

PEMPROGRAMAN LIFTING SURFACE GEOMETRY DENGAN VISUAL BASIC FOR APLICATION DI CATIA V5

Tugas Akhir

Diajukan untuk memenuhi salah satu syarat guna mencapai gelar Sarjana dalam
Ilmu Teknik Penerbangan

Disusun oleh :

Nama : Iqbal R.Pamungkas

NIM : 13600055

Pembimbing :

Ir. Rais Zain, M.Eng



Institut Teknologi Bandung



**DEPARTEMEN TEKNIK PENERBANGAN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI BANDUNG
2005**

TUGAS SARJANA

Diberikan kepada : Iqbal R.Pamungkas

Dosen pembimbing : Ir. Rais Zain, M. Eng.

Jangka waktu penyelesaian : 1 (satu) tahun

Judul : PEMPROGRAMAN LIFTING SURFACE
GEOMETRY DENGAN VISUAL BASIC FOR
APLICATION DI CATIA V5

Bandung, Desember 2005



Ir. Rais Zain, M. Eng.

Dosen Pembimbing

Catatan :

1. Pembimbing Tugas Sarjana
2. File Jurusan
3. Mahasiswa ybs.

LEMBAR PENGESAHAN

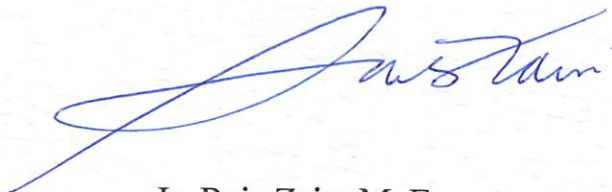
Tugas Sarjana ini telah diterima dan disahkan sebagai syarat
memperoleh gelar

SARJANA TEKNIK PENERBANGAN

**DEPARTEMEN TEKNIK PENERBANGAN
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI BANDUNG**

Bandung, Desember 2005

Dosen Pembimbing

A handwritten signature in blue ink, appearing to read 'Rais Zain', is written over a faint, circular official stamp.

Ir. Rais Zain, M. Eng.

RINGKASAN

Secara *de facto*, CATIA sebagai standard industri dalam industri pesawat terbang, digunakan oleh sebagian besar pabrik pesawat terbang. Diintergrasikan dengan DELMIA, ENOVIA, dan SMARTEAM, memungkinkan para pemakai untuk mensimulasikan keseluruhan cakupan dari proses desain industri, dari konsep awal dan desain produk ke analisa, perakitan dan pemeliharaan. CATIA memiliki kemampuan dalam desain solid model secara *real-time*, ketika terjadi perubahan spesifikasi (karena suatu hal terjadi perubahan satu atau lebih part dari bagian pesawat yang akan diproduksi) dapat digabungkan kembali secara cepat dan efisien. Model solid menjadi melebihi definisi objek 3D (dimana karakteristik part dan informasi manufaktur didefinisikan secara realistic).

Membuat aplikasi tambahan di CATIA berupa automasi untuk membuat surface sayap dengan menggunakan *Visual Basic for Application (VBA)*. Hanya dengan memasukkan input-input standar yang dibutuhkan maka secara cepat terbentuk model 3D sayap yang diharapkan. Aplikasi ini diharapkan dapat bermanfaat sebagai salah satu perangkat dalam pelaksanaan tugas perancangan pesawat udara di Departemen Teknik Penerbangan ITB, dan sebagai landasan untuk penelitian selanjutnya, sehingga hasil rancangan dapat lebih baik lagi. Sebagai tahap awal pengembangan aplikasi pemodelan pada CATIA, akan mempermudah dan membuka jalan untuk pengembangan aplikasi pemodelan lebih lanjut, sehingga dapat digunakan dalam industri penerbangan sebagai salah satu perangkat perancangan.

Pengenalan berbagai macam *workbench* pada CATIA dengan studi kasus model surface dan solid dari struktur sayap dan wingbox N250-100PN. Disebut sebagai N250-100PN karena model yang dibangun dengan CATIA ini merupakan rekonstruksi dari data sayap N250-100 milik PT. Dirgantara Indonesia. Model solid struktur wingbox ini tidak bisa dikatakan sama dengan milik N250-100 sesungguhnya karena data-data yang didapat dari PT. Dirgantara Indonesia berbeda-beda sehingga sulit untuk menentukan data yang benar yang akan menjadi referensi.

Pada penyusunan produk dari sayap N250-100PN ini dikelompokkan menjadi beberapa *sub-assembly* dan didalam *sub-assembly* terdapat *sub-assembly* yang lebih kecil lagi hingga menjadi *sub-assembly* yang hanya memiliki part. Pengelompokan produk dan part pada tugas akhir ini berdasarkan kesamaan posisi. Metoda ini dilakukan guna memudahkan dalam manajemen model.

KATA PENGANTAR

Segala puji bagi Allah Swt, Tuhan semesta alam. Syukur Alhamdulillah tiada hentinya dipanjatkan ke hadirat Allah SWT atas segala nikmat yang telah diberikan kepada kita sampai detik ini. Atas segala kuasa-Nya jugalah, penulis diberi kesempatan untuk menyelesaikan tugas akhir dan kuliah di Departemen Teknik Penerbangan ITB dengan lancar.

Tugas akhir ini diselesaikan sebagai salah satu syarat kelulusan di Departemen Teknik Penerbangan ITB. Penyusunan tulisan ini diharapkan dapat menjadi langkah awal pengembangan lebih lanjut dari penelitian yang dilakukan penulis. Menyadari segala keterbatasan yang ada pada diri penulis, sudah barang tentu tulisan ini masih jauh dari sempurna. Namun demikian mudah-mudahan jerih payah penulis dengan segala keterbatasan dan kemampuannya dapat bermanfaat bagi pihak-pihak yang terkait.

