBonjean = Nothing  
Set rsTable = Nothing
```



With SSTab1

```
.TabEnabled(1) = False  
.TabEnabled(2) = False  
.TabEnabled(3) = False  
.TabEnabled(4) = False  
.Tab = 0
```

End With

cmdNext0.Enabled = False

comboPalat.Text = 10

For iRate = 7 To 16 Step 0.5

```
    comboPalat.AddItem Format(iRate, "0.0")
```

Next iRate

convSudut

End Sub

Private Sub Form\_Load()

Dim iRate As Single

Set cnnDataBonjean = Nothing

Set rsTable = Nothing

With SSTab1

```
.TabEnabled(1) = False  
.TabEnabled(2) = False  
.TabEnabled(3) = False  
.TabEnabled(4) = False  
.Tab = 0
```

End With

cmdNext0.Enabled = False

comboPalat.Text = 10

For iRate = 7 To 16 Step 0.5

```
    comboPalat.AddItem Format(iRate, "0.0")
```

Next iRate

convSudut

End Sub

Private Sub Form\_Unload(Cancel As Integer)

Set cnnDataBonjean = Nothing

Set rsTable = Nothing

End Sub



```
Private Sub mnuFileExit_Click()
Set cnnDataBonjean = Nothing
Set rsTable = Nothing
Unload Me
End Sub

Private Sub tampilBonjean()
adoBonjean.ConnectionString = "Provider=Microsoft.Jet.OLEDB.4.0;" & _
    "Data Source=" & strDBName & ";Persist Security Info=False"
adoBonjean.CommandType = adCmdTable
adoBonjean.RecordSource = strRSName
adoBonjean.Refresh
dgdBonjean.ReBind
End Sub

Private Sub OpenDatabase()
Set cnnDataBonjean = Nothing
Set rsTable = Nothing

cdlgMain.Filter = "Database Files (.mdb)|*.mdb"
cdlgMain.ShowOpen

On Error GoTo errorHandler
If cdlgMain.FileName <> "" Then
    strDBName = cdlgMain.FileName
    cnnDataBonjean.Open "Provider=Microsoft.Jet.OLEDB.4.0;" & _
        "Data Source=" & strDBName
    strRSName = txtNamaKapal.Text
    Set rsTable = New ADODB.Recordset
    rsTable.CursorType = adOpenKeyset
    rsTable.LockType = adLockOptimistic
    rsTable.Open strRSName, cnnDataBonjean, , , adCmdTable
End If
recordDatabase

Exit Sub

errorHandler:
Select Case Err.Number
    Case 3705
        MsgBox ("Nama kapal tidak terdapat pada database!") _
            , vbCritical, "Error Message"
    Case -2147217865
        MsgBox ("Nama kapal tidak terdapat pada database!") _
            , vbCritical, "Error Message"
```



Case Else

```
    MsgBox ("Error pada database!") _  
        , vbCritical, "Error Message"
```

End Select

End Sub

Private Sub mnuFileOpen\_Click()

OpenDatabase

```
If strDBName <> "" Then  
    cmdNext0.Enabled = True  
    cmdNext0.SetFocus  
End If
```

End Sub

Private Sub recordDatabase()

Dim i As Integer

Dim j As Integer

i = 0

```
jumField = rsTable.Fields.Count  
jumRecord = rsTable.RecordCount
```

rsTable.MoveFirst

Do Until rsTable.EOF

```
    WL(i) = CDbl(rsTable.Fields(0))
```

i = i + 1

rsTable.MoveNext

Loop

i = 0

rsTable.MoveFirst

Do Until rsTable.EOF

```
    For j = 1 To jumField - 1 Step 1
```

```
        stat(j - 1, i) = CDbl(rsTable.Fields(j))
```

Next j

i = i + 1

rsTable.MoveNext

Loop

End Sub

Private Sub iterNol()



Dim i, k As Integer

```
For k = 0 To jumField - 2 Step 1
    For i = 0 To jumRecord - 1 Step 1
        ordeB(k, 0, i) = stat(k, i)
    Next i
Next k
```

End Sub

Private Sub iterSatu()

Dim i, k As Integer

```
For k = 0 To jumField - 2 Step 1
    For i = 1 To jumRecord - 1 Step 1
        ordeB(k, 1, i) = stat(k, i) - stat(k, i - 1)
        ordeB(k, 1, i) = ordeB(k, 1, i) / (WL(i) - WL(i - 1))
    Next i
Next k
```

End Sub

Private Sub iterDua()

Dim i, j, k As Integer

