



**SKRIPSI - ME 141501**

**PERENCANAAN SIMULASI PENGATURAN  
PEMBANGKITAN DAYAPADA KAPAL FAST  
PATROL BOAT 60 M DENGAN PROPULSI  
HYBRID**

**FIKRI NURUDDIN MUZAKKI NRP 4212 100 072**

**Dosen Pembimbing :  
Dr. Eddy Setyo K. ST. MSc  
Adi Kurniawan, ST. MT**

**JURUSAN TEKNIK SISTEM PERKAPALAN  
Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember  
Surabaya 2016**

*“Halaman ini sengaja di kosongkan”*



---

**FINAL PROJECT - ME 141501**

**DESIGN SIMULATION ARRANGEMENT OF  
POWER GENERATIONS OF FAST PATROL  
BOAT 60 M WITH HYBRID PROPULSION**

**FIKRI NURUDDIN MUZAKKI NRP 4212 100 072**

**Supervisors :**

**Dr. Eddy Setyo K. ST. MSc**

**Adi Kurniawan, ST. MT**

**DEPARTEMENT OF MARINE ENGINEERING  
Faculty of Marine Technology  
Institute of Technology Sepuluh Nopember  
Surabaya 2016**

*“Halaman ini sengaja di kosongkan”*

# LEMBAR PENGESAHAN

## PERENCANAAN SIMULASI PENGATURAN PEMBANGKITAN DAYA PADA KAPAL FAST PATROL BOAT 60 M DENGAN PROPULSI HYBRID

### SKRIPSI-

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada  
Bidang Study *Marine Electrical and Automation System*  
(MEAS) Program Studi S-1 Jurusan Teknik Sistem  
Perkapalan Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember

Oleh :

**FIKRI NURUDDIN MUZAKKI**  
NRP. 4212100072

Disetujui oleh Dosen Pembimbing Tugas Akhir

Dr. Eddy Setyo Koenhardono, ST. MSc.



Adi Kurniawan, ST. MT.



*“Halaman ini sengaja di kosongkan”*

## LEMBAR PENGESAHAN

### PERENCANAAN SIMULASI PENGATURAN PEMBANGKITAN DAYA PADA KAPAL FAST PATROL BOAT 60 M DENGAN PROPULSI HYBRID

### SKRIPSI

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Teknik  
pada

Bidang Study *Marine Electrical and Automation System*  
(MEAS) Program Studi S-1 Jurusan Teknik Sistem  
Perkapalan Fakultas Teknologi Kelautan  
Institut Teknologi Sepuluh Nopember

Oleh :

**FIKRI NURUDDIN MUZAKKI**

NRP. 42 12 100 072

Disetujui oleh Ketua Jurusan Teknik Sistem Perkapalan



Dr. Eng. M. Badruz Zaman, S.T., M.T.

NIP. 197708022008011007

*“Halaman ini sengaja di kosongkan”*



**PERENCANAAN SIMULASI PENGATURAN  
PEMBANGKITAN DAYA PADA KAPAL FAST  
PATROL BOAT 60 M DENGAN PROPULSI  
HYBRID**

**Nama Mahasiswa** : Fikri Nuruddin Muzakki  
**NRP** : 4212100072  
**Jurusan** : Teknik Sistem Perkapalan  
**Dosen Pembimbing** : 1. Dr. Eddy Setyo  
Koenhardono, ST. M.Sc.  
2. Adi Kurniawan, ST. MT.

**Abstrak**

Perkembangan teknologi memberikan berbagai inovasi baru termasuk dalam hal propulsi kapal. Perkembangan tersebut antara lain adalah desain-desain baru mengenai sistem propulsi hybrid yang memberikan berbagai keuntungan untuk pengoperasian kapal. Seperti pertimbangan konsumsi bahan bakar yang rendah, pengaturan kecepatan dengan penggunaan propulsi yang optimal ataupun khusus untuk jenis kapal tertentu. Salah satu contohnya sistem boosting untuk menambah daya yang dihasilkan pada ruangan konstruksi propulsi yang terbatas. Tentunya sistem propulsi hybrid memiliki kelamahan pula, salah satunya adalah pengaturan pembangkitan daya yang kompleks dibandingkan dengan system propulsi lain.

Setiap kecepatan akan membutuhkan daya yang berbeda dan bahan bakar yang berbeda pula. Keputusan yang kurang tepat dapat mengurangi keunggulan dari pemasangan sistem propulsi hybrid. Oleh karena itu dibutuhkan *Decision Support System* untuk membantu nahkoda menentukan pengaturan propulsi yang optimal pada kecepatan yang diinginkan dan pada kondisi yang ada pada perairan yang dilalui. Adanya *Decision Support System* akan memberikan informasi kepada nahkoda dengan menunjukkan sistem propulsi yang tersedia dengan jumlah konsumsi bahan bakar yang diperlukan,

sehingga nahkoda dapat memilih propulsi yang optimal dengan konsumsi bahan bakar yang minimum, dan dengan jumlah daya yang diperlukan untuk pengoperasian tersebut.

Penggunaan *Decision Support System* pada kapal fast Patrol Boat 60 m ini menunjukkan pemakaian terbaik pada kecepatan 20 knot dengan menggunakan sistem Shaft Generator pada daya 1181 kW disertai pengoperasian genset pada daya 0 kW dengan putaran mesin 1334 rpm dan jumlah pemakaian bahan bakar 482,99 kg perjam.

***Keyword : Decision Support System, Fast Patrol Boat 60 m, Hybrid Propulsion System.***

# **DESIGN SIMULATION ARRANGEMENT OF POWER GENERATION ON FAST PATROL BOAT 60 M WITH HYBRID PROPULTION**

**Student Name : Fikri Nuruddin Muzakki**

**NRP : 4212100072**

**Departement : Marine Engineering**

**Consulting Lecturer : 1. Dr. Eddy Setyo**

**Koenhardono, ST. M.Sc.**

**2. Adi Kurniawan, ST. MT.**

## **Abstract**

Development of Technology give a lot of new innovation including the term of ships propulsion. Particularly new designs of hybrid propulsion system which give any advantage for ships operation, such as low fuel oil consumptions, speed control with optimum operation of propulsion system or ship with particular purpose, for example with boosting system for increasing power generation with limited space for construction of propulsion system. Of course Hybrid propulsion system have a disadvantage, such as a very complex power generating arrangement than any other propulsion system.

In every operation speed will need a different power supply and fuel oil consumption also different, inaccurate decision will lead to decrease the advantage of hybrid propulsion system itself, so decision support system is needed for help master to choose propulsion system that have optimal thrust for minimum fuel oil consumption, with torque and rotation speed of engine in very right operation.

Utilization of Decision Support system for Fast Patrol Boat 60 m will show every possible propulsion system that available at that very speed and sail condition to choose the best possible system such as, for vessel speed at 20 knot will use Shaft Generator propulsion system at 0 kW power with Generator set

operation at 1131 kW and 1334 rpm engine rotation and fuel oil consumption 482,99 kg per hour.

***Keyword :*** *Decision Support System, Fast Patrol Boat 60 m, Hybrid Propulsion.*

## KATA PENGANTAR

Alhamdulillah... Segala puja dan puji syukur penulis panjatkan kehadirat Allah SWT sebagai Penguasa Suluruh Alam, atas rahmat dan karunia-Nya serta bimbingan-Nya sehingga penulis dapat menyelesaikan Skripsi dengan judul “Perencanaan Simulasi Pembangkitan Daya Pada Kapal Fast Patrol Boat 60 M Dengan Propulsi Hybrid”.

Tugas Skripsi adalah salah satu syarat wajib untuk menyelesaikan studi program sarjana (S1) di Jurusan Teknik Sistem Perkapalan, Fakultas Teknologi Kelautan, Institut Teknologi Sepuluh Nopember Surabaya. Dalam tugas skripsi ini, penulis merencanakan sebuah sistem propulsi berbasis hybrid pada Kapal Fast Patrol Boat 60 M untuk di simulasi tentang pembangkitan daya saat operasi dan dianalisa tentang pengaruh pada konsumsi bahan bakar, sebagai sumber tenaga mesin penggerak maupun pembangkit listrik di kapal.

Dengan adanya pembuatan simulasi pengaturan pembangkitan daya ini penulis berharap agar dapat membantu untuk studi lanjutan dan bermanfaat untuk kemajuan teknologi tentang dunia perkapalan terutama mengenai sistem propulsi dan pembangkitan tenaga listrik.

Banyak dorongan dan bantuan yang penulis dapatkan selama penyusunan hingga terselesaikannya Skripsi ini. Pada kesempatan ini perkenankanlah penyusun menyampaikan banyak terimakasih kepada:

1. Bapak Dr. Eng. M Badrus Zaman, ST. MT. selaku Ketua Jurusan Teknik Sistem Perkapalan yang telah memberikan bimbingan dan bantuan

sehingga penulis dapat menyelesaikan seluruh kegiatan perkuliahan dan pengerjaan Skripsi.

2. Bapak Eddy Setyo K. ST. MSc, selaku dosen pembimbing dalam pengerjaan skripsi ini. Terima kasih banyak atas ilmu dan perhatian yang bapak berikan.

3. Bapak Adi Kurniawan, ST. MT. selaku dosen pembimbing dalam pengerjaan Skripsi ini. Terima kasih banyak atas ilmu dan perhatian yang bapak berikan.

4. Bapak Dr. ABB Dinariyana D.P., ST. M.Sc. selaku dosen wali yang telah banyak meluangkan waktunya selama ini dan memberikan semangat untuk cepat lulus.

5. Bapak-bapak dosen laboratorium MEAS yang telah membantu memberikan pengarahan serta semangat dan motivasi kepada penulis dalam mengerjakan skripsi berikut.

6. Kepada kedua orang tua: Bapak Amin Ashari dan Ibu Juwarti sebagai pendorong motivasi serta pemberi semangat dan dukungan doa yang tak pernah putus.

7. Kepada teman-teman semua yang telah memberikan motivasi dan dukungan terhadap penulis.

Penulis menyadari bahwa penulisan Skripsi ini masih memiliki banyak kekurangan, sehingga mengharapkan kritik dan saran dari pembaca. Dan penulis memohon maaf jika terdapat kesalahan yang membuat pihak manapun kurang berkenan. Seoga tugas skripsi ini dapat memberikan manfaat bagi kita semua, Amin.

## Daftar Isi

Abstrak .....	vii
KATA PENGANTAR.....	xi
Daftar Isi.....	xiii
Daftar Gambar.....	xi
<b>BAB I PENDAHULUAN .....</b>	<b>xv</b>
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah .....	3
1.3 Tujuan Penulisan .....	4
1.4 Manfaat Penulisan .....	4
<b>BAB II TINJAUAN PUSTAKA .....</b>	<b>5</b>
2.1 Hybrid Propulsion System.....	5
2.2 Decision Support System (DSS).....	11
<b>BAB III METODOLOGI PENELITIAN .....</b>	<b>13</b>
<b>BAB IV ANALISA DATA DAN PEMBAHASAN ....</b>	<b>17</b>
4.1 Umum.....	17
4.2 Data Kapal.....	17
4.3 Analisa Rencana Permesinan dengan Sistem Hybrid.....	21
4.4 Perencanaan Pengerjaan Simulasi .....	24
4.5 Desain Interface DSS FPB 60 M.....	26
4.6 Analisa EPM pada Hasil Program Simulasi .....	31
4.7 Analisa Bahan Bakar Simulasi .....	35
<b>BAB V KESIMPULAN DAN SARAN .....</b>	<b>41</b>
5.1 Kesimpulan.....	41
5.2 Saran.....	42

DAFTAR PUSTAKA..... 43



## Daftar Gambar

Gambar 2.1. desain propulsi hybrid .....	6
Gambar 2.2. grafik peforma motor diesel (Kwasiackyj, 2013).....	7
Gambar 2.3 grafik kosumsi bahan bakar tunggal pada motor diesel L48/60 CR .....	7
Gambar 2.4 perbandingan efisiensi Permanent Magnet Generator terhadap Eletrically Excited Synchronous Generator ( <i>the switch</i> , 2006) .....	9
Gambar 2.5 Struktur Decision Support System.....	12
Gambar 3.1 Diagram Alur Metodologi penelitian.....	15
Gambar 4.1 Desain Lines Plan FPB 60 M menggunakan software Maxurf .....	18
Gambar 4.2 Grafik Tahanan dinas FPB 60 M.....	19
Gambar. 4.3 Performa Mesin Diesel (MTU 16V 4000 90M) .....	21
Gambar 4.4 engine propeller matching pada kecepatan 10-30 knot dengan SM 1.1 .....	22
Gambar 4.5 Performa dan SFOC MTU 16V 4000 90M .....	24
Gambar 4.6 Gambar Langkah Kerja Simulasi.....	25
Gambar 4.7 Gambar Interface Program Simulasi untuk Mode Elektris .....	28
Gambar 4.8 Gambar Interface Program Simulasi untuk Mode Shaft Generator .....	29
Gambar 4.9 Gambar Interface Program Simulasi untuk Mode Mekanis .....	30
Gambar 4.10 Grafik daya dan putaran pada kecepatan 17 knot dengan SM 1-2 dibanding kecepatan 10-30 .....	31

Gambar 4.11 Gambar perbandingan daya dan putaran dari perubahan SM dan kecepatan.....	32
Gambar 4.12 Analisa pemakaian Shaft Generator pada kebutuhan listrik 100 kW, 300 kW dan 500 kW pada SM 1.1 .....	34
Gambar 4.13 Gambar Analisa Konsumsi Bahan Bakar SM 1.2 Kebutuhan Listrik 300 kW .....	35
Gambar 4.14 Gambar Analisa Konsumsi Bahan Bakar SM 1.2 Kebutuhan Listrik 500 kW .....	36
Gambar 4.15 Gambar Analisa Konsumsi Bahan Bakar SM 1.2 Kebutuhan Listrik 700 kW .....	37
Gambar 4.16 Gambar Perbandingan Konsumsi Bahan Bakar antara Hybrid system dan Sistem Mekanis SM 1.2 Kebutuhan Listrik 500 kW .....	39
Gambar 4.17 Gambar Perbandingan Konsumsi Bahan Bakar antara Hybrid system dan Sistem Mekanis SM 1.2 Kebutuhan Listrik 700 kW .....	40

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Patroli pengawasan perairan merupakan hal yang penting dalam menjaga kedaulatan perairan, terutama sebagai negara dengan luas perairan yang luas dan kaya sumber daya akan memerlukan patroli pengawasan yang intensif untuk mencegah kapal asing masuk maupun keluar untuk mencuri. Dalam hal patrol disamping persenjataan dan perlengkapan yang baik, hal yang perlu diperhatikan adalah kecepatan kapal. Kecepatan yang tinggi diperlukan untuk pengejaran, agar kapal yang terdapati melaut masuk melewati batas wilayah dapat terkejar. Kondisi patrol normal tidak membutuhkan kecepatan secara signifikan, dan untuk menjaga penggunaan bahan bakar, kapal akan dioperasikan pada kecepatan rendah.

Jenis propulsi kapal semakin beragam sesuai perkembangan teknologi, dengan berbagai pertimbangan yang utamanya merujuk pada kepentingan dan pada penurunan kebutuhan tenaga (bahan bakar). Pada kapal yang menggunakan sistem propulsi mekanis (*Diesel Mechanical Propulsion*) kondisi tersebut membuat kebutuhan operasi yang tinggi. Hal ini disebabkan pengoperasian motor diesel pada beban yang rendah lebih boros bahan bakar namun akan lebih ekonomis untuk penggunaan pada kecepatan penuh, dikarenakan efisiensi untuk motor diesel masih merupakan efisiensi terbaik untuk motor bakar pada kondisi saat ini, sistem ini lebih memberikan keuntungan pada saat kondisi pengejaran.