Cukup banyak kendala yang dihadapi selama proses pelaksanaan tugas akhir ini. Hanya berkat rahmat Allah Swt yang Maha Pengasih serta dukungan berbagai pihak yang membuat penulis mampu menyelesaikannya dengan hasil yang memuaskan. Untuk itu ucapan terima kasih dan penghormatan yang tidak terkira penulis haturkan kepada siapapun yang terkait dengan penulisan ini, khususnya bagi :

1. Papa dan Mama tercinta yang merupakan sumber semangat bagi penulis. Tulisan ini tidak senilai dengan pengorbanan keduanya dalam membesarkan penulis, namun hanya inilah yang bisa dipersembahkan sampai saat ini. Untuk Kakak-kakak tercinta, terima kasih atas bimbingan dan dukungan sebagai kakak yang selalu memberi teladan bagi adiknya.
2. Bapak Rais Zain, M.Eng. sebagai pembimbing. Dukungan moril dan materiil yang membuat penulis tetap optimis menyelesaikan tugas ini.
3. Seluruh dosen dan staf Departemen Teknik Penerbangan ITB atas bimbingan, pelayanan, dan penyediaan segala fasilitas yang mendukung pelaksanaan tugas akhir ini.
4. Seluruh teman-teman di Departemen Teknik Penerbangan ITB khususnya angkatan 2000.
6. Teman-teman di kost Ciung Wanara 14.
7. Seluruh pihak yang tak mungkin disebutkan satu persatu.

Sebagai penutup, penulis memohon maaf yang sebesar-besarnya bila selama pelaksanaan tugas akhir ada pihak-pihak yang dirugikan atau ada tindakan-tindakan penulis yang kurang berkenan. Sebagai harapan penulis, semoga tugas akhir ini dapat terus dikembangkan untuk kemanfaatan bersama.

Bandung, Desember 2005

Iqbal R.Pamungkas

DAFTAR ISI

RINGKASAN	i
KATA PENGANTAR	ii
DAFTAR ISI	iv
DAFTAR GAMBAR	vii
BAB 1	
PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Tujuan	2
1.3 Manfaat Tugas Akhir	3
BAB 2	
OVERVIEW KEMAMPUAN CATIA	4
2.1 Pendahuluan	4
2.2 Product Lifecycle Management (PLM)	8
2.3 CATIA V5	10
2.4 Beberapa workbench pada CATIA	11
2.4.1 Workbench Mechanical Design	15
2.4.1.a Part Design	16
2.4.1.b Workbench Assembly Design	17
2.4.1.c Workbench Sketcher	17
2.4.1.d Workbench Generatif Drafting	17
2.4.2 Workbench Shape	18
2.4.3 Workbench Analysis & Simulation	19
2.4.3.a Workbench Advanced Meshing Tools	20
2.4.3.b Workbench Generatif Struktural Analysis	21
2.5 Pemrograman / Automation pada CATIA	22
2.5.1 Workbench Knowledgeware	23
2.5.2 Macros / .CATScripts	23
2.5.3 Visual basic for Aplication (VBA) / .CATvba	25

2.5.4 CAA	26
2.6 Interaksi CATIA dengan software lain	27
 BAB 3	
PEMPROGRAMAN LIFTING SURFACE GEOMETRY MENGGUNAKAN VBA	28
3.1 Pendahuluan	28
3.2 Antarmuka Antara Aplikasi Pemodelan dengan CATIA	28
3.2.1 Simple Wing	29
3.2.2 Inboard-Outboard Wing	34
3.3 Airfoil database	38
3.3 Model matematika	40
3.4 Penyusunan titik koordinat airfoil.	42
 BAB 4	
PEMODELAN TIGA DIMENSI (3D) UNTUK STRUKTUR WINGBOX	44
4.1 Wire-frame modeling	45
4.2 Surface modeling	45
4.3 Solid modeling	46
4.4 Hierarki parent-children	48
4.5 Assembly dari model solid wingbox	51
 BAB 5	
PEMAKAIAN WORKBENCH ANALYSIS & SIMULATION UNTUK WINGBOX.	56
5.1 Pendahuluan	56
5.2 Pembentukan model	56
5.3 Elemen	58
5.3 Pembentukan mesh	60
5.4 Penentuan properties, contact, restraint, dan load.	

BAB 6	
KESIMPULAN DAN SARAN	68
6.1 Kesimpulan	68
6.2 Saran	69
REFERENSI	70
LAMPIRAN	
CONTOH PROGRAM PEMODELAN	71

DAFTAR GAMBAR

Gambar 2.1 : A typical product cycle	4
Gambar 2.2 : CAD and CAM disciplines	6
Gambar 2.3 : CATIA V5 P3 user interface	12
Gambar 2.4 : Aplikasi utama dan workbench yang terdapat didalamnya	13
Gambar 2.5 : Part Rib.Tip menggunakan workbench <i>Aerospace Sheet Metal Design</i>	14
Gambar 2.6 : Part Rib.Tip menggunakan workbench <i>Part Design</i>	15
Gambar 2.7 : Aplikasi utama Mechanical Design dan workbench di dalamnya	16
Gambar 2.8 : Pembuatan gambar 2D (drafting) dari hasil model 3D	18
Gambar 2.9 : Aplikasi utama Shape dan workbench di dalamnya	19
Gambar 2.10 : Aplikasi utama Analysis & Simulation dan workbench di dalamnya	20
Gambar 2.11 : Mesh dari model solid	21
Gambar 2.12 : Hasil analisis static dengan finite elemen model	22
Gambar 2.13 : Aplikasi utama Knowledgeware dan workbench di dalamnya	23
Gambar 2.14 : Window Macros editor	24
Gambar 2.15 : Window VBA editor	26
Gambar 2.16 : Model CATIA dapat diterima oleh perangkat lunak lain	27
 Gambar 3.1 : Toolbar	 29
Gambar 3.2 : Window dialog lifting surface geometry (Create), tab platform, dengan pilihan Simple Platform	30
Gambar 3.3 : Window dialog simple platform	31
Gambar 3.4 : Window dialog lifting surface geometry (Create), tab- Section, tipe Simple Platform	32
Gambar 3.5 : Sayap dengan sudut <i>sweep</i> dan <i>dihedral positif</i>	33
Gambar 3.6 : Sayap dengan sudut <i>sweep</i> dan <i>dihedral negative</i>	33
Gambar 3.7 : Window dialog lifting surface geometry (Create), tab platform, dengan pilihan Inboard-Outboard	34
Gambar 3.8 : Window dialog Inboard-Outboard platform	35