```
For k = 0 To jumField - 2 Step 1
    For j = 2 To jumRecord - 1 Step 1
        For i = 1 To jumRecord - j Step 1
            ordeB(k, j, i) = ordeB(k, j - 1, i + 1) - ordeB(k, j - 1, i)
            ordeB(k, j, i) = ordeB(k, j, i) / (WL(i + j - 1) - WL(i - 1))
        Next i
    Next j
Next k
```

End Sub

Private Sub hitungPers()

iterNol  
iterSatu  
iterDua

End Sub



Private Function hitPersBonjean(ST As Integer, X As Double) As Double

Dim j As Integer

Dim dblTemp As Double

Dim dblTemp2(100) As Double

dblTemp2(0) = 1

For j = 1 To jumRecord - 1 Step 1

    dblTemp2(j) = dblTemp2(j - 1) \* (X - WL(j - 1))

Next j

dblTemp = ordeB(ST, 0, 0)

For j = 1 To jumRecord - 1 Step 1

    dblTemp = dblTemp + dblTemp2(j) \* ordeB(ST, j, 1)

Next j

hitPersBonjean = dblTemp

End Function

Private Sub hitPalat()

txtPalat.Text = txtPkpal.Text \* comboPalat / 100

End Sub

Private Sub txtBsepatu\_GotFocus()

hitBsepatuMin

If CDbl(txtBsepatu.Text) < BsepatuMin Then

    MsgBox "Lebar minimum : " & BsepatuMin

    txtBsepatu.SetFocus

End If

End Sub

Private Sub txtBsepatu\_LostFocus()

hitBsepatuMin

If txtBsepatu.Text < BsepatuMin Then

    MsgBox "Lebar minimum : " & BsepatuMin

    txtBsepatu.SetFocus

End If

End Sub

Private Sub txtMiring\_GotFocus()



```
convSudut  
End Sub
```

```
Private Sub txtMiring_LostFocus()  
convSudut  
End Sub
```

```
Private Sub txtPkapal_Change()  
hitPalat  
End Sub
```

```
Private Sub txtPkapal_Click()  
hitPalat  
End Sub
```

```
Private Sub txtPkapal_LostFocus()  
hitPalat  
End Sub
```

```
Private Sub hitungDataKapal()
```

```
Ptotal = CDbl(txtPkapal.Text) + CDbl(txtPalat.Text)  
LCGtotal = (CDbl(txtPkapal.Text * txtLCGkapal.Text) + CDbl(txtPalat.Text *  
txtLCGalat.Text)) / Ptotal  
End Sub
```

```
Private Sub hitBsepatuMin()
```

```
hitungDataKapal  
BsepatuMin = Ptotal / (txtTekanan.Text * txtS.Text * txtNsepatu.Text)  
End Sub
```

```
Private Sub convSudut()
```

```
sudutA = (Atn(1 / txtMiring.Text)) * 180 / Pi  
txtSudut.Text = Round(sudutA, 3)
```

```
End Sub
```

```
Private Sub PERIODE1()  
Dim PtotSinA, PtotCosA As Double  
Dim Fper1 As Double
```

```
strPerhitungan = "P E R I O D E I :" & vbCrLf
```





```
strPerhitungan = strPerhitungan & "*****" & vbCrLf
```

```
PtotSinA = Ptotal * Sin(sudutA * Pi / 180)
```

```
PtotCosA = Ptotal * Cos(sudutA * Pi / 180) * txtMuS.Text
```

```
Fper1 = PtotSinA - PtotCosA
```

```
If Fper1 < 0 Then
```

```
    MsgBox "Kapal tidak dapat meluncur sendiri !" & vbCrLf & "Harus ditambahkan gaya dorong : " & Round(Fper1, 3) & " ton"
```

```
    Exit Sub
```

```
End If
```

```
q(0) = Ptotal / S
```

```
X(0) = S / 2 + LCGtotal
```

```
qA(0) = (2 * q(0) * (2 * S - 3 * X(0))) / S
```

```
qF(0) = (2 * q(0) * (3 * X(0) - S)) / S
```

```
strPerhitungan = strPerhitungan & "P = " & vbTab & Round(Ptotal, 2) & vbCrLf & "ton" & vbCrLf
```

```
strPerhitungan = strPerhitungan & "F = " & vbTab & Round(Fper1, 2) & vbCrLf & "ton" & vbCrLf
```

```
strPerhitungan = strPerhitungan & "Q = " & vbTab & Round(q(0), 2) & vbCrLf & "ton" & vbCrLf
```

```
strPerhitungan = strPerhitungan & "qA = " & vbTab & Round(qA(0), 2) & vbCrLf & "ton/m" & vbCrLf
```

```
strPerhitungan = strPerhitungan & "qF = " & vbTab & Round(qF(0), 2) & vbCrLf & "ton/m" & vbCrLf
```