Pada saat pengoperasian kapal patroli, kecepatan akan diatur sesuai kondisi kegiatan yang dilaksanakan

kapal dan kondisi laut yang akan berubah-ubah pada waktu tersebut, contohnya saat ombak sedang dan tinggi akan memiliki nilai beban kapal yang berbeda. Pengaturan kecepatan merupakan nilai positif untuk sistem penggerak elektrik (*Diesel Electrical Propulsion*) karena dapat diatur dengan daya yang dibutuhkan pada kondisi perairan yang berubah untuk kecepatan yang diinginkan.

Sistem propulsi *Hybrid* dibuat untuk memaksimalkan potensi dari masing-masing jenis penggerak. Untuk perairan yang luas milik Indonesia, penggunaan penggerak akan dipertimbangkan untuk segi ekonomis karena daerah patrol yang luas akan membuat biaya pengoperasian tinggi. Sehingga konsep *hybrid* akan dipakai sebagaimana saat kapal kondisi operasi normal akan menggunakan DEP untuk pengaturan yang baik dan efisiensi pada kecepatan rendah, penggunaan motor elektik akan menggunakan motor DC. Dan penggunaan DMP untuk kondisi pengejaran yang lebih menguntungkan untuk kerja maksimal dan daya besar.

Permasalahan pada sebuah kapal yang menggunakan sistem propulsi *hybrid* adalah penggabungan pembangkitan daya yang optimum pada kondisi beban yang berbeda. Oleh karena itu, pada tugas akhir ini akan membuat *Decision Support System* pengaturan pembangkitan daya pada kapal *fast patrol* 60 m.

## 1.2 Rumusan Masalah

### a. Permasalahan

Penggunaan sistem propulsi *hybrid* diharapkan akan memaksimalkan penggunaan bahan bakar, dan membagi kebutuhan yang tepat akan penggunaan baik DEP maupun DMP, yang mana kombinasi keduanya akan menunjukkan peningkatan efisiensi akan penggunaan bahan bakar pada kapal fast patrol nantinya

Permasalahan yang akan dianalisis adalah proporsi yang tepat untuk menggunakan sistem yang tepat pada propulsi kapal Patroli 60 m.

1. Bagaimana kebutuhan daya untuk setiap kecepatan pada saat patroli?
2. Bagaimana distribusi sumber daya pada setiap kecepatan pada saat patroli?

### b. Batasan Masalah

Pada tugas akhir ini, sistem propulsi *hybrid* yang bekerja sebagai sebagai pada gambar diatas. Dimana untuk memudahkan analisa dilakukan beberapa asumsi sebagai berikut :

1. Pada saat M/E beroperasi, penambahan kebutuhan daya motor listrik diabaikan
2. Motor listrik yang dipergunakan adalah motor DC
3. Perhitungan konstruksi motor listrik diabaikan

### 1.3 Tujuan Penulisan

Tujuan tugas akhir ini adalah untuk mengatur proporsi daya yang tepat dan untuk mendapatkan nilai konsumsi bahan bakar yang terhemat untuk propulsi *hybrid* pada kapal **Fast Patrol Boat 60 M**

### 1.4 Manfaat Penulisan

1. Sistem propulsi *hybrid* akan lebih menguntungkan dalam pengoperasian kapal *fast patrol boat* karena memiliki pengaturan kecepatan sesuai tujuan patrol.
2. Perencanaan secara tepat proporsi dari daya yang dibutuhkan (genset) pada kondisi yang berubah-ubah.
3. Mendapatkan nilai konsumsi bahan bakar minimum untuk pemakaian kapal pada kecepatan yang diinginkan.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Hybrid Propulsion System

*Hybrid propulsion* di dalam perkembangannya merupakan perpaduan antara *Diesel Mechanical Propulsion (DMP)* dengan *Diesel Electric Propulsion (DEP)*. Sistem DMP yang dihubungkan langsung ke poros propeller (atau melalui *reduction-gear*). Dimana DMP mampu memberikan tingkat efisien lebih dari 95%. Sedangkan Diesel Electric Propulsion hanya mampu memberikan efisiensi sebesar 85% - 89% yang sedikit lebih rendah dibanding DMP

Diesel Electric Propulsion (DEP) merupakan sebuah alternatif yang digunakan sebagai sistem penggerak kapal (*ship propulsion*). Electric propulsion memberikan banyak keuntungan yang sangat penting pada aplikasi tertentu walaupun tidak dapat dipisahkan dengan adanya biaya yang tidak murah, mempunyai sistem instalasi yang berat, dan adanya losses yang besar pada sistem transmisinya.

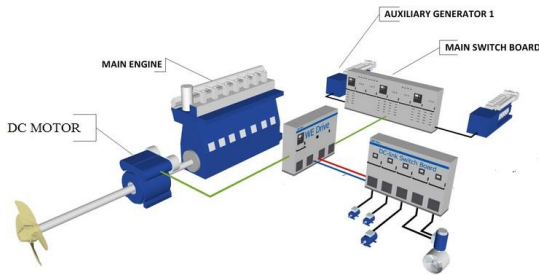
Diesel Electric Propulsion memberikan banyak keuntungan dibandingkan dengan Diesel Mechanical Propulsion (DMP)

diantaranya :

- Kemudahan dan kesesuaian dalam pengaturan kecepatan putaran.
- Sistem pengontrolan,
- Redundancy produksi power engine,
- Fleksibilitas perancangan kamar mesin,
- Efisiensi pemakaian bahan bakar,
- Tingkat kebisingan dan getaran yang rendah,
- Dampak polusi yang lebih rendah.

Pemilihan sistem propulsi adalah salah satu pertimbangan dasar dalam merancang sebuah kapal. Keputusan merancang sistem propulsi DMP ataupun DEP menjadi sangat sulit untuk memuaskan dan mencapai semua kebutuhan secara simultan. Bagaimanapun harus dibuat sebuah design dari sistem yang paling optimal yang memiliki sebanyak mungkin faktor-faktor yang menguntungkan (Wahyudi , Dedy. 2010).

Desain dari sistem propulsi *hybrid* ini bermacam macam salah satunya, seperti Gambar 2.1 berikut :

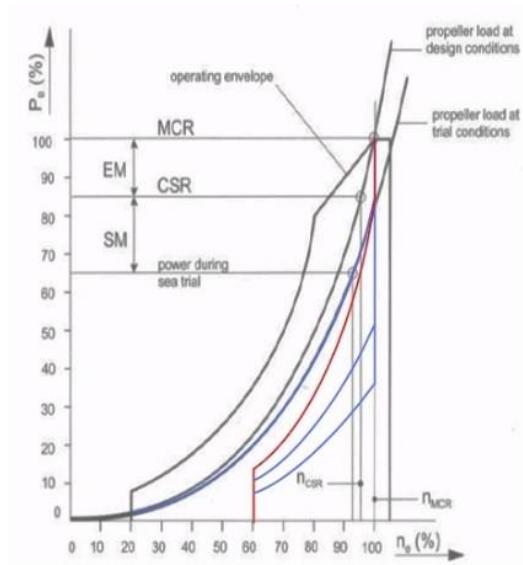


Gambar 2.1. desain propulsi hybrid

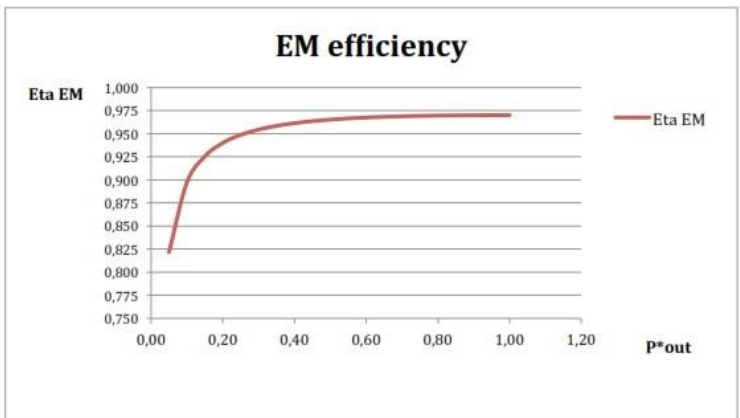
### 2.1.1. Motor Diesel

Motor diesel adalah salah satu motor bakar dalam yang memiliki tingkat efisiensi tenaga tertinggi dibandingkan dengan motor bakar lain, yang semakin naik untuk setiap persentase tenaga yang dipakai seperti gambar 2.2 mengenai efisiensi dengan tenaga dan 2.3 tentang konsumsi bahan bakar:





Gambar 2.2. Grafik performa motor diesel (Kwasieckyj, 2013)



Gambar 2.3 Grafik konsumsi bahan bakar tunggal pada motor diesel L48/60 CR

Hal ini dikarenakan sistem kerja motor diesel dengan menggantungkan kompresi udara, semakin tinggi persentase tenaga yang digunakan maka tekanan dari kompresi akan semakin maksimal, sehingga tingkat efisiensi naik.

Motor diesel banyak dipakai di dunia perkapalan pada kapal komersil, karena kapal komersil lebih cenderung berkecepatan tetap atau dengan penggunaan kapal kecepatan tinggi yang memiliki persentase sangat tinggi, yaitu pada saat berlayar dilautan dengan memanfaatkan efisiensi tertinggi atau seperti yang ditunjukkan gambar 2.2 pada titik 85% tenaga.

### 2.1.2 Motor Listrik

Penggunaan motor listrik banyak dipakai pada kapal yang bergerak dengan variasi kecepatan, dikarenakan pengaturan kecepatan yang mudah dan lebih efisien pada tenaga rendah jika dibandingkan dengan motor diesel.

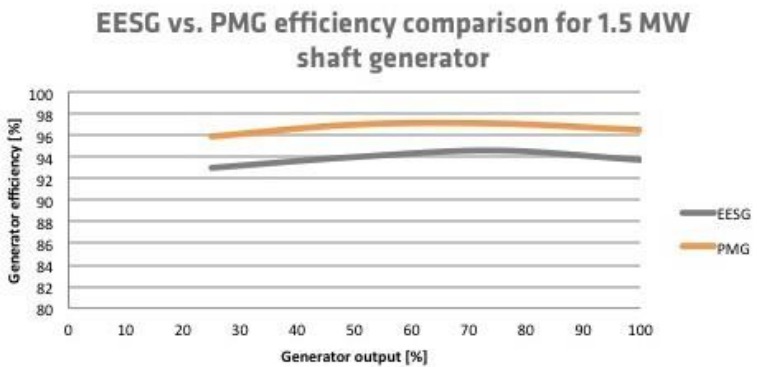
Kemudahan dan kesesuaian dalam pengaturan kecepatan putaran propeller beserta arah putarannya dapat diatur pada *electric drive* merupakan kelebihan dari sistem ini. Sistem *electric* digunakan karena kemudahan dalam pengontrolan dapat diatur walaupun dalam jarak yang cukup jauh, sehingga dapat secara langsung diatur oleh operator.

Koneksi elektrik antara generator dan motor listrik dapat dirancang dan diinstalasi lebih bebas, tidak seperti pada sistem penggerak mekanik, penggerak utama beserta generatornya dapat diletakkan dimanapun diminta dan berbentuk secara mekanis berupa engine generator set. Propulsion genset dapat digunakan sebagai sumber power untuk peralatan yang lain apabila tidak digunakan untuk menggerakkan

propulsor, antara lain digunakan untuk *fire fighting*, *refrigerating*, *machine gun*, *rocket launcher*, dan lain-lain yang berupa supplemental service. (Ega W, 2007).

### 2.3.1. Permanent Magnet Generator

Permanent Magnet Generator adalah sistem generator yang menggunakan motor DC dengan magnet permanen pada rotor untuk mengurangi rugi untuk menghasilkan medan daya. sehingga mengurangi nilai pemakaian listrik untuk eksitasi pada motor/generator sinkron maupun asinkron pada umumnya. Nilai efisiensi akan ditunjukkan pada Gambar 2.4 berikut



Gambar 2.4 perbandingan efisiensi Permanent Magnet Generator terhadap Eletrically Excited Synchronous Generator (*the switch*, 2006)

### 2.1.3 Shaft Generator System

Sistem ini memanfaatkan kinerja motor diesel untuk menompang beban dari generator set yang biasa dipakai dengan motor listrik yang tersambung dengan poros motor diesel, karena sebagaimana prinsipnya motor listrik dapat bekerja pula sebagai generator yang

merujuk pada kesamaan konstruksi. (Kwasiieckyj, 2013).

Shaft generator ini menjadi pilihan jika ingin menghemat bahan bakar dengan memanfaatkan kinerja motor diesel pada posisi penuh namun daya dorong yang dibutuhkan untuk kecepatan yang diinginkan memiliki rentang yang cukup tinggi, dan cukup untuk memenuhi kebutuhan listrik akomodasi kapal tersebut.

#### 2.1.3.1 Power Take In

*Power take in* atau PTI adalah system masuknya daya ke sistem propulsi kapal dari genset melalui motor listrik yang terhubung pada shaft, sistem ini digunakan selain untuk menggerakkan propulsi secara mandiri, juga dapat digunakan sebagai sistem *boosting* untuk penambah daya pada penggerak diesel demi menambah kecepatan tanpa memasang mesin dengan daya yang lebih besar.

#### 2.1.3.2 Power Take Out

*Power take out* atau PTO, berkebalikan dengan PTI, sistem ini memanfaatkan daya berlebih dari mesin diesel kepada generator untuk membantu suplai listrik di kapal. Dikarenakan mesin diesel akan memiliki efisiensi bahan bakar tinggi saat pada daya 85% (Kwasiieckyj, 2013) maka sistem ini dipilih saat kapal pada kecepatan yang membutuhkan mesin diesel daya rendah, untuk memaksimalkan efisiensi daya dari mesin diesel dioperasikan pada daya tinggi yang sebagian dayanya di konversi ke daya listrik.

## 2.2 Decision Support System (DSS)

Pengertian DSS dalam hal manajemen pembangkitan daya di kapal ini adalah suatu program computer yang bertujuan untuk membantu menganalisa suatu kebutuhan kapal dalam hal pembangkitan daya untuk penggerak utama dan sebagai pembangkit daya penunjang operasional kapal, serta menyediakan informasi yang bersifat sebagai pertimbangan dalam pengambilan keputusan.

DSS bukanlah sebagai pengambil keputusan, namun hanya sebagai alat bantu untuk nahkoda dalam mengambil keputusan yang dianggap terbaik dalam kondisi pelayaran yang diinginkan dengan pertimbangan ekonomi pemakaian bahan bakar. Secara fundamental terdapat 3 komponen dari DSS seperti pada Gambar 2.5 yaitu:

### 2.2.1 Database Management System (DBMS)

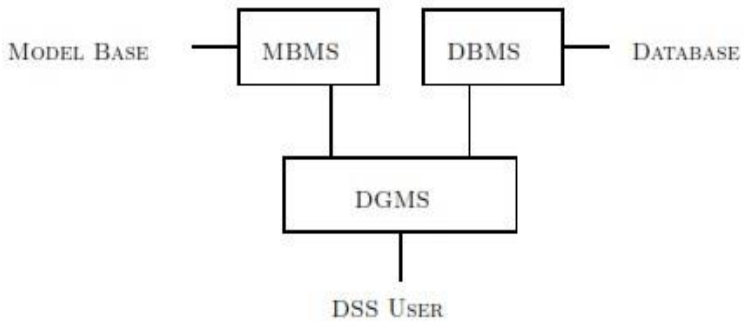
DBMS berisi tentang data-data penting yang diperlukan sebagai perhitungan dan pertimbangan dalam DSS untuk penyusunan desain dan logika yang akan dimasukkan dan berinteraksi secara tersusun dan berurutan dengan interaksi yang dilakukan oleh pengguna. DBMS menghubungkan langsung dan melewati berbagai aspek fisik dan proses perhitungan yang rumit dan menunjukkan data-data yang tersedia kepada pengguna.