Gambar 3.9 : Tab- <i>Section</i> , tipe Inboard-Outboard	36
Gambar 3.10 : Sayap N250-100PN	37
Gambar 3.11 : Sayap denagn sudut <i>sweep outboard</i> dan <i>dihedral outboard negative</i>	38
Gambar 3.12 : Window dialog select a text file	39
Gambar 3.13 : Contoh format data airfoil	40
Gambar 3.14 : Plane equation	41
Gambar 3.15 : Coordinate transformation equation	41
Gambar 3.16 : Line equation	42
Gambar 3.17 : Line-plane intersection equation	42
Gambar 3.18 : Contoh progam untuk proses iterasi	43
 Gambar 4.1 : Assembly dari seluruh sayap secara keseluruhan	 44
Gambar 4.2 : Wireframe	45
Gambar 4.3 : Informmasi data yang jelas	46
Gambar 4.4 : Thick Surface dari rip cap	47
Gambar 4.5 : Intersection	47
Gambar 4.6 : Specification tree	49
Gambar 4.7 : External references 1	50
Gambar 4.8 : External references 2	51
Gambar 4.9 : WingBox	51
Gambar 4.10 : <i>Desk</i> (link antara produk ke part dan master)	52
Gambar 4.11 : Master- duplicate	53
Gambar 4.12 : Open part in new window	54
Gambar 4.13 : Modifikasi pada part duplicate	55
 Gambar 5.1 : Finite element model (bar dan shell)	 57
Gambar 5.2 : Material properties	57
Gambar 5.3 : Solid model	58
Gambar 5.4 : Window dialog beam meshing	59
Gambar 5.5 : Window dialog octree triangle mesh	59

Gambar 5.6 : Window dialog octree tetrahedron mesh	59
Gambar 5.7 : Solid mesh dengan element linear tetrahedron	62
Gambar 5.8 : Mesh dari finite element model (bar dan shell)	63
Gambar 5.9 : Window dialog 1D dan 2D properties	64
Gambar 5.10 : Restrain berupa Clamp	65
Gambar 5.11 : Load yang berupa Distributed force	65
Gambar 5.12 : Window dialog dari Model Checker	66
Gambar 5.13 : Hasil analisis Von Mises Stres untuk model beam dan shell	67
Gambar 5.14 : Hasil analisis Von Mises Stres untul model solid	67

BAB 1

PENDAHULUAN

1.1 Latar Belakang

Teknologi *Computer Aided Design* (CAD) sudah sangat umum digunakan dalam perancangan pesawat udara. Kemampuan CAD dalam penayangan gambar interaktif sangat sesuai dengan kebutuhan visualisasi dalam proses perancangan. Melalui CAD, obyek-obyek gambar dua dimensi maupun tiga dimensi dapat didefinisikan dengan cepat dalam derajat akurasi yang tinggi.

Pada fasa perancangan konseptual pesawat udara, peran CAD secara ringkas dapat dijelaskan sebagai berikut :

1. Pemodelan Geometri

Pemodelan geometri adalah suatu teknik untuk merepresentasikan suatu geometri dua dimensi atau tiga dimensi yang sering diperlukan dalam kegiatan perancangan dan analisa teknik, serta manufaktur. Sedangkan model geometri adalah diskripsi matematis suatu bentuk geometri yang dapat ditampilkan dan dimanipulasi pada monitor.

2. Penggambaran Otomatis (*Automated Drafting*)

Untuk tujuan presentasi, file gambar teknik dapat dibuat *hard copy* atau diplot atau dicetak.

3. *Review* dan Evaluasi Rancangan

Hasil akhir suatu rancangan perlu diperiksa lagi, seperti penampilan benda dari berbagai sudut pandang, memeriksa ketepatan dimensi, posisi suatu komponen, interferensinya dengan komponen lain, dan kinematika komponen. Dengan sistem CAD, evaluasi suatu rancangan dapat lebih mudah dan fleksibel karena dilakukan dengan membuat perintah standar yang telah disediakan suatu aplikasi CAD.

4. Analisa Teknik

Dari hasil pemodelan geometri dan input lain, dapat dilakukan analisis hasil rancangan seperti properti (luas, volume, masa, momen inersia, titik berat), perhitungan tegangan, aerodinamika, dan *heat transfer*.

Dengan perannya tersebut, sistem CAD sangat tangguh digunakan untuk pembuatan gambar teknik, visualisasi tiga dimensi, dan analisa teknik hasil perancangan pesawat udara. Lebih dari itu visualisasi tiga dimensi sangat penting dalam perancangan sebuah pesawat udara. Selain menunjukkan bentuk yang sesungguhnya, model tiga dimensi juga sangat membantu dalam konseptualisasi yang akan memberikan hasil rancangan yang lebih baik. Sistem CAD sendiri juga menyediakan fasilitas yang cukup menarik untuk model tiga dimensi yaitu memberikan tampilan untuk tiap arah pandangan yang berbeda.

Dalam industri besar banyak digunakan suatu sistem CAD yang terintegrasi dengan CAM (*Computer Aided Manufactur*), dahulu CAD/CAM hanya mampu dijalankan dengan *mainframe*. Makin meningkatnya teknologi perangkat keras computer, CAD/CAM menjadi dapat dijalankan dengan *personal computer* (PC) yang ada saat ini.