```
End Sub
```

```
Private Sub PERIODE2()
```

```
Dim i, lk, FS, Langkah As Integer
```

```
Dim Station As Integer
```

```
Dim Sigma1, Sigma2 As Double
```

```
Dim Z1(100), Z2(100) As Double
```

```
Dim Luas As Double
```

```
Dim strTemp2 As String
```

```
Dim gA(100) As Double
```

```
Dim gAmax As Double
```

```
strPerhitungan = strPerhitungan & vbCrLf & vbCrLf & "P E R I O D E II :" & vbCrLf
```

```
strPerhitungan = strPerhitungan & "*****" & vbCrLf
```

```
Langkah = 1
```

```
awalP3 = 1
```



Do

Sigma1 = 0

Sigma2 = 0

lk = Langkah \* 2

strTemp2 = strTemp2 & "Langkah " & Langkah & " : s/d station "  
& lk & vbCrLf

Ta(Langkah) = (lk - 0) \* hStat \* Tan(sudutA \* Pi / 180)

strTemp2 = strTemp2 &amp; "Ta = " &amp; vbTab &amp; Round(Ta(Langkah), 2) &amp; vbCrLf

strTemp2 = strTemp2 &amp; headSigma

gAmax = 0

For i = 0 To lk

If i = 0 Or i = lk Then

FS = 1

Else

If i Mod 2 = 0 Then

FS = 2

Else

FS = 4

End If

End If

Station = i

Luas = hitPersBonjean(Station, (lk - i) \* hStat \* Tan(sudutA \* Pi / 180))

gA(i) = Luas

If gA(i) &gt; gAmax Then

gAmax = gA(i)

Else

gAmax = gAmax

End If

Sigma1 = Sigma1 + Luas \* FS

Sigma2 = Sigma2 + Luas \* FS \* i

strTemp2 = strTemp2 &amp; i &amp; vbTab

strTemp2 = strTemp2 &amp; Round(Luas, 2) &amp; vbTab

strTemp2 = strTemp2 &amp; FS &amp; vbTab

strTemp2 = strTemp2 &amp; Round(Luas \* FS, 2) &amp; vbTab

strTemp2 = strTemp2 &amp; i &amp; vbTab

strTemp2 = strTemp2 &amp; Round(Luas \* FS \* i, 2) &amp; vbCrLf

Next i

gLuas(Langkah) = gAmax

Z1(Langkah) = Sigma1

Z2(Langkah) = Sigma2

So(Langkah) = 2 \* Langkah \* hStat

Sn(Langkah) = So(Langkah) + z

Vaks(Langkah) = 1 / 3 \* hStat \* Z1(Langkah)



$gVaks(\text{Langkah}) = 1.004 * 1.025 * Vaks(\text{Langkah})$   
 $f(\text{Langkah}) = Z2(\text{Langkah}) / Z1(\text{Langkah}) * hStat$   
 $d(\text{Langkah}) = (S + h) - f(\text{Langkah})$   
 $gVaks_d(\text{Langkah}) = gVaks(\text{Langkah}) * d(\text{Langkah})$   
 $A(\text{Langkah}) = (Sn(\text{Langkah}) - Lamda) - g$   
 $b(\text{Langkah}) = (Sn(\text{Langkah}) - Lamda) - f(\text{Langkah})$   
 $Ptotal_a(\text{Langkah}) = Ptotal * A(\text{Langkah})$   
 $gVaks_b(\text{Langkah}) = gVaks(\text{Langkah}) * b(\text{Langkah})$   
 $Qbesar(\text{Langkah}) = Ptotal - gVaks(\text{Langkah})$   
If  $(So(\text{Langkah}) - h + z) < \text{Lamda}$  Then  
     $Saks(\text{Langkah}) = S$   
     $xAks(\text{Langkah}) = ((gVaks_b(\text{Langkah}) - Ptotal_a(\text{Langkah})) - Qbesar(\text{Langkah})) / Sn(\text{Langkah}) - Lamda - h$   
Else  
     $Saks(\text{Langkah}) = (S + h) - f(\text{Langkah}) - b(\text{Langkah})$   
     $xAks(\text{Langkah}) = (gVaks_b(\text{Langkah}) - Ptotal_a(\text{Langkah})) / Qbesar(\text{Langkah})$   
End If  
     $Saks23(\text{Langkah}) = 2 / 3 * Saks(\text{Langkah})$   
     $Saks12(\text{Langkah}) = 1 / 2 * Saks(\text{Langkah})$   
     $Saks13(\text{Langkah}) = 1 / 3 * Saks(\text{Langkah})$   
If  $xAks(\text{Langkah}) > Saks23(\text{Langkah})$  Then  
     $q(\text{Langkah}) = Qbesar(\text{Langkah}) / (3 * xAks(\text{Langkah}))$   
     $qA(\text{Langkah}) = 0$   
     $qF(\text{Langkah}) = q(\text{Langkah}) * 2$   
    GoTo segiTiga  
Else  
    If  $xAks(\text{Langkah}) > Saks13(\text{Langkah})$  Then  
         $q(\text{Langkah}) = Qbesar(\text{Langkah}) / Saks(\text{Langkah})$   
         $qA(\text{Langkah}) = 2 * q(\text{Langkah}) * (2 * Saks(\text{Langkah}) - 3 * xAks(\text{Langkah})) / Saks(\text{Langkah})$   
         $qF(\text{Langkah}) = 2 * q(\text{Langkah}) * (3 * xAks(\text{Langkah}) - Saks(\text{Langkah})) / Saks(\text{Langkah})$   
        GoTo segiTiga  
    Else  
         $q(\text{Langkah}) = Qbesar(\text{Langkah}) / (3 * xAks(\text{Langkah}))$   
         $qA(\text{Langkah}) = q(\text{Langkah}) * 2$   
         $qF(\text{Langkah}) = 0$   
        GoTo segiTiga  
    End If  
End If  
segiTiga:  
     $\text{strTemp2} = \text{strTemp2} & \text{vbTab} & \text{vbTab} & \text{vbTab} & \text{Round}(Z1(\text{Langkah}), 2) - \& \text{vbTab} & \text{vbTab} & \text{Round}(Z2(\text{Langkah}), 2) & \text{vbCrLf} & \text{vbCrLf}$   
If  $gVaks_d(\text{Langkah}) > Ptotal_c$  Then