### 2.2.2 Model-based Management System (MBMS)

Berbeda dengan DBMS, MBMS merupakan data atau proses yang terdapat secara independen dan tidak bergantung pada situasi dan data yang diperlukan, fungsi dari MBMS adalah merubah data dari DBMS menjadi proses yang akan menghasilkan informasi yang menjadi pertimbangan dalam pengambilan keputusan.

### 2.2.3 Dialog Generation and Management System (DGMS)

Merupakan produk utama dari DSS itu sendiri, dimana DGMS akan menunjukkan hasil akhir yang biasanya menjadi pertimbangan utama dari penggunaan DSS terhadap sistem yang dianalisa. DGMS akan ditampilkan kepada pengguna dalam bentuk interface yang memudahkan pengguna untuk memberikan data memperoleh hasil yang dicari (Marek J.D dan R.Roger, 2002).



Gambar 2.5 Struktur Decision Support System

## **BAB III METODOLOGI PENELITIAN**

Metode yang digunakan untuk menyelesaikan permasalahan diatas adalah dengan perumusan dan deskripsi melalui program simulasi. Dengan menggunakan program Matlab untuk menguraikan permasalahan dan analisa perhitungan. Tahapan-tahapan dari pengerjaan tersebut diperlihatkan pada gambar 3.1 adalah sebagaimana berikut :

1. Studi Literatur dan Pengumpulan Data
2. Penyusunan Perencanaan Program Simulasi
3. Pembuatan Program Simulasi
4. Analisa Program Simulasi
5. Kesimpulan

### 3.1 Studi Literatur dan Pengumpulan Data

Study Literatur dilakukan dengan mencari bahan referensi mengenai propulsi hybrid (DMP +DEP), perhitungan tahanan dan *engine propeller matching* dan penyusunan program simulasi menggunakan program Matlab, serta itu mengambil beberapa sumber lain yang mendukung dengan sumber :

1. Buku/E-book
2. Paper
3. Internet
4. Laporan Tugas Akhir

Data-data yang diperlukan adalah data fast patrol boat 60 m, data mengenai diesel engine, generator dan motor listrik yang dipakai.

### 3.2 Penyusunan Perencanaan Program Simulasi

Pembuatan program simulasi ini akan menggunakan software MatLab dengan menyusun flowchart simulasi dan dilanjutkan dengan penyusunan program sesuai kondisi yang diinginkan untuk analisa. Hasil program akan mencakup kebutuhan setiap kecepatan dengan sea-margin yang berbeda, selain itu untuk menentukan penggunaan sumber tenaga. Penyusunan ini digunakan untuk menyusun secara runtut dan jelas akan proses simulasi ini akan dilakukan.

### 3.3 Pembuatan Program Simulasi

Setelah penyusunan dari perencanaan program, maka akan disusun program untuk menunjukkan kebutuhan daya pada setiap kecepatan dan sea margin, baik kebutuhan daya dorong kapal maupun daya generator untuk menopang kebutuhan listrik di kapal.

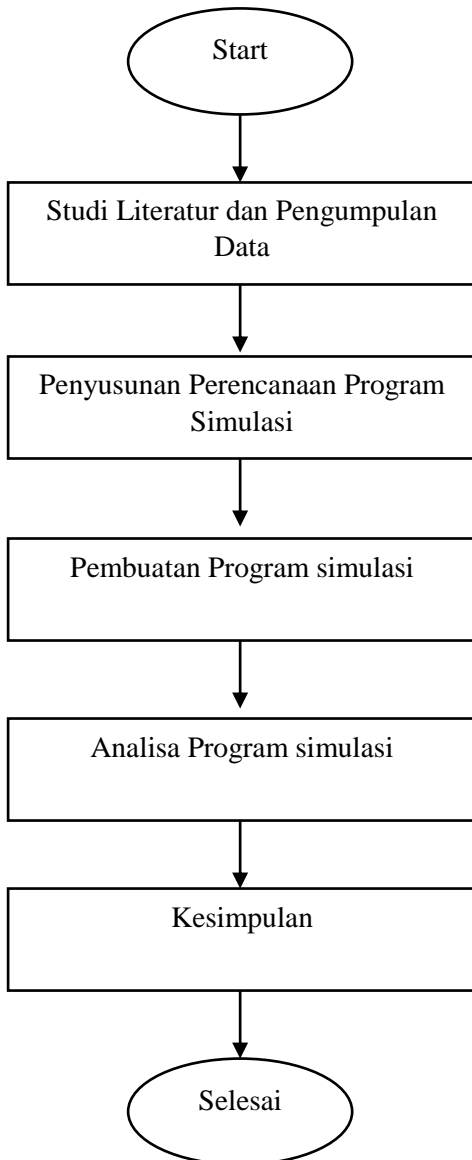
### 3.4 Analisa Program

Analisa akan dilakukan dengan menunjukkan hasil dari kebutuhan daya yang dipakai pada setiap sistem dan konsumsi bahan bakar untuk sumplai. Pada proses ini akan menunjukkan pilihan terbaik dalam hal konsumsi bahan bakar, *load factor* dari generator yang digunakan atau hanya perlu menggunakan Diesel sebagai penompang kerja dan suplai listrik menggunakan sistem *shaft generator*

### 3.5 Kesimpulan

Kesimpulan yang diambil adalah pemakaian sistem terbaik pada setiap kondisi yang di simpulkan dari pemakaian bahan bakar terendah.





Gambar 3.1 Diagram Alur Metodologi penelitian

*“Halaman ini sengaja di kosongkan”*

## BAB IV

### ANALISA DATA DAN PEMBAHASAN

#### 4.1 Umum

Pada bab ini akan dikemukakan tentang data-data teknis Fast Patrol Boat 60 m yang menjadi obyek pembahasan dalam penyusunan tugas akhir ini dan melakukan perhitungan ulang untuk terhadap penggunaan sistem propulsi hybrid shaft generator pada FPB 60m. Selanjutnya disampaikan alur proses penyusunan program simulasi pengaturan pembangkitan daya untuk kapal FPB 60 m yang menggunakan sistem propulsi hybrid shaft generator. Penyusunan program ini diharapkan dapat menjadi acuan bagi kapten kapal dalam mengendalikan kapal FPB 60 m dengan efisiensi yang tinggi.

Kemudian dijelaskan pula analisa terhadap hasil keluaran dari program yang telah disusun, dimana sebagai data masukkan berupa nilai kecepatan kapal, daya kebutuhan listrik kapal, dan nilai *service/sea margin*. Dengan menunjukkan hasil analisa proses perhitungan berupa daya yang diperlukan serta putaran kerja engine, beserta hasil yang dirujuk berupa konsumsi bahan bakar untuk sistem kerja yang dipakai.

#### 4.2 Data Kapal

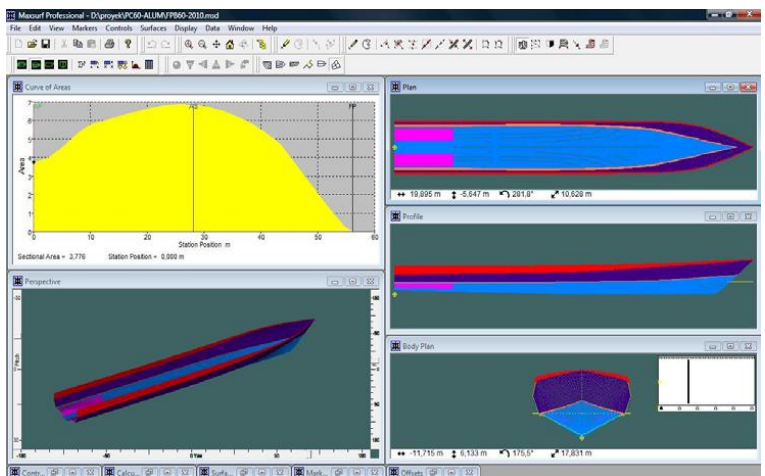
FPB 60 m merupakan kapal cepat berpengerak propulsi sendiri yang digunakan untuk patroli sehingga konstruksi kapal ini memiliki lambung berbentuk V seperti yang ditunjukkan Gambar 4.1 . Data FPB 60 yang digunakan untuk dalam penyusunan program

simulasi pembangkitan daya pada kapal Fast Patrol adalah sebagai berikut :

#### 4.2.1 Data Umum Kapal

Tabel 4.1 Data ukuran FSB 60 m

Length Over All (LOA)	60,32	m
Length Water Line (LWL)	55,82	m
Breadth	8,37	m
Height	5,09	m
Draft	2,00	m
Displacement	291,5	ton



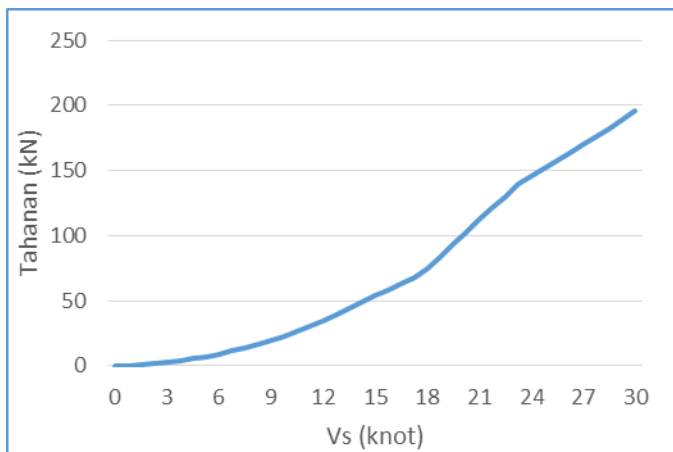
Gambar 4.1 Desain Lines Plan FPB 60 M menggunakan software Maxsurf

Dari data yang diproses oleh maxsurf diperoleh gambaran umum dari bentuk badan kapal, *displacement*, *koefisien prismatic*, panjang garis air (LWL), koefisien permukaan basah dan data umum lainnya, walaupun

tidak menunjukkan desain dan bangunan atas kapal, namun gambar dan simulasi cukup untuk memberikan data tahanan dan dan kemungkinan trim yang diberikan saat kapal pada kecepatan tertentu, dari data diatas dapat pula diberikan sebagai input untuk perhitungan nilai tahanan untuk kondisi fluida yang ditentukan, nilai tahanan dan daya akan tersedia dalam bentuk Metode Savisky dan Metode Holtrop. Savisky sendiri diperuntukkan untuk kapal-kapal cepat dan Metode Holtrop digunakan untuk kapal dengan nilai *Displacement* besar. Sehingga untuk perencanaan kali ini akan digunakan Metode Savisky sebagai pedoman nilai tahanan, meskipun memiliki nilai daya mesin, namun nilai tersebut cenderung tidak memiliki kesamaan dengan hitungan selanjutnya.

#### 4.2.2 Data Tahanan Kapal

Dengan desain yang dimasukkan kedalam software maxsurf sesuai data umum, maka didapat nilai tahanan kapal seperti pada Gambar 4.2 untuk FPB 60 m berdasarkan variasi kecepatan.



Gambar 4.2 Grafik Tahanan dinas FPB 60 M

Kapal Fast Patrol Boat akan disusun dengan kondisi operasi di laut yang seperti pada Tabel 4.2

Tabel 4.2 Tabel Operasi Fast Patrol Boat 60 M dengan SM 1.1

Operasi Kapal	pengintaian	patroli	pengejaran
kecepatan (knot)	17	21	29
BHP (kW)(x2)	492	1069	2276
RPM engine	1117	1424	1875

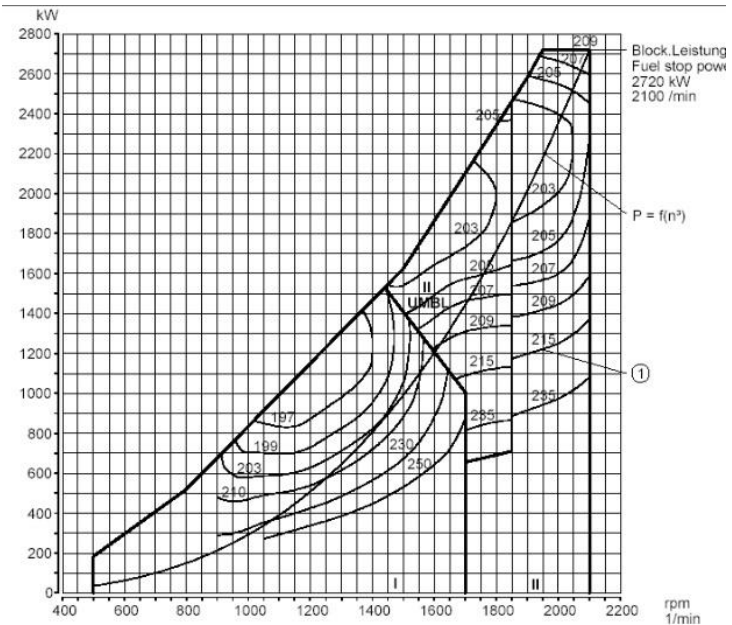
Dengan pertimbangan desain perencanaan dan data kapal diatas dan pertimbangan untuk factor efisiensi dari diesel engine senilai 90% ( $2720 \times 0.9 = 2448$  kW) dan untuk memenuhi putaran yang tepat dengan perhitungan, serta untuk pertimbangan motor listrik yang dapat memenuhi kebutuhan maksimal dari akomodasi listrik maka akan dipilih sistem permesinan untuk kapal fast patrol boat sebagai berikut pada tabel 4.3

Tabel 4.3 Tabel Komponen Fast Patrol Boat 60 M

no	Komponen	Merk
1	Main Diesel	MTU 16V 4000 M90 2720/2100 kW/rpm 2 buah
2	Motor Listrik	PMM/G the Switch 550/1500 kW/rpm 2 buah
3	Generator/ Auxillary Engine	Scania genset sg600 DC 1649A 440 kW 4 (+ 1) buah
4	Converter	PFC the Switch 1000 kW 2 buah
5	Gearbox	ZF 7600 1:1.486 4 buah
6	Propeller	B5-90 HS-series 1.308 m 2 buah

### 4.3 Analisa Rencana Permesinan dengan Sistem Hybrid

Analisa untuk mesin utama akan dilakukan sesuai *engine envelope* yaitu pada MTU 16v 4000 M90 dengan menyetarakan nilai konsumsi bahan bakar pada setiap kondisi kerja mesin. Karena pada mesin diesel nilai konsumsi bahan bakar (SFOC) akan bervariasi tergantung daya dan putaran yang dikeluarkan, performa dari mesin diesel dapat dilihat pada gambar berikut:

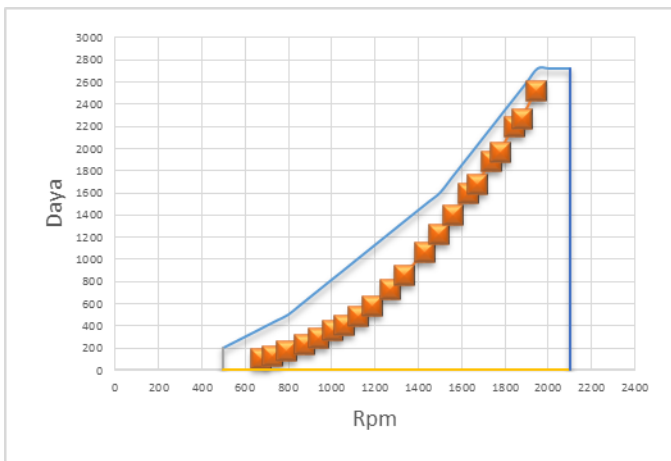


Gambar. 4.3 Performa Mesin Diesel (MTU 16V 4000 90M)

Gambar 4.3 menunjukkan titik-titik kerja maksimal untuk memperoleh nilai konsumsi bahan bakar

minimal. Dengan menaikkan tingkat daya pada titik putaran yang sama, yaitu dengan menambah beban mesin diesel sebelum disalurkan ke propeller dengan cara menambahkan *shaft generator* dengan sistem PTO untuk dimanfaatkan sebagai penompang kebutuhan listrik akomodasi, diharapkan dengan pemakaian shaft generator dapat mengganti pemakaian generator dari *auxillary engine* dan mendapatkan konsumsi bahan bakar yang lebih rendah dibandingkan memakai model mekanis.