CATIA yang diakui secara *de facto* sebagai standard industri dalam industri pesawat terbang memiliki kemampuan lebih dari sekedar CAD/CAM. CATIA sudah menjadi salah satu solusi dari *Product Lifecycle Management (PLM)* .

Tugas akhir ini dilakukan pengembangan terhadap aplikasi CATIA untuk tujuan pemodelan tiga dimensi dalam perancangan *lifting surface geometry*.

1.2 Tujuan

Tugas akhir ini memiliki dua tujuan utama, pertama adalah membuat model solid dari struktur wingbox N250-100PN. Disebut sebagai N250-100PN karena model yang dibangun dengan CATIA ini merupakan rekonstruksi dari data sayap N250-100 milik PT.Dirgantara Indonesia. Model solid struktur wingbox ini tidak bisa dikatakan sama dengan milik N250-100 sesungguhnya karena data-data yang didapat dari PT.DI berbeda-beda sehingga sulit untuk menentukan data yang benar yang akan menjadi referensi.

Tujuan kedua dari tugas akhir ini adalah membuat aplikasi tambahan di CATIA berupa automasi untuk membuat surface sayap dengan menggunakan *Visual Basic for Application (VBA)*.

1.3 Manfaat Tugas Akhir

Tugas akhir ini diharapkan dapat bermanfaat sebagai salah satu perangkat dalam pelaksanaan tugas perancangan pesawat udara di Departemen Teknik Penerbangan ITB, dan sebagai landasan untuk penelitian selanjutnya, sehingga hasil rancangan dapat lebih baik lagi.

Sebagai tahap awal pengembangan aplikasi pemodelan, tugas akhir ini diharapkan akan mempermudah dan membuka jalan untuk pengembangan aplikasi pemodelan lebih lanjut sehingga, dapat digunakan dalam industri penerbangan sebagai salah satu perangkat perancangan.

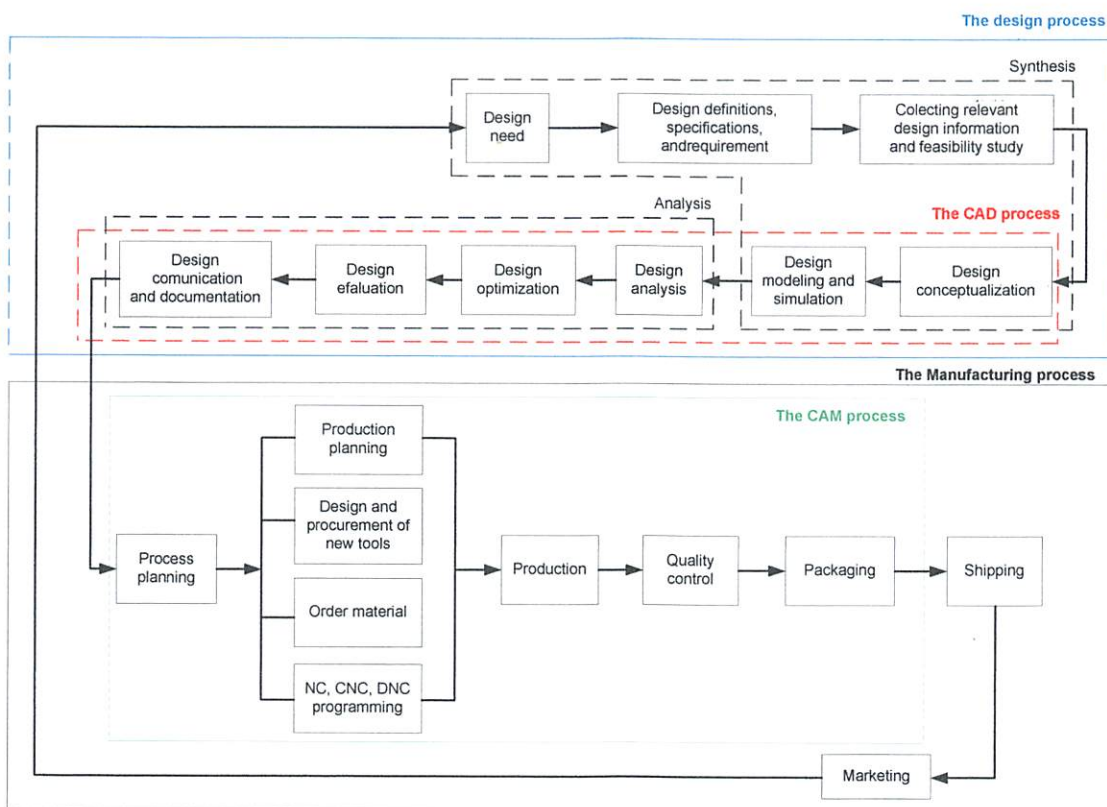
BAB 2

OVERVIEW KEMAMPUAN CATIA

2.1 Pendahuluan

CAD/CAM (Computer Aided Design / Computer Aided Manufacturing) telah digunakan di engineering practice dengan berbagai jalan termasuk drafting, design, simulasi, analisis, dan manufacturing.

Secara global referensi [1] menjelaskan struktur suatu produk yang disajikan pada gambar 2.1



Gambar 2.1 A typical product cycle. (Ref.1)

Produk berlangsung melalui dua proses utama, yakni proses desain dan proses manufaktur. Sintesis dan analisis adalah subproses utama dari proses desain. Selama sintesis, sebuah desain diambil dari bentuk skets (*sketches*) dan tampilan gambar (*layout drawings*) yang menunjukkan hubungan dari part-part suatu produk. *Sketches* dan *drawings* dapat dibuat

menggunakan system CAD/CAM atau gambar tangan pada kertas. Pada kegiatan ini dilakukan diskusi antar beberapa team desain.