```
GoTo SternLift
End If

awalP3 = awalP3 + 1
Langkah = Langkah + 1
Loop
SternLift:

strTemp2 = strTemp2 & "REKAPITULASI PERHITUNGAN PERIODE II :" &
vbCrLf & vbCrLf
strTemp2 = strTemp2 & "Nomor" & vbTab & "Notasi" & vbTab & _
"Satuan" & vbTab

For i = 1 To Langkah
    strTemp2 = strTemp2 & i & vbTab & vbTab
Next i

strTemp2 = strTemp2 & vbCrLf

strTemp2 = strTemp2 & "1" & vbTab & "Zigma1" & vbTab & "m2" & vbTab
    For i = 1 To Langkah
        strTemp2 = strTemp2 & Round(Z1(i), 2) & vbTab & vbTab
    Next i
    strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "2" & vbTab & "Zigma2" & vbTab & "m2" & vbTab
    For i = 1 To Langkah
        strTemp2 = strTemp2 & Round(Z2(i), 2) & vbTab & vbTab
    Next i
    strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "3" & vbTab & "So" & vbTab & "m" & vbTab
    For i = 1 To Langkah
        strTemp2 = strTemp2 & Round(So(i), 2) & vbTab & vbTab
    Next i
    strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "4" & vbTab & "Sn" & vbTab & "m" & vbTab
    For i = 1 To Langkah
        strTemp2 = strTemp2 & Round(Sn(i), 2) & vbTab & vbTab
    Next i
    strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "5" & vbTab & "Ta" & vbTab & "m" & vbTab
    For i = 1 To Langkah
        strTemp2 = strTemp2 & Round(Ta(i), 2) & vbTab & vbTab
    Next i
    strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "6" & vbTab & "gV" & vbTab & "ton" & vbTab
    For i = 1 To Langkah
```



```
strTemp2 = strTemp2 & Round(gVaks(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "7" & vbTab & "Ptotal" & vbTab & "ton" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Ptotal, 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "7b" & vbTab & "Q" & vbTab & "ton" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Qbesar(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "8" & vbTab & "Ptot.c" & vbTab & "ton m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Ptotal_c, 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "9" & vbTab & "gV.d" & vbTab & "ton m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(gVaks_d(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "10" & vbTab & "Ptot.a" & vbTab & "ton m" & vbTab
For i = 1 To Langkah
    If Len(Round(Ptotal_a(i), 2)) > 8 Then
        strTemp2 = strTemp2 & Round(Ptotal_a(i), 2) & vbTab
    Else
        strTemp2 = strTemp2 & Round(Ptotal_a(i), 2) & vbTab & vbTab
    End If
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "11" & vbTab & "gV.b" & vbTab & "ton m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(gVaks_b(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "12" & vbTab & "S" & vbTab & "m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Saks(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "13" & vbTab & "2/3S" & vbTab & "m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Saks23(i), 2) & vbTab & vbTab
Next i
```