Dengan beberapa data diatas akan dilakukan *matching* pada propulsi FPB 60, di rencanakan dengan mesin diesel MTU 16V 4000 90M akan mampu untuk memenuhi kebutuhan hingga kecepatan 29 knot. Yang mana dipasang pula gear box ZF 7600 dengan rasio 1:486 pada lokasi setelah poros mesin keluar atau sebelum terhubung dengan motor listrik dan setelah motor listrik untuk penyesuaian putaran propeller yang tepat, perhitungan dilakukan pada service margin sebesar 1.1



Gambar 4.4 Hasil *engine propeller matching* pada kecepatan 10-30 knot dengan SM 1.1

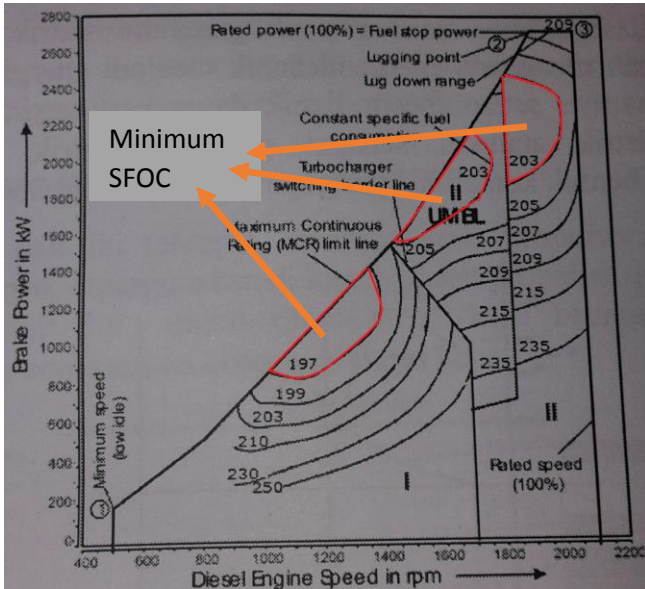


Gambar 4.4 menunjukkan grafik daya yang dibutuhkan untuk memperoleh kecepatan dari 10 knot hingga 30 knot pada SM 1.1 dengan penggunaan motor diesel MTU 16V 4000 90M sebagaimana nilai tersebut masuk kedalam *engine envelope*.

Penambahan motor listrik yang sekaligus dapat bertindak sebagai generator saat digerakkan dengan arus DC dan dalam konstruksinya tersusun oleh magnet permanen sebagai penompang agar tidak diperlukan arus eksitasi atau beban tambahan untuk medan magnet pada saat beroperasi. Pemilihan dilakukan sengan daya output spesifikasi 550 kW berjumlah 2 buah.

Penambahan permanen magnet motor/generator inilah yang menjadikan sistem ini akan menghasilkan tenaga yang sesuai dan dengan konsumsi bahan bakar minimum dengan output masing-masing motor hingga 550x90% yaitu 495 kW, cukup untuk mendukung kebutuhan kecepatan hingga 17 knot pada SM 1.1.

Sistem selanjutnya adalah penggunaan shaft generator untuk memanfaatkan tenaga mesin diesel (*Main Engine*) untuk menggerakkan motor listrik sebagai penyuplai sumber daya listrik untuk kebutuhan akomodasi kapal, selain itu untuk mengarahkan kearah konsumsi bahan bakar (SFOC) yang lebih rendah, yang bertujuan untuk mendapat nilai konsumsi bahan bakar total lebih kecil pada daerah yang ditunjukkan gambar 4.5. Nilai SFOC yang rendah cenderung berada pada daerah dengan daya yang maksimal pada *engine envelope* di daerah putaran yang sama, penunjukan tersebut sama dengan karakteristik dari motor disel yang bekerja maksimal pada daya maksimal, sehingga tujuan dari shaft generator adalah mengarahkan operasi ke atas menuju ke angka SFOC yang terendah tersebut

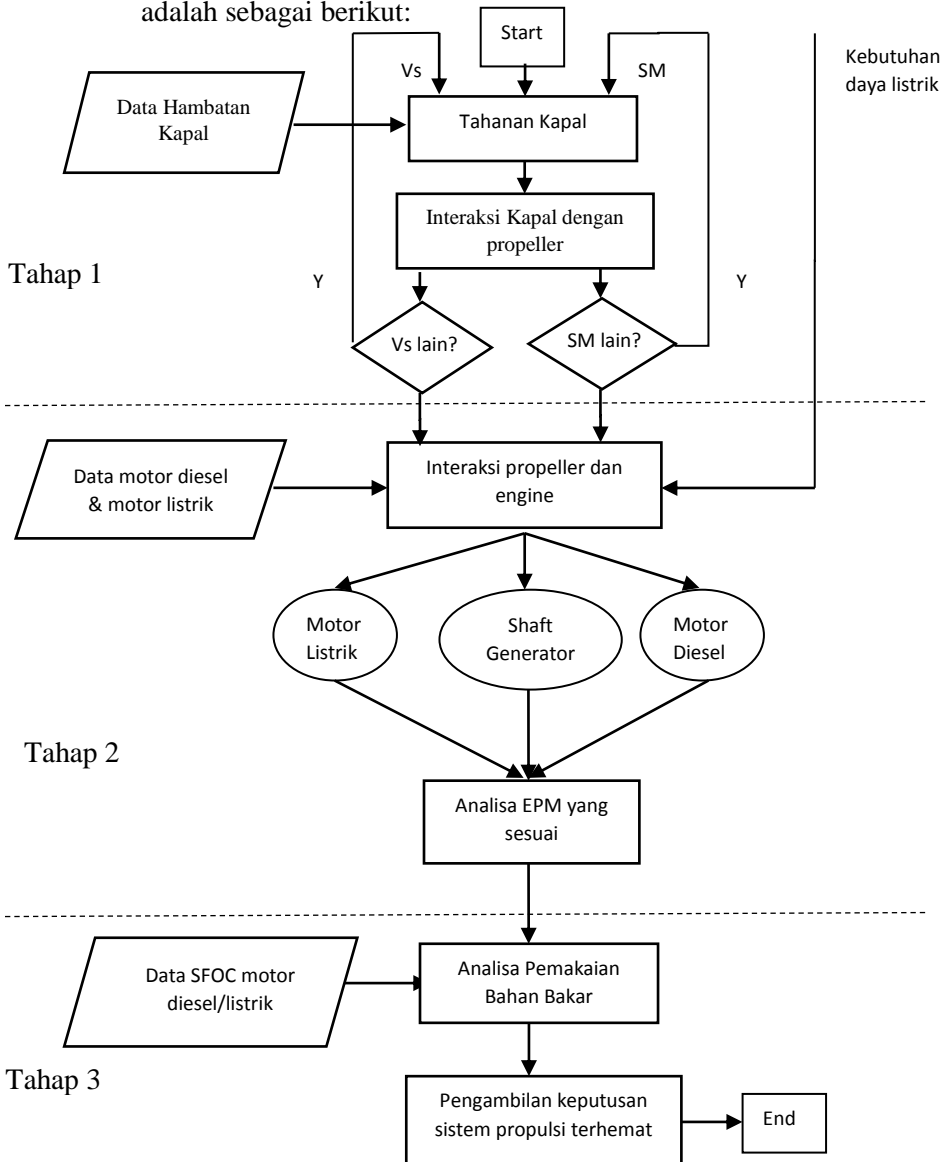


Gambar 4.5 Performa dan SFOC MTU 16V 4000 90M

#### 4.4 Perencanaan Pengerjaan Simulasi

Perencanaan ini menunjukkan proses yang akan diambil dalam penyusunan perencanaan pembangkitan daya,, tahapan sistem propulsi yang akan diambil berdasarkan langkah penyesuaian kapal, propeller dan engine yang dipakai dalam penyusunan sistem propulsi hybrid.

Langkah penyusunan pengaturan pembangkitan daya adalah sebagai berikut:



Gambar 4.6 Gambar Langkah Kerja Simulasi

Desain proses simulasi berikut adalah memberikan nilai input yang diperlukan untuk perhitungan mulai dari kecepatan yang diinginkan, data service margin yang dipakai kapal saat berlayar dan jumlah kebutuhan daya listrik pada kondisi tersebut. Dengan adanya masukan nilai kecepatan akan didapatkan nilai tahanan kapal saat perairan tenang, jika nilai kecepatan tidak terhitung bulat pun akan didapatkan dengan metode interpolasi, selanjutnya dengan nilai *service margin* (SM) dipakai nilai tahanan kapal yang dipakai pada kondisi tersebut, dari kondisi ini sebenarnya sudah dapat diproses ke tahap selanjutnya, yaitu perhitungan kebutuhan daya untuk sistem mekanis dan elektrik, namun untuk memenuhi atau mengetahui pada sistem *shaft generator* perlu ditambahkan input ketiga yaitu nilai kebutuhan listrik akomodasi kapal. Dengan nilai tambahan kebutuhan daya listrik maka daya kebutuhan motor diesel untuk memenuhi sistem shaft generator dapat diketahui dan untuk metode mekanis dan elektrik dapat diketahui daya penggunaan generator atau *auxillary engine* yang dipakai.

#### **4.5 Desain Interface DSS FPB 60 M**

Bagian akhir mengenai DSS adalah desain interface yang sederhana dan mudah untuk dimengerti dan dioperasikan. Desain dari DSS tersedia input dan output yang dicantumkan dalam satu halaman dan tertata secara berurutan agar memudahkan pembedaan, sehingga membantu pengguna memahami langkah kerja DSS ini.

Dengan pencantuman untuk ketersediaan sistem setelah ke 3 input di masukkan dan di proses, setelah itu dapat di pilih proses berikutnya untuk mengetahui

jumlah konsumsi bahan bakar untuk sistem yang diinginkan serta daya yang dioperasikan oleh genset.

- Mode Elektris

Adalah mode dimana FPB digerakkan oleh motor listrik dan dengan daya yang diplai sepenuhnya oleh genset, nilai dari konsumsi bahan bakar akan merujuk kepada project guide pada genset yang dipakai

- Mode Shaft generator

Adalah mode dimana FPB akan bergerak menggunakan mesin diesel dan daya tambahan akan disuplai ke motor untuk menghasilkan listrik sebagai penompang kebutuhan akomodasi

- Mode Mekanis

Adalah mode paling konvensional yaitu menggunakan mesin diesel untuk menggerakkan FPB dan genset untuk menyuplai kebutuhan listrik, biasanya beroperasi untuk kecepatan tinggi.

The screenshot shows a software interface for a simulation. The window title is "simulasi". It is organized into three main sections:

- input:** Contains three input fields: "Kecepatan Kapal" (17 knot), "Sea Margin" (1.1), and "Kebutuhan Listrik" (500 kWh). A "proses" button is located below these fields.
- analisa daya:** This section is divided into three rows:
  - Daya Motor Listrik:** Shows 482.304 kWh and 751.483 RPM. A "Push Button" is next to the RPM value. The "FOC total" is 307.515 kg/h.
  - Diesel + SG:** Shows 788.334 kWh and 1116.7 RPM. A "Push Button" is next to the RPM value. The "FOC total" field is empty.
  - Daya Diesel:** Shows 492.147 kWh and 1116.7 RPM. A "Push Button" is next to the RPM value. The "FOC total" field is empty.
- daya generator:** Shows four "Genset" (1, 2, 3, 4) each with a value of 366.152 kWh. A "Reset" button is located to the right of this section.

Gambar 4.7 Gambar Interface Program Simulasi untuk Mode Listrik

Gambar 4.7 adalah interface yang ditunjukkan dengan nilai 3 input dan setelah diproses dengan kecepatan 17, servise margin 1.1 dan kebutuhan listrik 500 kWh mendapatkan nilai-nilai daya mesin pada masing-masing sistem dan putarannya sebelum dilanjutkan ke gearbox dan e propeller, hasil diatas dilaksanakan pada sistem listrik yang diproses untuk mendapatkan nilai kebutuhan bahan bakar dan jumlah atau daya yang diperlukan pada *auxillary engine* atau genset untuk menompangnya dengan menekan tombol di samping kanan dari daya dan putaran sistem listrik tersebut.

The screenshot shows a software interface for a simulation. It is titled "simulasi" and has a standard window control bar. The interface is organized into three main sections:

- input:** Contains three input fields: "Kecepatan Kapal" with the value 17 and unit "knot", "Sea Margin" with the value 1.1, and "Kebutuhan Listrik" with the value 500 and unit "kWh". A "proses" button is located below these fields.
- analisa daya:** This section is divided into three rows of data:
  - Daya Motor Listrik:** Input fields for "Daya Motor Listrik" (482.304 kWh) and "RPM" (751.483). A "Push Button" is to the right, followed by an empty "FOC total" field with unit "kg/h".
  - Diesel + SG:** Input fields for "Diesel + SG" (788.334 kWh) and "RPM" (1116.7). A "Push Button" is to the right, followed by an empty "FOC total" field with unit "kg/h".
  - Daya Diesel:** Input fields for "Daya Diesel" (492.147 kWh) and "RPM" (1116.7). A "Push Button" is to the right, followed by an empty "FOC total" field with unit "kg/h".
- daya generator:** Contains four input fields for "Genset 1", "Genset 2", "Genset 3", and "Genset 4", all with the value 0 and unit "kWh". A "Reset" button is located to the right of these fields.

Gambar 4.8 Gambar Interface Program Simulasi untuk Mode Shaft Generator

Gambar 4.8 adalah interface yang ditunjukkan dengan nilai 3 input dan setelah diproses dengan kecepatan 17, servise margin 1.1 dan kebutuhan listrik 500 kWh mendapatkan nilai-nilai daya mesin pada masing-masing sistem dan putarannya sebelum dilanjutkan ke gearbox dan ke propeller, hasil diatas dilaksanakan pada sistem shaft generator yang diproses untuk mendapatkan nilai kebutuhan bahan bakar dan jumlah atau daya yang diperlukan pada *auxillary engine* atau genset untuk menompangnya, namun dengan pemakaian shaft generator tersebut membuat kebutuhan listrik untuk di tamping *auxillary engine* menjadi bernilai nol.

The screenshot shows a software interface for a simulation. The window title is "simulasi".

**input**

Kecepatan Kapal	Sea Margin	Kebutuhan Listrik
17 knot	1.1	500 kWh

proses

**analisa daya**

Daya Motor Listrik	RPM	FOC total
482.304 kWh	751.483	kg/h
Diesel + SG	RPM	FOC total
788.334 kWh	1116.7	kg/h
Daya Diesel	RPM	FOC total
492.147 kWh	1116.7	417.271 kg/h

**daya generator**

Genset 1	Genset 2	Genset 3	Genset 4
263.158 kWh	263.158 kWh	0 kWh	0 kWh

Reset

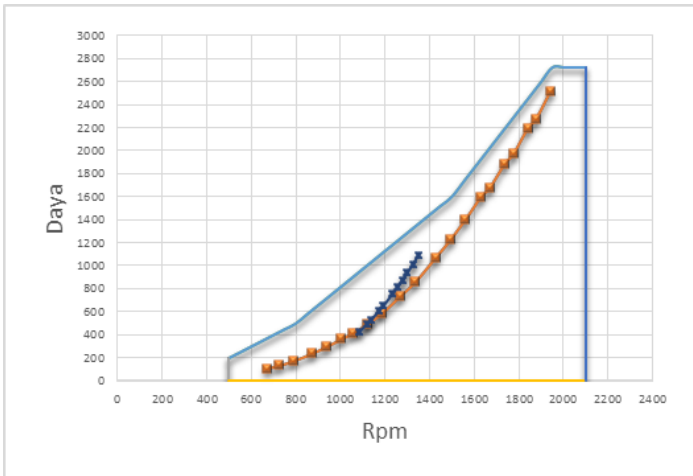
Gambar 4.9 Gambar Interface Program Simulasi untuk Mode Mekanis

Gambar 4.9 adalah interface yang ditunjukkan dengan nilai 3 input dan setelah diproses dengan kecepatan 17, servise margin 1.1 dan kebutuhan listrik 500 kWh mendapatkan nilai-nilai daya mesin pada masing-masing sistem dan putarannya sebelum dilanjutkan ke gearbox dan ke propeller, hasil diatas dilaksanakan pada sistem mekanis yang diproses untuk mendapatkan nilai kebutuhan bahan bakar dan jumlah atau daya yang diperlukan pada *auxillary engine* atau genset untuk menompangnya dengan menekan tombol di samping kanan dari daya dan putaran sistem elektris tersebut.