Subproses analisis dimulai dengan usaha memasukkan desain konseptual kedalam konteks *engineering sciences* untuk menghitung performa dari produk yang diharapkan, hal ini dibutuhkan desain model dan simulasi. Aspek penting dari analisis adalah memilih desain dari berbagai macam desain yang ada dan menemukan solusi terbaik dari tiap permasalahan desain. Hasil keluaran dari analisis adalah dokumentasi desain berupa *engineering drawings* yang biasa dikenal sebagai *blueprints*.

Proses manufaktur dimulai dengan proses perencanaan (*process planning*) dan diakhiri oleh produk nyata. Proses perencanaan benar benar dipertimbangkan karena merupakan awal dari proses manufaktur. Proses perencanaan merupakan usaha untuk menentukan urutan yang paling efisien dalam memproduksi sebuah produk. Orang yang berada pada proses perencanaan ini harus sadar atas berbagai aspek dalam manufaktur untuk merencanakan dengan baik. Perencana secara khusus bekerja dengan *blueprints* dan dapat berkomunikasi dengan team desain guna mengklarifikasi atau meminta perubahan didalam desain untuk menyesuaikan dengan permintaan manufaktur. Hasil dari *process planning* adalah sebuah perencanaan produksi (*production planning*), pengadaan peralatan (*tools procurement*), *material order*, dan *machine programming*.

Setelah fase proses perencanaan terpenuhi, produksi nyata dari produk dimulai. Part-part yang dihasilkan harus diinspeksi dan biasanya harus sesuai dengan control kualitas standar yang diminta. Part yang telah diinspeksi kemudian diasembli, dibungkus, diberi label dan dikirim kepada pengguna. Tanggapan dari pasar (*market feedback*) biasanya disatukan kedalam proses desain. Dengan feedback ini maka siklus tertutup dari produk diperoleh, seperti gambar 2.1

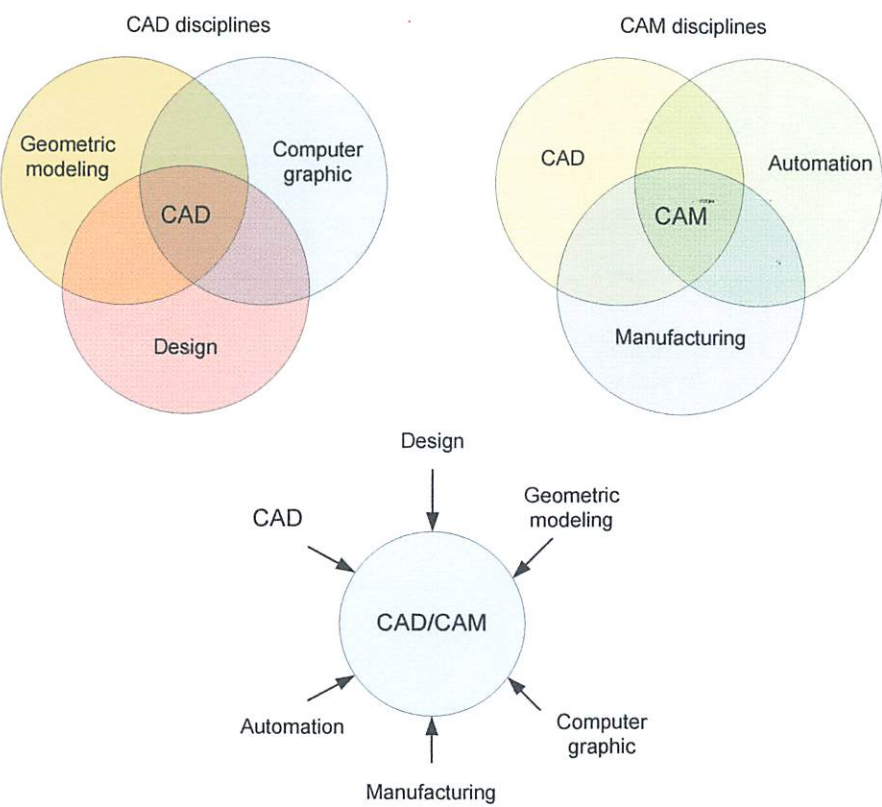
Siklus produk yang terlihat pada gambar 2.1 menunjukkan dasar dari pendefinisian ruang lingkup CAD/CAM. Proses CAD adalah subset dari proses desain dan proses CAM adalah subset dari proses manufaktur. Para engineer yang dilibatkan didalam proses desain biasanya

mereka sendiri *designer* CAD yang menetapkan proses CAD. Begitu pula para engineer manufaktur yang menetapkan proses CAM.

Pada inti dari proses CAD dan CAM adalah model geometri dari produk dibawah desain. Aktivitas dari proses CAD termasuk sifat massa, analisis elemen hingga (*FEM*), *dimensioning*, *tolerancing*, *assembly modeling*, *generating shaded images*, serta *drafting* dan dokumentasi. Aktivitas dari proses CAM mencakup *CAPP (Computer Aided Process Planning)*, *NC (Numerical Control) programming*, *design of injection mold*, *CMM (coordinate measuring machines) verification*, *inspection*, *assembly via robots*, dan *packaging*.

Proses CAD menggunakan tiga disiplin yaitu *modeling*, *computer graphics*, dan *design*. Proses CAM menggunakan disiplin dari CAD itu sendiri, manufaktur dan *automation*.

Gambar 2.2 menunjukkan disiplin dari kedua proses tersebut.



Gambar 2.2 CAD and CAM disciplines (Ref.1)

Sekarang system CAD/CAM telah banyak sehingga dapat diklasifikasikan menjadi empat kelompok yang tergantung pada peralatan dan pasar yang dilayaninya, fungsi atau kegunaannya, dan fleksibilitas yang disediakan. Keempat kelompok itu adalah *low end*, *midrange*, *high end*, dan *specialized*.