```
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "14" & vbTab & "1/2S" & vbTab & "m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Saks12(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "15" & vbTab & "1/3S" & vbTab & "m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(Saks13(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "16" & vbTab & "x" & vbTab & "m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(xAks(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "17" & vbTab & "q" & vbTab & "ton/m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(q(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "18" & vbTab & "qA" & vbTab & "ton/m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(qA(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf
strTemp2 = strTemp2 & "19" & vbTab & "qF" & vbTab & "ton/m" & vbTab
For i = 1 To Langkah
    strTemp2 = strTemp2 & Round(qF(i), 2) & vbTab & vbTab
Next i
strTemp2 = strTemp2 & vbCrLf

strTemp2 = strTemp2 & vbCrLf
strPerhitungan = strPerhitungan & strTemp2
```

End Sub

Private Function headSigma() As String

Dim strTemp As String

```
strTemp = strTemp & "Stat" & vbTab
strTemp = strTemp & "Luas" & vbTab
strTemp = strTemp & "FS" & vbTab
strTemp = strTemp & "A*FS" & vbTab
```



```
strTemp = strTemp & "FM" & vbTab
strTemp = strTemp & "A*FS*FM" & vbCrLf

headSigma = strTemp

End Function

Private Sub PER3()
Dim Langkah3, Sarat3, i, Station3 As Integer
Dim strTemp3 As String
Dim Lk3, FS3 As Integer
Dim Luas3, Luas3FS, Luas3FSi As Double
Dim Z31(100, 3), Z32(100, 3), d3(100, 3) As Double
Dim b3(3), a3(3) As Double

strPerhitungan = strPerhitungan & vbCrLf & "P E R I O D E III :" & vbCrLf
strPerhitungan = strPerhitungan & "*****" & vbCrLf
Langkah3 = awalP3
Do
    strTemp3 = strTemp3 & "Langkah " & Langkah3 & " : s/d station " _
        & Langkah3 * 2 & vbCrLf
    Lk3 = Langkah3 * 2
    So(Langkah3) = Lk3 * hStat
    Sn(Langkah3) = So(Langkah3) + z
    For Sarat3 = 1 To 3 Step 1
        Ta3(Langkah3, Sarat3) = ((h + S) - (0 * hStat)) * _
            Tan(Sarat3 / 3 * sudutA * Pi / 180) + (So(Langkah3) - (h + S)) _ 
            * Tan(Sarat3 / 3 * sudutA * Pi / 180)
        strTemp3 = strTemp3 & "Langkah " & Langkah3 & " ; sarat = " _
            & Round(Ta3(Langkah3, Sarat3), 2) & vbCrLf
        strTemp3 = strTemp3 & headSigma
        Luas3FS = 0
        Luas3FSi = 0
        For i = 0 To Lk3 Step 1
            If i = 0 Or i = 20 Or i = Lk3 Then
                FS3 = 1
            Else
                If i Mod 2 = 0 Then
                    FS3 = 2
                Else
                    FS3 = 4
                End If
            End If
            Station3 = i
            Luas3 = hitPersBonjean(Station3, ((h + S) - (i * hStat)) * _
                Tan(Sarat3 / 3 * sudutA * Pi / 180) + (So(Langkah3) - (h + S)) _
```



$$* \text{Tan}(\text{Sarat3} / 3 * \text{sudutA} * \pi / 180))$$

$$\text{Luas3FS} = \text{Luas3FS} + \text{Luas3} * \text{FS3}$$

$$\text{Luas3FSi} = \text{Luas3FSi} + \text{Luas3} * \text{FS3} * i$$

If  $i > 20$  Then

    GoTo byPass

Else

$$\text{strTemp3} = \text{strTemp3} & i & \text{vbTab} & \text{Round}(\text{Luas3}, 2) & \text{vbTab} & \text{FS3}$$

$$& \text{vbTab} & \text{Round}(\text{Luas3} * \text{FS3}, 2) & \text{vbTab} & i & \text{vbTab} _$$

$$& \text{Round}(\text{Luas3} * \text{FS3} * i, 2) & \text{vbCrLf}$$

End If

byPass:

    Next i

$$Z31(\text{Langkah3}, \text{Sarat3}) = \text{Luas3FS}$$

$$Z32(\text{Langkah3}, \text{Sarat3}) = \text{Luas3FSi}$$

$$gV3(\text{Langkah3}, \text{Sarat3}) = 1.004 * 1.025 * 1 / 3 * hStat * Z31(\text{Langkah3}, \text{Sarat3})$$

$$d3(\text{Langkah3}, \text{Sarat3}) = (S + h) - (Z32(\text{Langkah3}, \text{Sarat3}) / Z31(\text{Langkah3}, \text{Sarat3})) *$$

$hStat$ )

$$gV3d(\text{Langkah3}, \text{Sarat3}) = gV3(\text{Langkah3}, \text{Sarat3}) * d3(\text{Langkah3}, \text{Sarat3})$$

$$\text{strTemp3} = \text{strTemp3} & \text{vbTab} & \text{vbTab} & \text{vbTab} & \text{Round}(\text{Luas3FS}, 2) & \text{vbTab}$$

$$& \text{vbTab} & \text{Round}(\text{Luas3FSi}, 2) & \text{vbCrLf}$$

$$\text{strTemp3} = \text{strTemp3} & "gV : " & \text{vbTab} & \text{Round}(gV3(\text{Langkah3}, \text{Sarat3}), 2) & \text{vbCrLf}$$

$$\text{strTemp3} = \text{strTemp3} & "Ta : " & \text{vbTab} & \text{Round}(Ta3(\text{Langkah3}, \text{Sarat3}), 2) & \text{vbCrLf}$$

    strTemp3 = strTemp3 & vbCrLf

    Next Sarat3

$$b3(1) = Ta3(\text{Langkah3}, 1)$$

$$b3(2) = (Ta3(\text{Langkah3}, 2) - Ta3(\text{Langkah3}, 1)) / (gV3d(\text{Langkah3}, 2) - gV3d(\text{Langkah3}, 1))$$

$$b3(3) = ((Ta3(\text{Langkah3}, 3) - Ta3(\text{Langkah3}, 2)) / (gV3d(\text{Langkah3}, 3) - gV3d(\text{Langkah3}, 2)) -$$

$$- (Ta3(\text{Langkah3}, 2) - Ta3(\text{Langkah3}, 1)) / (gV3d(\text{Langkah3}, 2) - gV3d(\text{Langkah3}, 1)))$$

$$/ (gV3d(\text{Langkah3}, 3) - gV3d(\text{Langkah3}, 1))$$

$$Ta(\text{Langkah3}) = b3(1) + b3(2) * (Ptotal_c - gV3d(\text{Langkah3}, 1)) + b3(3) * _$$

$$(Ptotal_c - gV3d(\text{Langkah3}, 1)) * (Ptotal_c - gV3d(\text{Langkah3}, 2))$$

$$a3(1) = gV3(\text{Langkah3}, 1)$$

$$a3(2) = (gV3(\text{Langkah3}, 2) - gV3(\text{Langkah3}, 1)) / (Ta3(\text{Langkah3}, 2) - Ta3(\text{Langkah3}, 1))$$