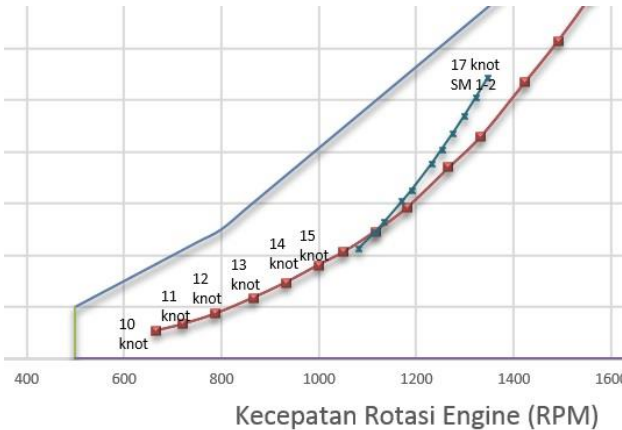
#### 4.6 Analisa EPM pada Hasil Program Simulasi

Pembahasan hasil simulasi berhubungan dengan tujuan kapal untuk patrol pada kecepatan 17 knot, dimana diasumsikan bahwa keadaan laut tidak selalu tenang. Analisa EPM menjadi semakin sulit dengan adanya perubahan keadaan-keadaan yang ada di laut selama kapal beroperasi, yang menyebabkan terjadinya perubahan hambatan kapal. Perubahan tersebut disebabkan oleh timbulnya fouling pada badan kapal propeller, perubahan kondisi cuaca pelayaran, perubahan displacement kapal dan kedalaman laut. Perbandingan antara perubahan tahanan yang disebabkan alas-analasan diatas dengan kondisi kapal baru dan pada keadaan *calm water* disebut dengan nilai SM atau *service margin* (Woud dan Stapersma, 2002). Nilai SM adalah salah satu input utama dalam program simulasi nantinya. Berikut contoh pengaruh SM terhadap EPM ditunjukkan oleh Gambar 4.10 :



Gambar 4.10 Grafik daya dan putaran pada kecepatan 17 knot dengan SM 1-2 dibanding kecepatan 10-30

Pada Gambar 4.11 penilaian dari nilai daya yang dibutuhkan setiap perubahan SM akan berbeda dengan kenaikan nilai kecepatan, dimana pada grafik diatas ditunjukkan dengan garis biru sedangkan untuk perbandingan daya yang disebabkan oleh kenaikan kecepatan ditunjukkan oleh garis merah.



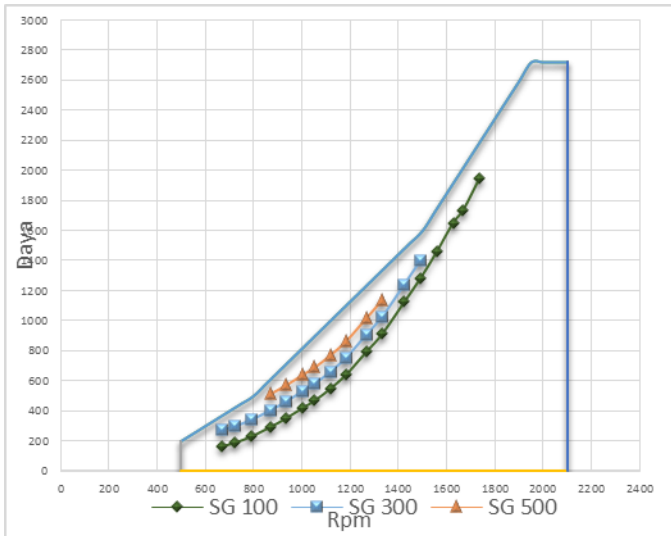
Gambar 4.11 Gambar perbandingan daya dan putaran dari perubahan SM dan kecepatan

Hasil berikut menunjukkan bahwa walaupun kecepatan dan SM akan berpengaruh terhadap perubahan tahanan, namun keduanya merupakan nilai masukan yang harus di pertimbangkann secara independen sehingga menjadikan nilai kecepatan dan SM menjadi salah satu input dari program simulasi.

Daya listrik akan diperlukan untuk memenuhi kebutuhan akomodasi dan penunjang pada operasi penggerak mesin diesel sendiri, jumlah bervariasi tergantung kebutuhan, sehingga nilai input dapat dimasukkan secara bebas. Factor daya listrik akan berpengaruh dalam pemilihan sistem pada penyusunan program simulasi pengaturan pembangkitan daya, terutama untuk masukan nilai kerja generator atau sebagai pengganti dalam sistem hybrid yaitu pemakaian *shaft generator*.

Pemakaian shaft generator dengan mode PTO untuk mensuplai daya listrik bergantung pada besar output shaft generator yang dipasang dan dengan pertimbangan daya masukan dari mesin diesel agar sesuai dengan engine envelope, dan tidak melalui garis kerja yang ada, jika melebihi garis yang disediakan akan dipastikan bahwa daya yang diperlukan untuk suplai PTO dan propulsi tidak dapat dipenuhi.

Shaft generator akan menyebabkan output daya dinaikkan pada putaran yang tetap namun dengan output menuju propeller sama seperti sebelumnya. Dengan ini penggunaan PTO akan menaikkan mengarahkan kerja mesin menuju ke titik hemat daya, pada beberapa titik tertentu. Sebagai contoh untuk operasi yang ditunjukkan pada Gambar 4.12 , dengan menggunakan PTO pada kecepatan 10-30 knot dan pada SM 1.1. dianalisa dengan kebutuhan listrik yang di suplai oleh shaft generator pada daya 100 kW, 300 kW dan 500 kW



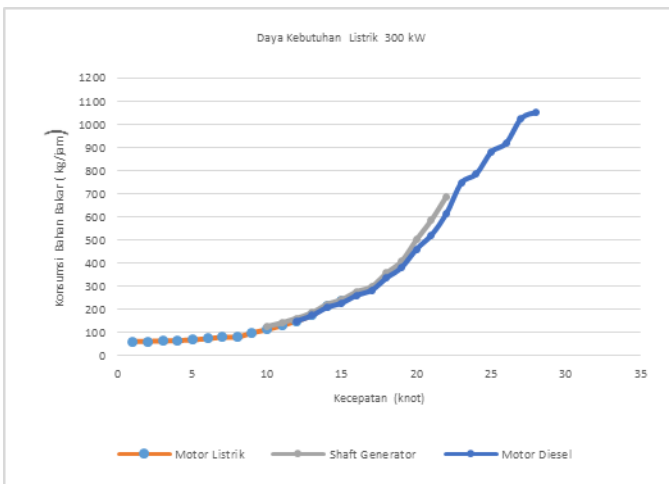
Gambar 4.12 Analisa pemakaian Shaft Generator pada kebutuhan listrik 100 kW, 300 kW dan 500 kW pada SM 1.1

Dengan menggunakan shaft generator untuk menyalurkan daya ke kebutuhan listrik kapal, dengan memperhatikan engine envelope mesin diesel MTU 16M 4000 90M menunjukkan ketersediaan yang berbeda bagi setiap kecepatan maupun kebutuhan listrik. Untuk kebutuhan listrik 100 kW tercatat dapat digunakan hingga kecepatan 26 knot, sedangkan untuk daya listrik 300 kW hanya dari kecepatan 10 knot mencapai 22 knot, dan untuk 500 kW hanya mencakup kecepatan 13 hingga 20 knot dengan catatan untuk batas envelope adalah 90% daya atau batasan maksimal.

Dengan ini disimpulkan pula bahwa kebutuhan listrik memang merupakan salah satu input utama dari perencanaan pembangkitan daya.

#### 4.7 Analisa Bahan Bakar Simulasi

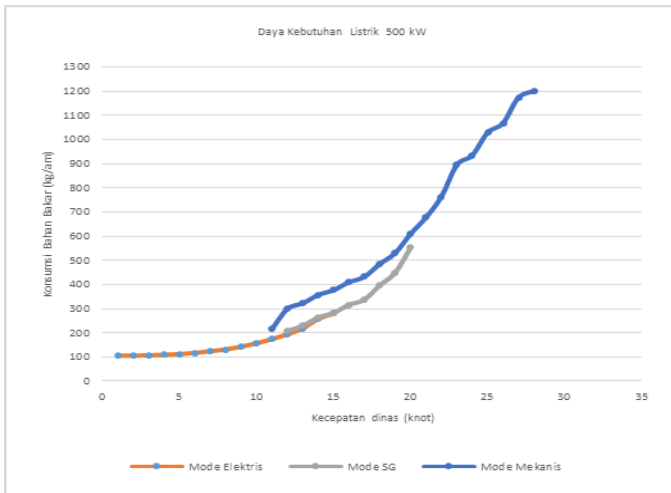
Konsumsi bahan bakar merupakan aspek tujuan yang dicari dan diperhitungkan dalam perencanaan pembangunan daya FPB 60 M ini. Untuk ulasan berikut akan di skemakan hasil simulasi untuk penghasil daya dengan acuan jumlah konsumsi bahan bakar yang akan dipakai disetiap kondisi. Dengan mencantumkan tiap kecepatan dari 1-30 knot dengan SM 1.2 dan dengan kebutuhan daya listrik masing-masing 300 kW, 500 kW, 700 kW.



Gambar 4.13 Gambar Analisa Konsumsi Bahan Bakar SM 1.2 Kebutuhan Listrik 300 kW

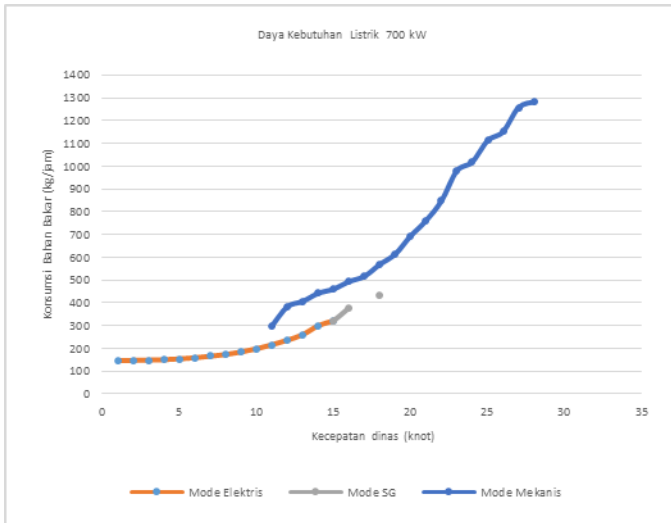
Gambar 4.13 diatas menunjukkan bahwa untuk kecepatan rendah sistem propulsi eletris belum tentu menjadi solusi terbaik dengan menyediakan konsumsi bahan bakar terendah, sebaliknya, pada kecepatan 13 knot hingga 22 knot sistem mekanis memiliki nilai konsumsi bahan bakar yang lebih rendah meskipun tidak signifikan. Setelah dianalisa, hal ini disebabkan

pemakaian generator untuk kebutuhan daya 300kW akan berada pada konsumsi disekitar 75% daya dari daya maksimal generator yang mana mempunyai nilai SFOC yang terendah untuk pemakaian generator yang dipakai dan saat diabung dengan konsumsi bahan bakar pada motor diesel, sstem mekanis memiliki nilai konsumsi bahan bakar total perjam yang lebih rendah. Untuk selanjutnya data grafik pada kebutuhan listrik 500kW



Gambar 4.14 Gambar Analisa Konsumsi Bahan Bakar SM 1.2 Kebutuhan Listrik 500 kW

Gambar 4.14 menunjukkan nilai pemakaian bahan bakar yang lebih rendah untuk sistem propulsi elektris pada kecepatan 12 knot hingga 15 knot. Dan 15 knot hingga 20 knot *shaft generator* menunjukkan nilai konsumsi bahan bakar lebih baik jika dibandingkan dengan propulsi mekanis. Nilai ini sesuai dengan teori awal bahwasanya propulsi elektris dan shaft generator akan lebih hemat bahan bakar pada kecepatan-kecepatan rendah.



Gambar 4.15 Gambar Analisa Konsumsi Bahan Bakar SM 1.2 Kebutuhan Listrik 700 kW

Gambar 4.15 juga menunjukkan kesamaan dengan teori bahwa sistem propulsi elektrik membuat konsumsi bahan bakar lebih minimum jika dibandingkan dengan propulsi mekanis, selain itu gambar 4.15 menunjukkan pada kecepatan 16 knot, service margin 1.2 dengan kebutuhan listrik 700 kW, bahwa pemakaian shaft generator tidak tersedia, namun untuk kecepatan 17 knot, *shaft generator* dapat digunakan. Setelah diperiksa ternyata daya yang dibutuhkan melebihi dari 90% daya pada putaran yang ditunjukkan ke kecepatan 16 knot namun dengan condong keatasnya engine envelope mengakibatkan pada kecepatan 17 knot, shaft generator justru mengalami ketersediaan. Kondisi inilah salah satu alasan mengapa simulasi akan membantu, dan menunjukkan bahwa ketersediaan dan jumlah kebutuhan bahan bakar akan tersebar secara acak dan membuat

perencanaan simulasi pembangkitan daya ini dibutuhkan nahkoda.

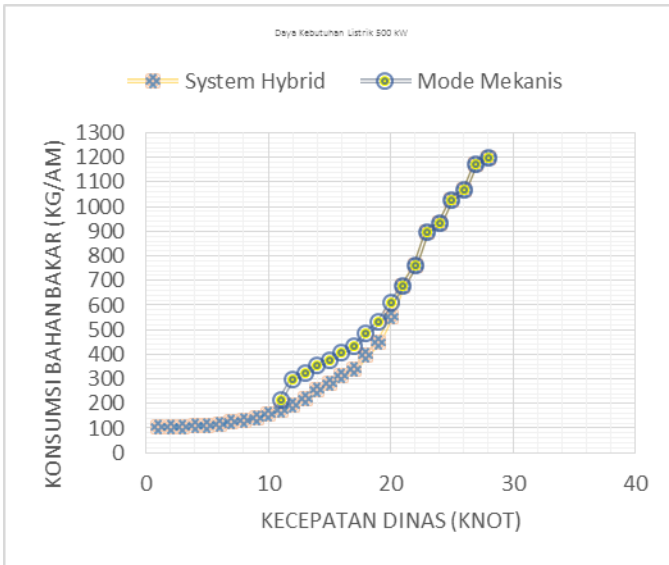
Dengan perbandingan ke 3 grafik, 4.13, 4.14 dan 4.15 dapat dilihat bahwa tren dalam pemakaian propulsi mekanis memiliki pola yang sama, hal ini disebabkan konsumsi bahan bakar yang sama untuk motor diesel dalam setiap kebutuhan listrik, dengan penjumlahan dengan nilai konsumsi bahan bakar pada *auxillary engine* tergantung kebutuhan listrik yang cenderung tetap dalam satu kondisi tersebut.

*Decision Support System* hanya bersifat membantu untuk menunjukkan nilai optimum dalam hal efisiensi atau pemakaian sistem yang terbaik, dan bukan sebagai pemberi keputusan, keputusan pemakaian akan tetap di pertanggungkan kepada nahkoda kapal.