System *low-end* cenderung pada target pengguna yang tidak membutuhkan kecanggihan dan produk tidak kompleks; pada umumnya produk terdiri atas jumlah part yang sedikit dimana geometri tidak rumit. penggunaannya cenderung focus dalam model geometri sederhana dan *drafting*. sebagai contoh yang termasuk didalamnya ialah AutoCAD, Autodesk Inventor, dan CADKEY.

System taraf menengah (*midrange*) pada target pengguna yang memiliki dan membutuhkan model kompleks. Jumlah part tiap produk cukup besar buat aplikasi-aplikasi taraf menengah. Tidak seperti system *low-end*, system *midrange* menunjang aplikasi desain dan manufaktur. Masing-masing dimasukkan ke dalam engine model geometri atau bekerjasama dengan aplikasi lain. Yang termasuk didalamnya ialah SolidWorks, Pro/E, dan MasterCAM.

System *High-end* merupakan warisan dari system CAD/CAM yang muncul pada kisaran tahun 1970 dan terus berkembang tiap tahunnya. System ini menunjang modeling, analisis, dan manufaktur untuk produk-produk yang kompleks seperti pesawat udara, mobil dan lainnya. Yang termasuk didalamnya ialah Unigraphics, Parasolid, SDRC I-DEAS dan CATIA. Unigraphics, Parasolid, dan I-DEAS dikembangkan oleh EDS sedangkan SolidWorks dan CATIA dikembangkan oleh Dassault Systemes.

System khusus (*specialized*), termasuk didalamnya ACIS (Spatial corp.) dan Parasolid (EDS). Masing-masing menyediakan model geometri universal dan *graphics kernel (software)* yang sangat kuat, perusahaan ini memperoleh ijin untuk membangun system CAD/CAM secara penuh.

System mana yang akan dipilih oleh suatu perusahaan? Pemilihan suatu system CAD/CAM sebagian besar dipengaruhi oleh ukuran dan tingkat kompleksitas system dari perusahaan. Perusahaan berukuran kecil, semakin khusus CAD/CAM yang dibutuhkan dan semakin sedikit kemampuan CAD/CAM yang dibutuhkan. Sebagai contoh agen konsultan kecil cenderung untuk memilih system yang mudah digunakan seperti AutoCAD, CADKEY, SolidEdge.

Sebuah perusahaan berukuran sedang akan memilih SolidWorks, Pro/E, Unigraphics, I-DEAS, atau CATIA, tetapi sering tidak dengan PLM (Product Lifecycle Management) atau modul dari software *high-end* yang lain.

Perusahaan Besar cenderung membeli system CAD/CAM seperti perusahaan berukuran sedang tetapi memperoleh lebih, termasuk aplikasi *high-end*, PLM. Organisasi besar dapat dengan mudah memutuskan harga atas dari system tersebut karena mereka memiliki produk-produk yang kompleks dengan banyak departemen yang harus berbagi dan modifikasi data produk secara bersamaan. Kondisi ini mengakibatkan PLM menjadi modul yang harus dimiliki.

2.2 Product Lifecycle Management (PLM)

Tantangan pasar global saat ini, perusahaan harus berinovasi untuk dapat *survive*. Inovasi harus terjadi disemua dimensi - produk, proses dan organisasi – guna meningkatkan daya saing dan pencapaian bisnis. Untuk membuat mereka berbeda, perusahaan harus dapat menangkap, mengatur, dan meningkatkan asset intelektual mereka. Hal ini dapat terpenuhi melalui aplikasi yang sesuai dengan pendekatan *Product Lifecycle Management (PLM)*. Aplikasi ini merupakan jalan yang dibutuhkan oleh pihak-pihak yang berkaitan dengan perusahaan. PLM adalah suatu pendekatan strategi bisnis guna membantu perusahaan untuk mencapai gol bisnisnya seperti mengurangi biaya-biaya, meningkatkan mutu, dan memendekkan waktu untuk menjual, serta memperbarui produknya, jasa, dan operasi bisnis.

Sehingga *Product Lifecycle Management (PLM)* dapat didefinisikan sebagai :

- Sebuah pendekatan strategi bisnis yang menerapkan solusi bisnis yang konsisten dengan mendukung kolaborasi antara kreasi, manajemen, penyebaran dan pemakaian informasi dari *product definition*.
- Membantu pihak-pihak yang berhubungan dekat dengan perusahaan (pelanggan, designer, dll).
- Berputar dari konsep “*end of life*” dari sebuah produk atau industri.
- Mengintegrasikan sumber daya manusia (SDM), system bisnis, proses dan informasi.

Perlu diketahui bahwa PLM bukan didefinisikan sebagai satu atau beberapa bagian dari teknologi. PLM adalah definisi dari pendekatan bisnis guna menyelesaikan permasalahan akan manajemen informasi data lengkap dari *product definition* – menciptakan informasi tersebut, memenejemennya hingga selesai, serta penyebaran dan penggunaannya. PLM tidak hanya sekedar teknologi, tetapi merupakan sebuah pendekatan dimana proses sama penting atau lebih penting dari data.

Inti dari konsep dasar PLM adalah :

1. Universal, menjamin atau mengamankan, mengatur akses dan penggunaan informasi dari definisi produk (*product definition*).
2. Memelihara integritas dari *product definition* dan informasi yang berkaitan selama alur hidup produk.
3. Mengatur dan memelihara proses bisnis yang digunakan untuk menciptakan, mengatur, menyebarkan, berbagi dan menggunakan informasi tersebut.

Informasi meliputi semua media (elektronik dan hardcopy), PLM terutama tentang mengatur penyajian secara digital dari informasi itu .