```
a3(3) = ((gV3(Langkah3, 3) - gV3(Langkah3, 2)) / (Ta3(Langkah3, 3) - Ta3(Langkah3, 2))_
- (gV3(Langkah3, 2) - gV3(Langkah3, 1)) / (Ta3(Langkah3, 2) - Ta3(Langkah3, 1)))_
/ (Ta3(Langkah3, 3) - Ta3(Langkah3, 1))
gVaks(Langkah3) = a3(1) + a3(2) * (Ta(Langkah3) - Ta3(Langkah3, 1)) +
a3(3) *
(Ta(Langkah3) - Ta3(Langkah3, 1)) * (Ta(Langkah3) - Ta3(Langkah3, 2))
Qbesar(Langkah3) = Ptotal - gVaks(Langkah3)
q(Langkah3) = Qbesar(Langkah3) / (0.05 * S)

strTemp3 = strTemp3 & "Sarat sebenarnya : " & vbTab &
Round(Ta(Langkah3), 2)_
& vbTab & "gV : " & vbTab & Round(gVaks(Langkah3), 2) & vbCrLf &
vbCrLf

If gVaks(Langkah3) > Ptotal Then
    GoTo Floating
End If
Langkah3 = Langkah3 + 1
awalP4 = Langkah3
Loop
Floating:
strPerhitungan = strPerhitungan & strTemp3 & vbCrLf