Setelah analisa konsumsi bahan bakar untuk setiap sistem maka akan dilakukan perbandingan antara pemakaian sistem konvensional dengan sistem propulsi hybrid, sistem hybrid akan menggunakan sistem dengan konsumsi bahan bakar terendah seperti pada data diatas.

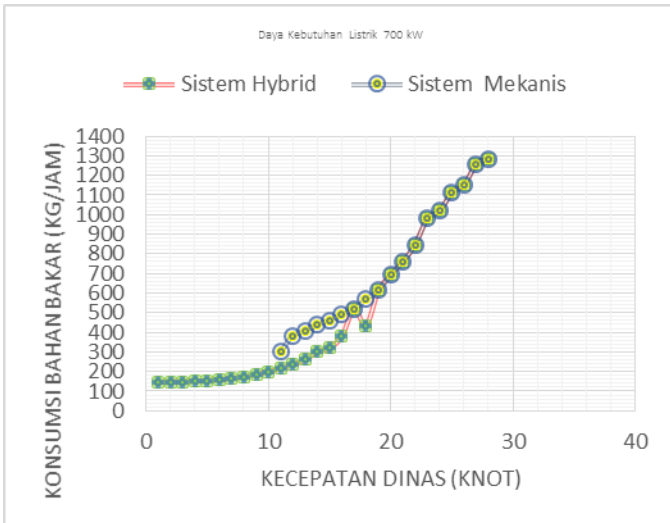
Dengan memperbandingkan sistem pada kecepatan 1-30 knot dan SM 1.2.





Gambar 4.16 Gambar Perbandingan Konsumsi Bahan Bakar antara Hybrid system dan Sistem Mekanis SM 1.2 Kebutuhan Listrik 500 kW

Dengan perbandingan pada gambar 4.16, untuk kondisi pada kecepatan 0 knot hingga 20 sistem propulsi hybrid akan memberikan keuntungan dengan menawarkan konsumsi bahan bakar yang lebih rendah jika dioperasikan dengan tepat sesuai simulasi yang telah dibuat ini, yang selanjutnya pada kecepatan 20 knot keatas motor diesel memiliki nilai bahan bakar rendah sehingga dipakailah sistem propulsi mekanis untuk menompang propulsi kapal dan kebutuhan listrik akomodasi yang tertera 500 kW.



Gambar 4.17 Gambar Perbandingan Konsumsi Bahan Bakar antara Hybrid system dan Sistem Mekanis SM 1.2 Kebutuhan Listrik 700 kW

Gambar 4.17 memberikan penyajian sama sesuai dengan teori kecuali nilai pada kecepatan 16 knot dan 17 dimana dikarenakan ketidak tersediaan pemakaian shaft generator pada kecepatan 16 knot, mengharuskan pemakaian sistem mekanis untuk propulsi kapal, namun pada 17 knot shaft generator kembali memberikan solusi pemakaian bahan bakar yang lebih rendah daripada sistem mekanis.

## LAMPIRAN

Program induk :

```
DiaProp = 1.308;
AAo = 0.90;
eDHP = 0.985;

VR1 = input('Kecepatan Kapal =');
LK = input('Kebutuhan Listrik Kapal =');
VRdat = xlsread('data tahanan.xls');
if VRdat<0 | VR1>max(VRdat(3:end,1));
    errorldg('Kecepatan tidak dalam
jangkauan', 'Warning');
    return;
end
SM =1:0.01:2;
SM =input('Sea Margin (1-2) = ');
for i=1:length(SM);
    if SM(i)<0 | SM(i)>2
        errorldg('Service Margin Berlebihan!');
        return;
    end
    resist=(interp1(VRdat(3:end,1),VRdat(3:end,2),VR1));
    Rt=SM(i)*resist;
end

rho= 1.025;
Vs=VR1/3.6*1.852;
DispV= 284.529;
t=0.179;
w=0.193230824;
ETaH=1.0176;

alpha=(resist./2)/Vs.^2
Va=Vs*(1-w);
betha=alpha/(rho*(1-t)*(1-w).^2*DiaProp.^2);

J=0:0.01:1;

KTsc=(J.^2)*betha;
KTss=KTsc.*SM;

%figure
plot ( J, KTss, J, KTprop, J, Effprop, Jpotong, py);
axis ([0 1 0 1]);

n=Vs/(Jpotong*DiaProp);
```

```

mVs=Va;
RPS= mVs./(Jpotong.*DiaProp);
Q= rho.*KQ./10.*(RPS.^2).*(DiaProp.^5);
DHP= 2.*pi.*RPS.*Q;
BHPn= (DHP./0.98);
RPM=RPS.*(1.486.*1.486).*60;
EPM = {mVs,RPS,Q,DHP,BHPn,RPM};
xlswrite('Hitung', EPM , 1, 'A1');

PME=BHPn./0.98./0.98;
SFOCa= xlsread('DatSFOC1.xlsx','Sheet1');
n1= SFOCa(1,2:36);
P1= SFOCa(2:58,1);
FC1=SFOCa(2:58,2:36);
FOCa= interp2(n1,P1,FC1,RPM,PME);
FOC1=(FOCa.*BHPn).*2;

if LK <=440*0.9;
    l=1;
else l=2;
end
PAE = LK./1;
if PAE<=220
    FC2=213.82+(((220-PAE)./220).*(213.82-208.73));
end
if PAE>200 & PAE<=330;
    FC2=208.73+(((330-PAE)./330).*(213.82-208.73));
else FC2=208.73+(((440-PAE)./440).*(215.73-208.73));
end
FOC2=LK.*FC2.*1;

PSG=(BHPn./0.9)+(LK./2./0.95);
SFOCb= xlsread('DatSFOC1.xlsx','Sheet1');
n2= SFOCb(1,2:36);
P2= SFOCb(2:58,1);
FC3=SFOCb(2:58,2:36);
FOCb= interp2(n2,P2,FC3,RPM,PSG);
FOC3=FOCb.*PSG.*2;

PEE=(2.*BHPn./0.9);
if PEE<=440
    k=1;
    PEE>440 & PEE<=880
    k=2;
else k=3;
    if PEE >(440.*3.*0.9)
        errordlg('Motor Listrik Tidak Mampu Menampung
Kebutuhan Daya');
    end
end

```

```

    end
end

PEE1=PEE+LK;

if PEE1>=0 & PEE1<=396;
    m=1;
elseif PEE1>396 & PEE1 <=792;
    m=2;
elseif PEE1>792 & PEE1 <=1188;
    m=3;
elseif PEE1>=1188;
    m=4;
end

PEE2=PEE1/m;
if PAE<=220
    FC4=213.82+(((220-PAE)/220).*(213.82-208.73));
end
if PAE>220 & PAE<=330;
    FC4=208.73+(((330-PAE)/330).*(213.82-208.73));
else FC4=208.73+(((440-PAE)/440).*(215.73-208.73));
end
FOC4=PEE2*FC4*m;

FOCs1= FOC1+FOC2;
for i =1:length(RPM)
    a=1;
    if RPM(i)>= 400 & RPM(i)<= 800
        BHPg = RPM(i)-300;
        deltaBHP = (0.9*BHPg)- BHPn(i);
        if deltaBHP >= 0
            BHPf(i)=BHPn(i);
        else
            errordlg('Daya Diluar Engine Envelope Diesel
Engine');
        end
    elseif RPM(i)>= 800 & RPM(i)<= 1500
        BHPg = 1.5714*RPM(i)-757.1429;
        deltaBHP =(0.9*BHPg)- BHPn(i);
        if deltaBHP >= 0
            BHPf(i)=BHPn(i);
        else
            errordlg('Daya Diluar Engine Envelope Diesel
Engine');
        end
    elseif RPM(i)>= 1500 & RPM(i)<= 1950
        BHPg = 2.48888*RPM(i)-2133.333;
        deltaBHP = (0.9*BHPg)- BHPn(i);

```

```

        if deltaBHP >= 0
            BHPf(i)=BHPn(i);
        else
            errorrdlg('Daya Diluar Engine Envelope Diesel
Engine');
        end
    elseif RPM(i)>= 1950 & RPM(i)<= 2100
        BHPg = 2720;
        deltaBHP =(0.9*BHPg)- BHPn(i);
        if deltaBHP >= 0
            BHPf(i)=BHPn(i);
        else
            errorrdlg('Daya Diluar Engine Envelope Diesel
Engine');
        end
    end

end
if BHPf(i)<=0
    FOC1=0;
end
if FOC1 <= 0;
    FOCsystemDiesel = 'invalid'
else FOCsystemDiesel = FOCs1
end
FOCs2= FOC4;
if PEE >=1100.*0.9;
    FOCsystemElectric = 'invalid'
else FOCsystemElectric = FOCs2
end

FOCs3= FOC3;
for i =1:length(RPM)
    a=1;
    if RPM(i)>= 400 & RPM(i)<= 800
        BHPg = RPM(i)-300;
        deltaBHP = (0.9*BHPg)- BHPn(i);
        if deltaBHP >= (LK./2)
            BHPf(i)=BHPn(i);
        else
            errorrdlg('Tidak Dapat Menggunakan SG');
        end
    elseif RPM(i)>= 800 & RPM(i)<= 1600
        BHPg = 1.5714*RPM(i)-757.1429;
        deltaBHP =(0.9*BHPg)- BHPn(i);
        if deltaBHP >= (LK./2)
            BHPf(i)=BHPn(i);
        else
            errorrdlg('Tidak Dapat Menggunakan SG');
        end
    end
end
end

```

```

end
elseif RPM(i) >= 1600 & RPM(i) <= 1950
    BHPg = 2.48888*RPM(i) - 2133.333;
    deltaBHP = (0.9*BHPg) - BHPn(i);
    if deltaBHP >= (LK./2)
        BHPn(i) = BHPn(i);
    else
        errorldg('Tidak Dapat Menggunakan SG');
    end
elseif RPM(i) >= 1950 & RPM(i) <= 2100
    BHPg = 2720;
    deltaBHP = (0.9*BHPg) - BHPn(i);
    if deltaBHP >= (LK./2)
        BHPn(i) = BHPn(i);
    else
        BHPn(i) = 0;
        errorldg('Tidak Dapat Menggunakan SG');
    end
end

end
if (deltaBHP - (LK./2)) <= 0
    PSG = 0;
end
if PSG <= 0;
    FOCsystemPSG = 'invalid'
else FOCsystemPSG = FOC3
end

```

Penyambungan Interface :

```
function varargout = simulasi(varargin)
% SIMULASI MATLAB code for simulasi.fig
%     SIMULASI, by itself, creates a new SIMULASI or
%     raises the existing
%     singleton*.
%
%     H = SIMULASI returns the handle to a new SIMULASI
%     or the handle to
%     the existing singleton*.
%
%     SIMULASI('CALLBACK', hObject,eventData,handles,...)
%     calls the local
%     function named CALLBACK in SIMULASI.M with the
%     given input arguments.
%
%     SIMULASI('Property','Value',...) creates a new
%     SIMULASI or raises the
%     existing singleton*. Starting from the left,
%     property value pairs are
%     applied to the GUI before simulasi_OpeningFcn gets
%     called. An
%     unrecognized property name or invalid value makes
%     property application
%     stop. All inputs are passed to
%     simulasi_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose
%     "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help
% simulasi

% Last Modified by GUIDE v2.5 18-Jul-2016 10:32:51

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn',  @simulasi_OpeningFcn, ...
                  'gui_OutputFcn',   @simulasi_OutputFcn, ...
                  ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
```



```

if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargin
    [varargout{1:nargout}] = gui_mainfcn(gui_State,
varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before simulasi is made visible.
function simulasi_OpeningFcn(hObject, ~, handles,
varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
% varargin   command line arguments to simulasi (see
VARARGIN)

% Choose default command line output for simulasi
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes simulasi wait for user response (see
UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the
command line.
function varargout = simulasi_OutputFcn(hObject,
eventdata, handles)
% varargout  cell array for returning output args (see
VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

```

```

% Get default command line output from handles structure
varargout{1} = handles.output;

function daya_diesel_Callback(hObject, eventdata,
handles)
% hObject      handle to daya_motor_listrik (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
daya_motor_listrik as text
%           str2double(get(hObject,'String')) returns
contents of daya_motor_listrik as a double

% --- Executes during object creation, after setting all
properties.
function daya_diesel_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to daya_motor_listrik (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function rpm_diesel_Callback(hObject, eventdata, handles)
% hObject      handle to rpm_diesel (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
rpm_diesel as text

```

```

%         str2double(get(hObject,'String')) returns
contents of rpm_diesel as a double

% --- Executes during object creation, after setting all
properties.
function rpm_diesel_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to rpm_diesel (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function FOC_listrik_Callback(hObject, eventdata,
handles)
% hObject    handle to FOC_listrik (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
FOC_listrik as text
%         str2double(get(hObject,'String')) returns
contents of FOC_listrik as a double

% --- Executes during object creation, after setting all
properties.
function FOC_listrik_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to FOC_listrik (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

```

```
% Hint: edit controls usually have a white background on
Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function daya_motor_listrik_Callback(hObject, eventdata,
handles)
% hObject    handle to daya_motor_listrik (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
daya_motor_listrik as text
%          str2double(get(hObject,'String')) returns
contents of daya_motor_listrik as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function daya_motor_listrik_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to daya_motor_listrik (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on
Windows.
```

```
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function rpm_motor_listrik_Callback(hObject, eventdata,
handles)
% hObject    handle to rpm_motor_listrik (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
```

```

% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
rpm_motor_listrik as text
%      str2double(get(hObject,'String')) returns
contents of rpm_motor_listrik as a double

% --- Executes during object creation, after setting all
properties.
function rpm_motor_listrik_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to rpm_motor_listrik (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function daya_SG_Callback(hObject, eventdata, handles)
% hObject      handle to daya_motor_listrik (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB
% handles      structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
daya_motor_listrik as text
%      str2double(get(hObject,'String')) returns
contents of daya_motor_listrik as a double

% --- Executes during object creation, after setting all
properties.
function daya_SG_CreateFcn(hObject, eventdata, handles)
% hObject      handle to daya_motor_listrik (see GCBO)
% eventdata    reserved - to be defined in a future version
of MATLAB

```

```
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on
Windows.
```

```
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function RPM_diesel_Callback(hObject, eventdata, handles)
% hObject    handle to RPM_diesel (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
RPM_diesel as text
%      str2double(get(hObject,'String')) returns
contents of RPM_diesel as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function RPM_diesel_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to RPM_diesel (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on
Windows.
```

```
%      See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
% --- Executes on button press in push_listrikfc.
```

```
function push_listrikfc_Callback(hObject, eventdata,
handles)
% hObject    handle to push_listrikfc (see GCBO)
```

```

% eventdata reserved - to be defined in a future version
of MATLAB
% handles structure with handles and user data (see
GUIDATA)

DiaProp = 1.308;
AAo = 0.90;
eDHP = 0.985;

VRdat = xlsread('data tahanan.xls');
if VRdat<0 | VR1>max(VRdat(3:end,1));
    errordlg('Kecepatan tidak dalam
jangkauan', 'Warning');
return;
end
%SM =1:0.01:2;

for i=1:length(SM);
    if SM(i)<0 | SM(i)>2
        errordlg('Service Margin Berlebihan!');
        return;
    end
    resist=(interp1(VRdat(3:end,1),VRdat(3:end,2),VR1));
    Rt=SM(i)*resist;
end

rho= 1.025;
Vs=VR1/3.6*1.852;
DispV= 284.529;
t=0.179;
w=0.193230824;
ETaH=1.0176;

alpha=(resist./2)/Vs.^2;
Va=Vs*(1-w);
beta=alpha/(rho*(1-t)*(1-w).^2*DiaProp.^2);

J=0:0.01:1;

KTsc=(J.^2).*beta;
KTss=KTsc.*SM;

mVs=Va;
RPS= mVs./(Jpotong.*DiaProp);
Q= rho.*KQ./10.*(RPS.^2).*(DiaProp.^5);
DHP= 2.*pi.*RPS.*Q;
BHPn= (DHP./0.98);