Berdasar pada pengalaman bertahun – tahun, solusi PLM dapat mendukung jangkauan yang luas dari produk, contoh dari produk tersebut termasuk produk manufaktur, seperti kendaraan, computer, telepon, mainan hingga pesawat terbang. Banyak juga produk sekarang termasuk didalamnya software, firmware dan komponen elektronik dimana data harus diatur. Beberapa organisasi memiliki asset – asset berumur panjang yang membutuhkan manajemen seperti peralatan distribusi jaringan, yaitu, air, gas, power, telekomunikasi, atau fasilitas seperti

pabrik, pengeboran minyak, gedung-gedung, bandar udara, pelabuhan, dan system rel kereta. Produk lain termasuk jembatan, jalan layang dan proyek sipil yang lain. Organisasi dengan sector industri yang banyak sangat sukses dengan menggunakan solusi PLM untuk mengatur informasi produk melalui siklus hidup dari seluruh produknya.

2.3 CATIA V5

PLM memungkinkan industri pesawat terbang untuk menciptakan produk inovatif, mensimulasikan siklus hidup produknya, dan menunjang pengembangan. Manfaat ini berpengaruh sangat besar pada industri pesawat terbang yang harus mengatur assembly kompleks dan sangat besar yang terdiri dari atas 3 juta part.

Yang ditawarkan Dassault Systèmes' PLM adalah terdiri atas suatu deretan solusi perangkat lunak yang terintegrasi sangat halus. Masing-Masing solusi melaksanakan suatu peranan tertentu, termasuk: perancangan produksi baru, manajemen pengembangannya sepanjang daur hidup produk, dan rancang-bangun proses pabrikasi yang terkait. Solusinya adalah :

1. Memungkinkan struktur pesawat terbang dan pengembangan sistem berdasar pada industry-proven terbaik,
2. Mendukung proses spesifik yang ada pada setiap sub-bagian *aerospace*, yaitu., pesawat terbang, helikopter, mesin/motor dan satelit,
3. Menyediakan suatu kolaboratif *3D workspace* di mana pesertanya pihak-pihak yang terkait langsung dengan industri tersebut (OEMS, *risk-sharing partners*, para penyalur dan pelanggan) yang dapat saling berhubungan real-time di seluruh dunia.

Dassault Systèmes' PLM Solutions adalah :

CATIA® - untuk kolaborasi antara disain produk dan simulasi.

DELMIA™ - untuk kolaborasi antara manufaktur dan proses pabrikasi.

ENOVIA™ dan SMARTEAM® - untuk kolaborasi antara manajemen data produk, *lifecycle management*, pendukung pengambilan keputusan bagi pihak-pihak yang terkait langsung dengan industri.

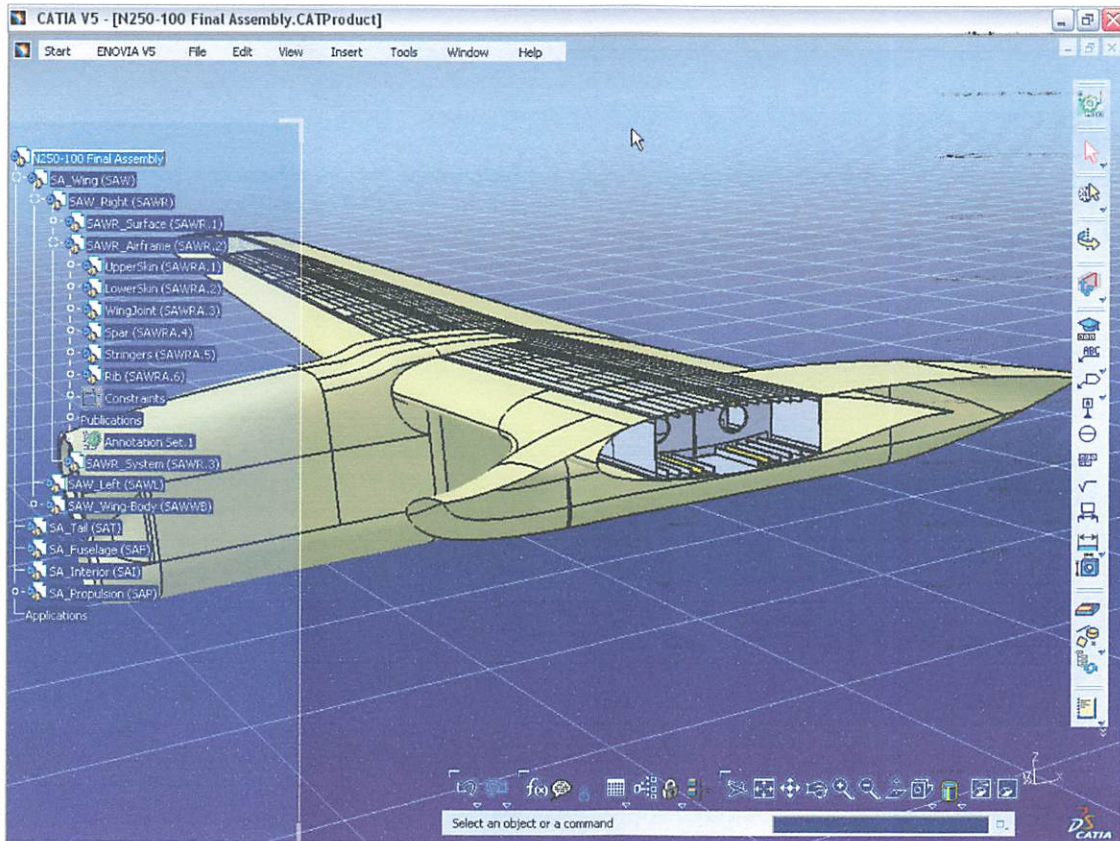
Sebagai pemimpin dunia dalam solusi *Product Lifecycle Management* (PLM) dan pelopor untuk software 3D, Dassault Systèmes memperoleh lebih dari 65,000 pelanggan didalam 80 negara. Dassault Systèmes mengembangkan dan memasarkan aplikasi perangkat lunak PLM

dan jasa yang mendukung proses industri dan menyediakan suatu tampilan 3D secara menyeluruh kelangsungan proses industri. Dassault Systèmes menawarkan solusi PLM yang terintegrasi untuk pengembangan produk. CATIA sudah diciptakan sekitar tahun 70'an dengan nama CATI oleh industri pesawat terbang Avions Marcel Dassault (AMD) yang kemudian mendirikan Dassault Systèmes ditahun 1981. Dari tahun 1982 hingga 1988 dikembangkan CATIA dari versi 1 hingga versi 3, CATIA – CADAM versi 4 ada sejak tahun 1993 dan pada bulan maret 1999, Dassault Systèmes memperkenalkan CATIA versi 5 yang ada hingga kini. Di bulan Mei tahun 2005 kini telah dikeluarkan CATIA version 5 release 15 (CATIA V5R15) dan dibulan Juni Boeing menggunakan V5R15 untuk 787 Dreamliner. Untuk lebih detail dan jelas mengenai sejarah dari CATIA dan Dassault Systèmes' PLM solution yang lain dapat dilihat referensi [3].