End Sub
```

```
Private Sub iterNolQ()
Dim i As Integer

For i = 0 To awalP3 Step 1
    ordeQ(0, i) = Qbesar(i + 1)
Next i
```

```
End Sub
```

```
Private Sub iterNolGV()
Dim i As Integer

For i = 0 To awalP3 Step 1
    ordeGV(0, i) = Sn(i + 1)
Next i
```

```
End Sub
```

```
Private Sub iterSatuQ()
```



Dim i As Integer

For i = 1 To awalP3 Step 1

    ordeQ(1, i) = Qbesar(i + 1) - Qbesar(i)

    ordeQ(1, i) = ordeQ(1, i) / (Sn(i + 1) - Sn(i))

Next i

End Sub

Private Sub iterSatuGV()

Dim i As Integer

For i = 1 To awalP3 Step 1

    ordeGV(1, i) = Sn(i + 1) - Sn(i)

    ordeGV(1, i) = ordeGV(1, i) / (gVaks(i + 1) - gVaks(i))

Next i

End Sub

Private Sub iterDuaQ()

Dim i, j As Integer

For j = 2 To awalP3 Step 1

    For i = 1 To awalP3 - j Step 1

        ordeQ(j, i) = ordeQ(j - 1, i + 1) - ordeQ(j - 1, i)

        ordeQ(j, i) = ordeQ(j, i) / (Sn(i + j) - Sn(i))

    Next i

Next j

End Sub

Private Sub iterDuaGV()

Dim i, j As Integer

For j = 2 To awalP3 Step 1

    For i = 1 To awalP3 - j Step 1

        ordeGV(j, i) = ordeGV(j - 1, i + 1) - ordeGV(j - 1, i)

        ordeGV(j, i) = ordeGV(j, i) / (gVaks(i + j) - gVaks(i))

    Next i

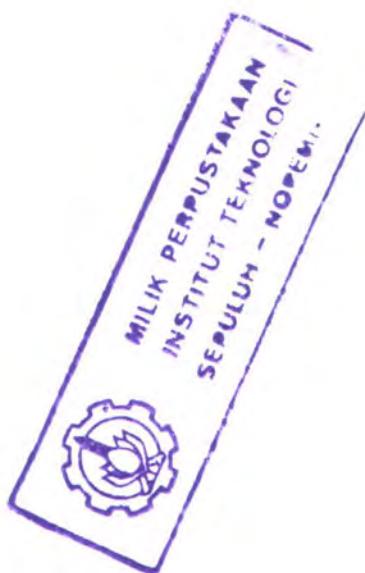
Next j

End Sub

Private Function hitPersQ(X As Double) As Double

Dim j As Integer

Dim dblTemp As Double





Dim dblTemp2(100) As Double

dblTemp2(0) = 1

For j = 1 To awalP3 Step 1

    dblTemp2(j) = dblTemp2(j - 1) \* (X - Sn(j))

Next j

dblTemp = ordeQ(0, 0)

For j = 1 To awalP3 Step 1

    dblTemp = dblTemp + dblTemp2(j) \* ordeQ(j, 1)

Next j

hitPersQ = dblTemp

End Function

Private Sub Gerak1()

Dim Percpt, dT As Double

Dim Kec(100), Jarak(100) As Double

Dim Sx, Tx As Double

strDinamis = ""

strDinamis = strDinamis & "Periode I : " & vbCrLf

Percpt = Grav \* (Sin(sudutA / 180 \* Pi) - mUd \* Cos(sudutA / 180 \* Pi))

grafA(0) = Percpt

strDinamis = strDinamis & "Waktu" & vbTab & "Kecpt" & vbTab & "Jarak" & vbCrLf

dT = 0

Kec(0) = 0

grafV(0) = Kec(0)

Jarak(0) = 0

grafS(0) = Jarak(0)

Tx = (2 \* jSpt / Percpt) ^ 0.5

Do

    Kec(dT) = Percpt \* dT

    Jarak(dT) = 1 / 2 \* Percpt \* dT ^ 2

    If Jarak(dT) >= jSpt Then

        GoTo AkhirPer1

    End If

    grafA(dT) = Percpt

    grafV(dT) = Kec(dT)

    V1g = grafV(dT)

    grafS(dT) = Jarak(dT)

    strDinamis = strDinamis & Round(dT, 1) & vbTab & Round(Kec(dT), 2) & vbTab

        & Round(Jarak(dT), 2) & vbCrLf



```
dT = dT + 0.1
Loop
AkhirPer1:
End Sub

Private Sub Gerak2()
Dim gLangkah As Integer
'Dim R, c, k, vx2 As Double
Dim Rwater, xWater, cWater, kWATER, vWater As Double