```

```

RPM=RPS.*(1.486.*1.486).*60;
EPM = {mVs,RPS,Q,DHP,BHPn,RPM};
xlswrite('Hitung', EPM , 1, 'A1');

PME=BHPn./0.98./0.98;
SFOCa= xlsread('DatSFOC1.xlsx','Sheet1');
n1= SFOCa(1,2:36);
P1= SFOCa(2:58,1);
FC1=SFOCa(2:58,2:36);
FOCa= interp2(n1,P1,FC1,RPM,PME);
FOC1=(FOCa.*BHPn).*2;

if LK <=440*0.9;
    l=1;
else l=2;
end
PAE = LK./1;
if PAE<=220
    FC2=213.82+(((220-PAE)./220).*(213.82-208.73));
end
if PAE>200 & PAE<=330;
    FC2=208.73+(((330-PAE)./330).*(213.82-208.73));
else FC2=208.73+(((440-PAE)./440).*(215.73-208.73));
end
FOC2=LK.*FC2.*1;

PSG=(BHPn./0.9)+(LK./2./0.95);
SFOCb= xlsread('DatSFOC1.xlsx','Sheet1');
n2= SFOCb(1,2:36);
P2= SFOCb(2:58,1);
FC3=SFOCb(2:58,2:36);
FOCb= interp2(n2,P2,FC3,RPM,PSG);
FOC3=FOCb.*PSG.*2;

PEE=(2.*BHPn./0.98);
if PEE<=440
    k=1;
    PEE>440 & PEE<=880
    k=2;
else k=3;
    if PEE >(440.*3.*0.9)
        errordlg('Motor Listrik Tidak Mampu Menampung
Kebutuhan Daya');
    end
end

PEE1=PEE+LK;

```



```

if PEE1>=0 & PEE1<=396;
    m=1;
elseif PEE1>396 & PEE1 <=792;
    m=2;
elseif PEE1>792 & PEE1 <=1188;
    m=3;
elseif PEE1>=1188;
    m=4;
end

PEE2=PEE1/m;
if PAE<=220
    FC4=213.82+(((220-PAE)/220).*(213.82-208.73));
end
if PAE>220 & PAE<=330;
    FC4=208.73+(((330-PAE)/330).*(213.82-208.73));
else FC4=208.73+(((440-PAE)/440).*(215.73-208.73));
end
FOC4=PEE2*FC4*m;

FOCs2= FOC4;
if PEE >=1100.*0.9;
    FOCsystemElectric = 0
else FOCsystemElectric = FOCs2
end

FOCListrik=FOCsystemElectric./1000;
if m>=1
    AE1=PEE2;
else AE1=0;
end
if m>=2
    AE2=PEE2;
else AE2=0;
end
if m>=3
    AE3=PEE2;
else AE3=0;
end
if m>=4
    AE4=PEE2;
else AE4=0;
end
if PEE2>=396;
    AE1=0;
    AE2=0;
    AE3=0;

```

```

    AE4=0;
end

set(handles.FOC_listrik,'string',FOCListrik);
set(handles.Daya_G1,'string',AE1);
set(handles.Daya_G2,'string',AE2);
set(handles.Daya_G3,'string',AE3);
set(handles.gen4,'string',AE4);

% --- Executes on button press in push_2SGfc.
function push_2SGfc_Callback(hObject, eventdata, handles)
% hObject    handle to push_2SGfc (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

DiaProp = 1.308;
AAo = 0.90;
eDHP = 0.985;

VRdat = xlsread('data tahanan.xls');
if VRdat<0 | VR1>max(VRdat(3:end,1));
    errordlg('Kecepatan tidak dalam
jangkauan','Warning');
    return;
end
%SM =1:0.01:2;

for i=1:length(SM);
    if SM(i)<0 | SM(i)>2
        errordlg('Service Margin Berlebihan');
        return;
    end
    resist=(interp1(VRdat(3:end,1),VRdat(3:end,2),VR1));
    Rt=SM(i)*resist;
end

rho= 1.025;
Vs=VR1/3.6*1.852;
DispV= 284.529;
t=0.179;
w=0.193230824;
ETaH=1.0176;

alpha=(resist./2)/Vs.^2;
Va=Vs*(1-w);
beta=alpha/(rho*(1-t)*(1-w).^2*DiaProp.^2);

```

```

J=0:0.01:1;

KTsc=(J.^2).*betha;
KTss=KTsc.*SM;

KQ=KQprop(index);

mVs=Va;
RPS= mVs./(Jpotong.*DiaProp);
Q= rho.*KQ./10.*(RPS.^2).*(DiaProp.^5);
DHP= 2.*pi.*RPS.*Q;
BHPn= (DHP./0.98);
RPM=RPS.*(1.486.*1.486).*60;
EPM = {mVs,RPS,Q,DHP,BHPn,RPM};
xlswrite('Hitung', EPM , 1, 'A1');

PME=BHPn./0.98./0.98;
SFOCa= xlsread('DatSFOC1.xlsx','Sheet1');
n1= SFOCa(1,2:36);
P1= SFOCa(2:58,1);
FC1=SFOCa(2:58,2:36);
FOCa= interp2(n1,P1,FC1,RPM,PME);
FOC1=(FOCa.*BHPn).*2;

if LK <=440*0.9;
    l=1;
else l=2;
end
PAE = LK./l;
if PAE<=220
    FC2=213.82+(((220-PAE)./220).*(213.82-208.73));
end
if PAE>200 & PAE<=330;
    FC2=208.73+(((330-PAE)./330).*(213.82-208.73));
else FC2=208.73+(((440-PAE)./440).*(215.73-208.73));
end
FOC2=LK.*FC2.*1;

PSG=(BHPn./0.9)+(LK./2./0.95);
SFOCb= xlsread('DatSFOC1.xlsx','Sheet1');
n2= SFOCb(1,2:36);
P2= SFOCb(2:58,1);
FC3=SFOCb(2:58,2:36);
FOCb= interp2(n2,P2,FC3,RPM,PSG);
FOC3=FOCb.*PSG.*2;

```

```

PEE=(2.*BHPn./0.9);
PEE1=PEE+LK;

if FOCa<=197;
    FOC1=0;
end
if FOCb<=197;
    FOC3=0;
end
if PEE1>=0 & PEE1<=400;
    m=1;
elseif PEE1>400 & PEE1 <=800;
    m=2;
elseif PEE1>800 & PEE1 <=1200;
    m=3;
else
    m=4;
end

PEE2=PEE1./m;
if PEE2 <=200;
    FC4=201.6+(((200-PEE2)./200).*(201.6-201.6));
elseif PEE2>200 & PEE2 <=300;
    FC4=201.6+(((300-PEE2)./300).*(201.6-201.6));
else FC4=201.6+(((400-PEE2)./400).*(210-201.6));
end
FOC4=PEE2.*FC4.*m;

FOCs3= FOC3;
for i =1:length(RPM)
    a=1;
    if RPM(i)>= 400 & RPM(i)<= 800
        BHPg = RPM(i)-300;
        deltaBHP = (0.9*BHPg)- BHPn(i);

    elseif RPM(i)>= 800 & RPM(i)<= 1600
        BHPg = 1.5714*RPM(i)-757.1429;
        deltaBHP =(0.9*BHPg)- BHPn(i);

    elseif RPM(i)>= 1600 & RPM(i)<= 1950
        BHPg = 2.48888*RPM(i)- 2133.333;
        deltaBHP = (0.9*BHPg)- BHPn(i);

    else RPM(i)>= 1950 & RPM(i)<= 2100
        BHPg = 2720;
        deltaBHP =(0.9*BHPg)- BHPn(i);

    end
    if deltaBHP >= (LK./2)

```

```

        BPh(i)=BPhn(i);
    else
        BPh(i)=0;
        errordlg('Tidak Dapat Menggunakan SG');
    end
end
if (deltaBHP - (LK./2))<=0
    PSG = 0;
end
if PSG <= 0;
    FOCsystemPSG = 0
else FOCsystemPSG = FOC3;
end

FOCPHG=FOCsystemPSG./1000;

set(handles.FOC_PSG,'string',FOCPHG);
set(handles.Daya_G1,'string',0);
set(handles.Daya_G2,'string',0);
set(handles.Daya_G3,'string',0);
set(handles.gen4,'string',0);

% --- Executes on button press in push_3dieselfc.
function push_3dieselfc_Callback(hObject, eventdata, handles)
% hObject    handle to push_3dieselfc (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

DiaProp = 1.308;
AAo = 0.90;
eDHP = 0.985;

VRdat = xlsread('data tahanan.xls');
if VRdat<0 | VR1>max(VRdat(3:end,1));
    errordlg('Kecepatan tidak dalam
jangkauan','Warning');
    return;
end
%SM =1:0.01:2;

for i=1:length(SM);
    if SM(i)<0 | SM(i)>2
        errordlg('Service Margin Berlebihan');
        return;
    end
end

```

```

        resist=(interp1(VRdat(3:end,1),VRdat(3:end,2),VR1));
        Rt=SM(i)*resist;
end

rho= 1.025;
Vs=VR1/3.6*1.852;
DispV= 284.529;
t=0.179;
w=0.193230824;
ETaH=1.0176;

alpha=(resist./2)/Vs.^2;
Va=Vs*(1-w);
beta=alpha/(rho*(1-t)*(1-w).^2*DiaProp.^2);

J=0:0.01:1;

KTsc=(J.^2).*beta;
KTss=KTsc.*SM;

mVs=Va;
RPS= mVs./(Jpotong.*DiaProp);
Q= rho.*KQ./10.*(RPS.^2).*(DiaProp.^5);
DHP= 2.*pi.*RPS.*Q;
BHPn= (DHP./0.98);
RPM=RPS.*(1.486.*1.486).^60;
EPM = {mVs,RPS,Q,DHP,BHPn,RPM};
xlswrite('Hitung', EPM , 1, 'A1');

PME=BHPn./0.98./0.98;
SFOCa= xlsread('DatSFOC1.xlsx','Sheet1');
n1= SFOCa(1,2:36);
P1= SFOCa(2:58,1);
FC1=SFOCa(2:58,2:36);
FOCa= interp2(n1,P1,FC1,RPM,PME);
FOC1=(FOCa.*BHPn).^2;

if LK <=440*0.9;
    l=1;
else l=2;
end
PAE = LK./1./0.95;
if PAE<=220
    FC2=213.82+(((220-PAE)./220).*(213.82-208.73));
end
if PAE>200 & PAE<=330;
    FC2=208.73+(((330-PAE)./330).*(213.82-208.73));
else FC2=208.73+(((440-PAE)./440).*(215.73-208.73));
end

```

```

FOC2=LK.*FC2.*1;

PSG=(BHPn./0.9)+(LK./2./0.95);
SFOCb= xlsread('DatSFOC1.xlsx','Sheet1');
n2= SFOCb(1,2:36);
P2= SFOCb(2:58,1);
FC3=SFOCb(2:58,2:36);
FOCb= interp2(n2,P2,FC3,RPM,PSG);
FOC3=FOCb.*PSG.*2;

PEE=(2.*BHPn./0.9);
if PEE<=440
    k=1;
    PEE>440 & PEE<=880
    k=2;
else k=3;
end

PEE1=PEE+LK;

if PEE1>=0 & PEE1<=396;
    m=1;
elseif PEE1>396 & PEE1 <=792;
    m=2;
elseif PEE1>792 & PEE1 <=1188;
    m=3;
elseif PEE1>=1188;
    m=4;
end

PEE2=PEE1/m;
if PAE<=220
    FC4=213.82+((220-PAE)./220).*(213.82-208.73));
end
if PAE>220 & PAE<=330;
    FC4=208.73+((330-PAE)./330).*(213.82-208.73));
else FC4=208.73+((440-PAE)./440).*(215.73-208.73));
end
FOC4=PEE2*FC4*m;

FOCs1= FOC1+FOC2;
for i =1:length(RPM)
    a=1;
    if RPM(i)>= 400 & RPM(i)<= 800
        BHPg = RPM(i)-300;
        deltaBHP = (0.9*BHPg)- BHPn(i);
    elseif RPM(i)>= 800 & RPM(i)<= 1500
        BHPg = 1.5714*RPM(i)-757.1429;
        deltaBHP = (0.9*BHPg)- BHPn(i);
    end
end

```

```

elseif RPM(i)>= 1500 & RPM(i)<= 1950
    BHPg = 2.48888*RPM(i)-2133.333;
    deltaBHP = (0.9*BHPg)- BHPn(i);
else RPM(i)>= 1950 & RPM(i)<= 2100
    BHPg = 2720;
    deltaBHP =(0.9*BHPg)- BHPn(i);
end
if deltaBHP >= 0
    BHPf(i)=BHPn(i);
else
    errordlg('Daya Diluar Engine Envelope Diesel
Engine');
end
end
if FOCa<=197;
    FOC1=0
end
if FOCb<=197;
    FOC3=0
end
if FOC1 <= 0;
    FOCsystemDiesel = 0
else FOCsystemDiesel = FOCs1
end
if l>=1
    AE1=PAE;
else AE1=0;
end
if l>=2
    AE2=PAE;
else AE2=0;
end
if l>=3
    AE3=PAE;
else AE3=0;
end
if l>=4
    AE4=PAE;
else AE4=0;
end

FOCmekanik=FOCsystemDiesel./1000;

set(handles.FOC_diesel,'string',FOCmekanik);
set(handles.Daya_G1,'string',AE1);
set(handles.Daya_G2,'string',AE2);
set(handles.Daya_G3,'string',AE3);
set(handles.gen4,'string',AE4);

```



```
function FOC_PSG_Callback(hObject, eventdata, handles)
% hObject    handle to FOC_PSG (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
FOC_PSG as text
%         str2double(get(hObject,'String')) returns
contents of FOC_PSG as a double
```

```
% --- Executes during object creation, after setting all
properties.
```

```
function FOC_PSG_CreateFcn(hObject, eventdata, handles)
% hObject    handle to FOC_PSG (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called
```

```
% Hint: edit controls usually have a white background on
Windows.
```

```
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end
```

```
function FOC_diesel_Callback(hObject, eventdata, handles)
% hObject    handle to FOC_diesel (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of
FOC_diesel as text
%         str2double(get(hObject,'String')) returns
contents of FOC_diesel as a double
```

```

% --- Executes during object creation, after setting all
properties.
function FOC_diesel_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to FOC_diesel (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in push_input.
function push_input_Callback(hObject, eventdata, handles)
% hObject    handle to push_input (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

DiaProp = 1.308;
AAo = 0.90;
eDHP = 0.985;

VRdat = xlsread('data tahananan.xls');
if VRdat<0 | VR1>max(VRdat(3:end,1));
    errordlg('Kecepatan tidak dalam
jangkauan','Warning');
    return;
end
%SM =1:0.01:2;

for i=1:length(SM);
    if SM(i)<0 | SM(i)>2
        errordlg('Service Margin Berlebihan');
        return;
    end
    resist=(interp1(VRdat(3:end,1),VRdat(3:end,2),VR1));
    Rt=SM(i)*resist;
end

rho= 1.025;

```

```

Vs=VR1/3.6*1.852;
DispV= 284.529;
t=0.179;
w=0.193230824;
ETaH=1.0176;

alpha=(resist./2)/Vs.^2;
Va=Vs*(1-w);
beta=alpha/(rho*(1-t)*(1-w).^2*DiaProp.^2);

J=0:0.01:1;

KTsc=(J.^2).*beta;
KTss=KTsc.*SM;

mVs=Va;
RPS= mVs./(Jpotong.*DiaProp);
Q= rho.*KQ./10.*(RPS.^2).*(DiaProp.^5);
DHP= 2.*pi.*RPS.*Q;
BHPn= (DHP./0.98);
RPM=RPS.*(1.486.*1.486).*60;
EPM = {mVs,RPS,Q,DHP,BHPn,RPM};
RPMm=RPM./1.486;
xlswrite('Hitung', EPM , 1, 'A1');

PME=BHPn./0.98./0.98;
SFOCa= xlsread('DatSFOC1.xlsx','Sheet1');
n1= SFOCa(1,2:36);
P1= SFOCa(2:58,1);
FC1=SFOCa(2:58,2:36);
FOCa= interp2(n1,P1,FC1,RPM,PME);
FOC1=(FOCa.*BHPn).^2;

if LK <=440*0.9;
    l=1;
else l=2;
end
PAE = LK./l;
if PAE<=220
    FC2=213.82+(((220-PAE)./220).*(213.82-208.73));
end
if PAE>220 & PAE<=330;
    FC2=208.73+(((330-PAE)./330).*(213.82-208.73));
else FC2=208.73+(((440-PAE)./440).*(215.73-208.73));
end
FOC2=LK.*FC2.*l;

PSG=(BHPn./0.9)+(LK./2./0.95);

```

```

SFOCb= xlsread('DatSFOC1.xlsx','Sheet1');
n2= SFOCb(1,2:36);
P2= SFOCb(2:58,1);
FC3=SFOCb(2:58,2:36);
FOCb= interp2(n2,P2,FC3,RPM,PSG);
FOC3=FOCb.*PSG.*2;

PEE=(2.*BHPn./0.98)
if PEE<=440
    k=1;
    PEE>440 & PEE<=880
    k=2;
else k=3;
    if PEE >(440.*3.*0.9)
        errordlg('Motor Listrik Tidak Mampu Menampung
Kebutuhan Daya');
    end
end

PEE1=PEE+LK;

if PEE1>=0 & PEE1<=396;
    m=1;
elseif PEE1>396 & PEE1 <=792;
    m=2;
elseif PEE1>792 & PEE1 <=1188;
    m=3;
elseif PEE1>=1188;
    m=4;
end

PEE2=PEE1/m;
if PAE<=220
    FC4=213.82+(((220-PAE)./220).*(213.82-208.73));
end
if PAE>220 & PAE<=330;
    FC4=208.73+(((330-PAE)./330).*(213.82-208.73));
else FC4=208.73+(((440-PAE)./440).*(215.73-208.73));
end
FOC4=PEE2*FC4*m;

FOCs1= FOC1+FOC2;
for i =1:length(RPM)
    a=1;
    if RPM(i)>= 400 & RPM(i)<= 800
        BHPg = RPM(i)-300;
        deltaBHP = (0.9*BHPg)- BHPn(i);
    elseif RPM(i)>= 800 & RPM(i)<= 1500

```

```

        BHPg = 1.5714*RPM(i)-757.1429;
        deltaBHP =(0.9*BHPg)- BHPn(i);
elseif RPM(i)>= 1500 & RPM(i)<= 1950
    BHPg = 2.48888*RPM(i)-2133.333;
    deltaBHP = (0.9*BHPg)- BHPn(i);
else RPM(i)>= 1950 & RPM(i)<= 2100
    BHPg = 2720;
    deltaBHP =(0.9*BHPg)- BHPn(i);
end
if deltaBHP >= 0
    BHPf(i)=BHPn(i);
else
    errordlg('Daya Diluar Engine Envelope Diesel
Engine');
end
end
if FOCa<=197;
    FOC1=0
end
if FOCb<=197;
    FOC3=0
end
if FOC1 <= 0;
    FOCsystemDiesel = 0
else FOCsystemDiesel = FOCs1
end
FOCs2= FOC4;
if PEE >=1100.*0.9;
    FOCsystemElectric = 0
else FOCsystemElectric = FOCs2
end
if PEE>=1100.*0.9
    PSE = 0
else PSE=PEE./2;
end

FOCs3= FOC3;
for i =1:length(RPM)
    a=1;
    if RPM(i)>= 400 & RPM(i)<= 800
        BHPg = RPM(i)-300;
        deltaBHP = (0.9*BHPg)- BHPn(i);

elseif RPM(i)>= 800 & RPM(i)<= 1600
    BHPg = 1.5714*RPM(i)-757.1429;
    deltaBHP =(0.9*BHPg)- BHPn(i);

elseif RPM(i)>= 1600 & RPM(i)<= 1950
    BHPg = 2.48888*RPM(i)- 2133.333;

```

```

        deltaBHP = (0.9*BHPg) - BHPn(i);

    else RPM(i) >= 1950 & RPM(i) <= 2100
        BHPg = 2720;
        deltaBHP = (0.9*BHPg) - BHPn(i);

    end
    if deltaBHP >= (LK./2)
        BHPn(i) = BHPn(i);
    else
        BHPn(i) = 0;
        error('Tidak Dapat Menggunakan SG');
    end
end
if (deltaBHP - (LK./2)) <= 0
    PSG = 0;
end
if PSG <= 0;
    FOCsystemPSG = 0
else FOCsystemPSG = FOC3
end
if FOC1 <= 0;
    PME = 0
end
if FOC4 <= 0;
    PSE = 0
end
if FOC3 <= 0;
    PSG = 0
end

if PME <= 0;
    RPMe = 0;
else RPMe = RPM;
end
if PSE <= 0;
    RPMm = 0;
end
if PSG <= 0;
    RPMs = 0;
else RPMs = RPM;
end

FOClistrik = FOCsystemElectric./1000;
FOCPsg = FOCsystemPSG./1000;
FOCmekanik = FOCsystemDiesel./1000;

set(handles.daya_motor_listrik, 'string', PSE);
set(handles.daya_diesel, 'string', PME);

```

```

set(handles.daya_SG, 'string', PSG)
set(handles.rpm_diesel, 'string', RPMm);
set(handles.rpm_motor_listrik, 'string', RPMs);
set(handles.RPM_diesel, 'string', RPMe);

function input_sea_margin_Callback(hObject, eventdata,
handles)
% hObject      handle to input_sea_margin (see GCBO)
% eventdata   reserved - to be defined in a future version
of MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Hints: get(hObject, 'String') returns contents of
input_sea_margin as text
%           str2double(get(hObject, 'String')) returns
contents of input_sea_margin as a double

% --- Executes during object creation, after setting all
properties.
function input_sea_margin_CreateFcn(hObject, eventdata,
handles)
% hObject      handle to input_sea_margin (see GCBO)
% eventdata   reserved - to be defined in a future version
of MATLAB
% handles     empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject, 'BackgroundColor'),
get(0, 'defaultUicontrolBackgroundColor'))
    set(hObject, 'BackgroundColor', 'white');
end

function input_kebutuhan_listrik_Callback(hObject,
eventdata, handles)
% hObject      handle to input_kebutuhan_listrik (see GCBO)
% eventdata   reserved - to be defined in a future version
of MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Hints: get(hObject, 'String') returns contents of
input_kebutuhan_listrik as text

```

```

%         str2double(get(hObject,'String')) returns
contents of input_kebutuhan_listrik as a double

% --- Executes during object creation, after setting all
properties.
function input_kebutuhan_listrik_CreateFcn(hObject,
eventdata, handles)
% hObject    handle to input_kebutuhan_listrik (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function input_kecepatan_Callback(hObject, eventdata,
handles)
% hObject    handle to input_kecepatan (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
input_kecepatan as text
%         str2double(get(hObject,'String')) returns
contents of input_kecepatan as a double

% --- Executes during object creation, after setting all
properties.
function input_kecepatan_CreateFcn(hObject, eventdata,
handles)
% hObject    handle to input_kecepatan (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

```



```
% Hint: edit controls usually have a white background on  
Windows.
```

```
% See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function Daya_G3_Callback(hObject, eventdata, handles)  
% hObject handle to Daya_G3 (see GCBO)  
% eventdata reserved - to be defined in a future version  
of MATLAB  
% handles structure with handles and user data (see  
GUIDATA)
```

```
% Hints: get(hObject,'String') returns contents of  
Daya_G3 as text  
% str2double(get(hObject,'String')) returns  
contents of Daya_G3 as a double
```

```
% --- Executes during object creation, after setting all  
properties.
```

```
function Daya_G3_CreateFcn(hObject, eventdata, handles)  
% hObject handle to Daya_G3 (see GCBO)  
% eventdata reserved - to be defined in a future version  
of MATLAB  
% handles empty - handles not created until after all  
CreateFcns called
```

```
% Hint: edit controls usually have a white background on  
Windows.
```

```
% See ISPC and COMPUTER.  
if ispc && isequal(get(hObject,'BackgroundColor'),  
get(0,'defaultUicontrolBackgroundColor'))  
    set(hObject,'BackgroundColor','white');  
end
```

```
function Daya_G2_Callback(hObject, eventdata, handles)  
% hObject handle to Daya_G2 (see GCBO)  
% eventdata reserved - to be defined in a future version  
of MATLAB  
% handles structure with handles and user data (see  
GUIDATA)
```

```

% Hints: get(hObject,'String') returns contents of
Daya_G2 as text
%         str2double(get(hObject,'String')) returns
contents of Daya_G2 as a double

% --- Executes during object creation, after setting all
properties.
function Daya_G2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Daya_G2 (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function Daya_G1_Callback(hObject, eventdata, handles)
% hObject    handle to Daya_G1 (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of
Daya_G1 as text
%         str2double(get(hObject,'String')) returns
contents of Daya_G1 as a double

% --- Executes during object creation, after setting all
properties.
function Daya_G1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to Daya_G1 (see GCBO)
% eventdata  reserved - to be defined in a future version
of MATLAB
% handles    empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.

```

```

%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function gen4_Callback(hObject, eventdata, handles)
% hObject     handle to gen4 (see GCBO)
% eventdata   reserved - to be defined in a future version
of MATLAB
% handles     structure with handles and user data (see
GUIDATA)

% Hints: get(hObject,'String') returns contents of gen4
as text
%           str2double(get(hObject,'String')) returns
contents of gen4 as a double

% --- Executes during object creation, after setting all
properties.
function gen4_CreateFcn(hObject, eventdata, handles)
% hObject     handle to gen4 (see GCBO)
% eventdata   reserved - to be defined in a future version
of MATLAB
% handles     empty - handles not created until after all
CreateFcns called

% Hint: edit controls usually have a white background on
Windows.
%           See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in clear.
function clear_Callback(hObject, eventdata, handles)
% hObject     handle to clear (see GCBO)
% eventdata   reserved - to be defined in a future version
of MATLAB
% handles     structure with handles and user data (see
GUIDATA)
set(handles.input_kecepatan,'string','');
set(handles.input_sea_margin,'string','');
set(handles.input_kebutuhan_listrik,'string','');

```

```
set(handles.FOC_listrik, 'string', '');
set(handles.FOC_PSG, 'string', '');
set(handles.FOC_diesel, 'string', '');
set(handles.daya_motor_listrik, 'string', '');
set(handles.daya_diesel, 'string', '');
set(handles.daya_SG, 'string', '');
set(handles.rpm_diesel, 'string', '');
set(handles.rpm_motor_listrik, 'string', '');
set(handles.RPM_diesel, 'string', '');
set(handles.Daya_G1, 'string', '');
set(handles.Daya_G2, 'string', '');
set(handles.Daya_G3, 'string', '');
set(handles.gen4, 'string', '');
```

## BAB V

### KESIMPULAN DAN SARAN

Sistem Propulsi Hybrid memberikan kombinasi yang rumit dalam pengoperasiannya, dengan berbagai factor input dan factor output yang berbeda untuk setiap sistem. Factor input terdiri dari kecepatan yang diinginkan, service margin sesuai keputusan nahkoda untuk nilai hambatan yang berpengaruh yang sesuai dengan kondisi kapal dan laut, dan nilai kebutuhan daya listrik yang mencakup komponen yang bekerja di kapal.

#### 5.1 Kesimpulan

Dengan pengerjaan tugas pengaturan simulasi pembangkitan daya ini, dan dengan analisa yang dilakukan dapat disimpulkan bahwa :

1. Decision Support System dapat membantu Nahkoda mengoperasikan Fast Patrol Boat 60 M dengan optimum dan mendapatkan nilai bahan bakar minimum untuk kondisi yang diinginkan dengan desain interface yang mudah dipahami dan dijalankan.
2. Efek dari perubahan nilai *service margin* berbeda dengan perubahan pada kecepatan karena *service margin* hanya akan berpengaruh dengan tahanan pada badan kapal.
3. Perubahan kebutuhan listrik berpengaruh terhadap kondisi ketersediaan sistem shaft generator, dan tidak berpola bahwa dengan kecepatan dan service margin lebih tinggi bahwa sistem *Shaft Generator* tidak dapat digunakan, seperti pada kebutuhan daya 700 kWh dengan service margin 1.2, pada saat kecepatan 16 knot

dan 17 knot, justru tersedia pada 18 knot dan pada 17 knot tidak.

4. Pemakaian Sistem Propulsi Hybrid menambah efisiensi pada kecepatan rendah seperti contohnya pada kecepatan 17 knot, dan dengan kebutuhan listrik 500 kW sistem Shaft generator membutuhkan konsumsi 337 kg/jam sedangkan untuk pemakaian sistem konvensional hingga 431 kg/jam, dengan menghemat pemakaian sekitar 94 kg/jam-nya.

## **5.2 Saran**

Penggunaan Decision Support System ini masih dalam tahap penelitian dan perlu dilakukan penelitian dengan pengoperasian sebenarnya, walaupun data, proses perhitungan dan hasilnya berdasarkan pada teori-teori yang sudah terbukti, namun penelitian lebih lanjut diperlukan agar dapat digunakan dalam membantu pengoperasian kapal yang memiliki karakteristik sama dengan kapal Fast Patrol Boat 60 M.

**DAFTAR PUSTAKA**

- Adji, S. W., 2006. Diktat Kuliah Pengenalan Sistem Propulsi Kapal, Teknik Sistem Perkapalan FTK-ITS, Surabaya.
- Adji, S.W., 2005. *Engine And Propeller Matching*, Teknik Sistem Perkapalan FTK ITS, Surabaya.
- Götz, Gerhard, 2004“*Technical Project Guide for MTU 12 and 16 series*” Germany.
- Wahyudi, Dedy, 2010. “Laporan Tesis Perancangan Sistem Propulsi Hybrid (DMP&DEP) pada Kapal Patroli jenis Trimaran,” Surabaya:.
- Storbacka, Mårten, 2015.*Energy Efficiency improvements in shipping*. Wolffintie 36 M10 FI-65200 Vaasa Finland
- Kwasieckyj, B., 2013, *Efficiency analysis and design methodology of hybrid propulsion systems*. 86150 Augsburg, Germany.
- Woud, H.K dan Strapersma, 2002. Design of propulsion and electric power generation systems,IMarEST, Institut of Marine Engineering, Science and Technology, London.
- Koenhardo. E.S., 2014. Laporan Sidang Terbuka, Manajemen Pembangkitan Daya pada Kapal Trimaran dengan Sistem Propulsi Hybrid Shaft Generator Berbasis Neural Network, Surabaya.

*“Halaman ini sengaja di kosongkan”*



## BIODATA PENULIS



Penulis lahir di Kediri, Jawa Timur, pada 20 Maret 1994, dengan menjadi anak pertama dari empat bersaudara. Penulis menjalani pendidikan formal pada SD Negeri Bringin, SMP Negeri 2 Pare, dan SMA Negeri 2 Pare. Dan dengan izin ALLAH SWT penulis berhasil diterima untuk menjalani pendidikan sarjana (S1) di Institut Teknologi Sepuluh Nopember pada Jurusan Teknik Sistem Perkapalan pada tahun 2012 dan terdaftar dengan NRP 4212100072. Selama perkuliahan Penulis cukup mengikuti kegiatan seminar dan pelatihan baik akademik maupun non-akademik sebagai penunjang ilmu yang diharapkan berguna untuk waktu kedepannya.