strDinamis = strDinamis & vbCrLf & "Periode II :" & vbCrLf
strDinamis = strDinamis & "Waktu" & vbTab & "Percpt" & vbTab & "Kecpt" _
& vbTab & "Jarak" & vbCrLf
gLangkah = 1
Do
If gLangkah = awalP3 Then
    GoTo LanjutD2
End If
xWater = (Ptotal - hitPersQ(Sn(gLangkah))) / Ptotal * 100
If xWater <= 50 Then
    cWater = 0 + (xWater - 0) * (20.5549845837616) + (xWater - 0) * (xWater - 4.865) * (-0.345662349158432) _
+ (xWater - 0) * (xWater - 4.865) * (xWater - 10.811) * (-5.11858835272062E-03) _
+ (xWater - 0) * (xWater - 4.865) * (xWater - 10.811) * (xWater - 21.892) *
(6.22752936306542E-04) _
+ (xWater - 0) * (xWater - 4.865) * (xWater - 10.811) * (xWater - 21.892) *
(xWater - 30) * (-2.58710542393829E-05) _
+ (xWater - 0) * (xWater - 4.865) * (xWater - 10.811) * (xWater - 21.892) *
(xWater - 30) * (xWater - 38.108) * (6.56537937658761E-07)
Else
    If xWater <= 92.703 Then
        cWater = 445 + (xWater - 50) * (5.08741097030802) + (xWater - 50) * (xWater - 60.811) * (0.153726088768308) _
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (6.58484654604447E-03) _
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (xWater - 72.162) * (1.50194668361913E-03) _
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (xWater - 72.162) * (xWater - 77.297) * (6.72817937057864E-05) _
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (xWater - 72.162) * (xWater - 77.297) * (xWater - 80) * (-4.29536060614839E-05) _
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (xWater - 72.162) * (xWater - 77.297) * (xWater - 80) * (xWater - 82.703) * (4.40426757877608E-06)
```



```
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (xWater - 72.162) * (xWater - 77.297) * (xWater - 80) * (xWater - 82.703) * (xWater - 85)  
* (-1.99588925537081E-07)  
+ (xWater - 50) * (xWater - 60.811) * (xWater - 70) * (xWater - 72.162) * (xWater - 77.297) * (xWater - 80) * (xWater - 82.703) * (xWater - 85)  
* (xWater - 88.108) * (2.57151884934383E-09)  
Else  
cWater = 1025 + (xWater - 92.703) * (-6.1667488899852) + (xWater - 92.703) * (xWater - 96.757) * (-3.38069029968462)  
End If  
End If  
'MsgBox cWater  
kWater = (Ptotal - hitPersQ(Sn(gLangkah))) ^ (2 / 3) / cWater  
vWater = V1g * 3.28083989501312  
Rwater = kWater * vWater ^ 2  
grafA(gLangkah) = Grav / Ptotal * (hitPersQ(Sn(gLangkah)) * (Sin(sudutA /  
180 * Pi) - mUd)  
* Cos(sudutA / 180 * Pi)) - Rwater)  
  
'k = (Grav / Ptotal) * 6.5 * (gLatas(gLangkah) + 9.25 * Aspt) / jSpt  
'c = (1 - (hitPersQ(Sn(gLangkah)) / Ptotal)) ^ (k * jSpt ^ 2) / (jSpt ^ 2)  
'vx2 = (2 * Grav * (sudutA - mUd) * jSpt * (1 - c / 3 * jSpt ^ 2) + V1g ^ 2) / (k  
* jSpt ^ 2)  
  
'R = 6.5 * (gLatas(gLangkah) + 9.25 * Aspt) * vx2  
'R = 0  
'grafA(gLangkah) = Grav / Ptotal * (hitPersQ(Sn(gLangkah)) * (Sin(sudutA /  
180 * Pi) - mUd)  
* Cos(sudutA / 180 * Pi)) - R)  
  
'grafT(gLangkah) = (2 * Sn(gLangkah) / grafA(gLangkah)) ^ 0.5  
'grafV(gLangkah) = (2 * grafA(gLangkah) * Sn(gLangkah)) ^ 0.5  
  
strDinamis = strDinamis & Round(grafT(gLangkah), 2) & vbTab &  
Round(grafA(gLangkah), 2)  
& vbTab & Round(grafV(gLangkah), 2) & vbTab & Round(Sn(gLangkah),  
2) & vbCrLf  
  
gLangkah = gLangkah + 1  
Loop  
LanjutD2:  
  
End Sub  
  
Sub yDrawing()  
' This example creates a spline object in model space.
```



On Error Resume Next

```
' Connect to the AutoCAD application
Set acadApp = GetObject _
    ("AutoCAD.Application")
If Err Then
    Err.Clear
    Set acadApp = CreateObject _
        ("AutoCAD.Application")
If Err Then
    MsgBox Err.Description
    Exit Sub
End If
End If

' Connect to the AutoCAD drawing
Set acadDoc = acadApp.ActiveDocument
End Sub

Sub OpenDrawing()

    Dim dwgName As String
    dwgName = App.Path & "\MODELtemplate.dwg"
    If Dir(dwgName) <> "" Then
        acadDoc.Application.Documents.Open dwgName
    Else
        MsgBox "File " & dwgName & " does not exist."
    End If
    Set acadDoc = acadApp.ActiveDocument
End Sub
```