Secara *de facto*, CATIA sebagai standard industri dalam industri pesawat terbang, digunakan oleh sebagian besar pabrik pesawat terbang. Diintergrasikan dengan DELMIA, ENOVIA, dan SMARTEAM, memungkinkan para pemakai untuk mensimulasikan keseluruhan cakupan dari proses desain industri, dari konsep awal dan desain produk ke analisa, perakitan dan pemeliharaan. CATIA memiliki kemampuan dalam desain solid model secara *real-time*, ketika terjadi perubahan spesifikasi (karena suatu hal terjadi perubahan satu atau lebih part dari bagian pesawat yang akan diproduksi) dapat digabungkan kembali secara cepat dan efisien. Model solid menjadi melebihi definisi objek 3D (dimana karakteristik part dan informasi manufaktur didefinisikan secara realistic).

2.4 Beberapa workbench pada CATIA

Berdasarkan kelengkapan modul yang dimilikinya, CATIA terbagi atas 3 *platform*, yaitu P1, P2 dan P3. pembagian ini bertujuan agar para pelanggan dapat menyesuaikan kebutuhannya dengan modul yang ada pada CATIA, disamping semakin banyak modul yang dimiliki maka pelanggan akan membayar lebih. P3 merupakan aplikasi yang memiliki jumlah modul yang lengkap. Gambar 2.3 merupakan tampilan dari P3, tampilan ini membutuhkan permintaan akan *video card* yang tinggi.



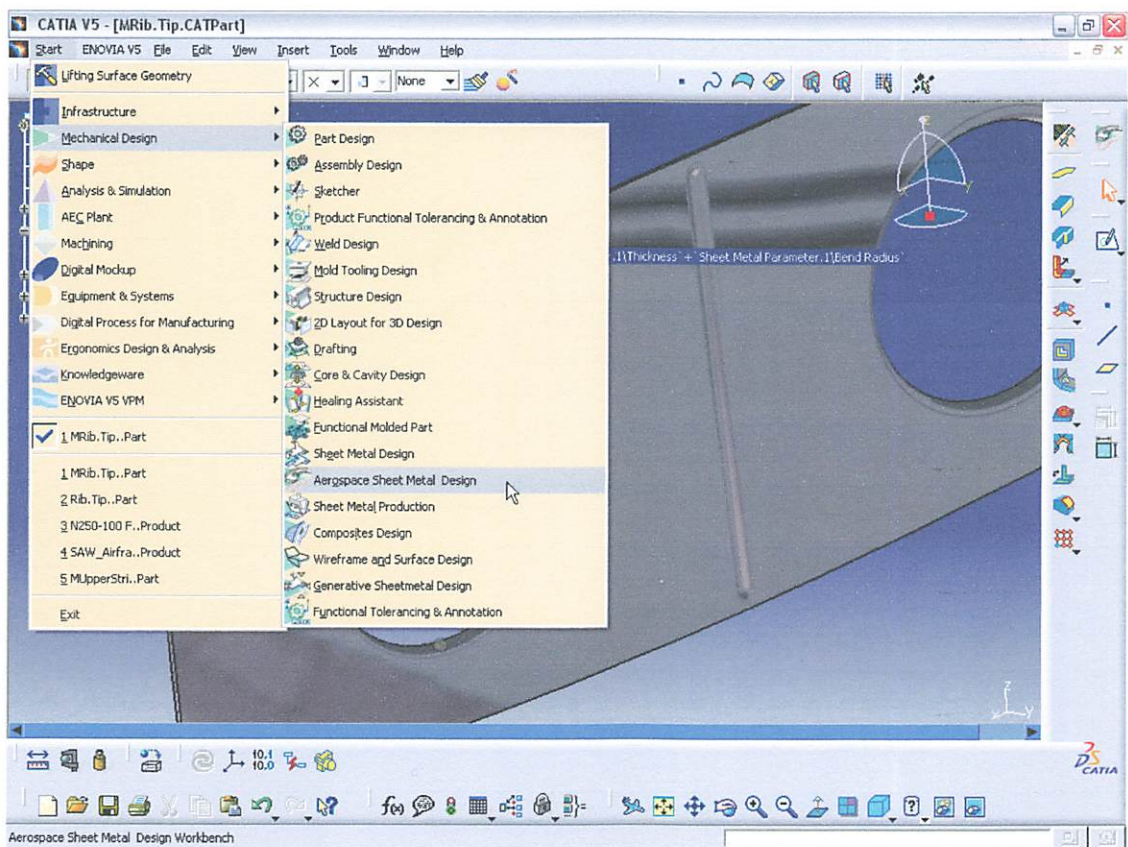
Gambar 2.3 CATIA V5 P3 user interface

Perangkat lunak CATIA dahulu hanya dapat dimiliki oleh industri-industri besar yang memiliki super computer, tetapi dengan perkembangan teknologi yang sangat cepat pada perangkat keras computer maka kini, berdasarkan referensi [2], CATIA dapat dijalankan pada PC biasa dengan persyaratan minimum perangkat kerasnya sebagai berikut :

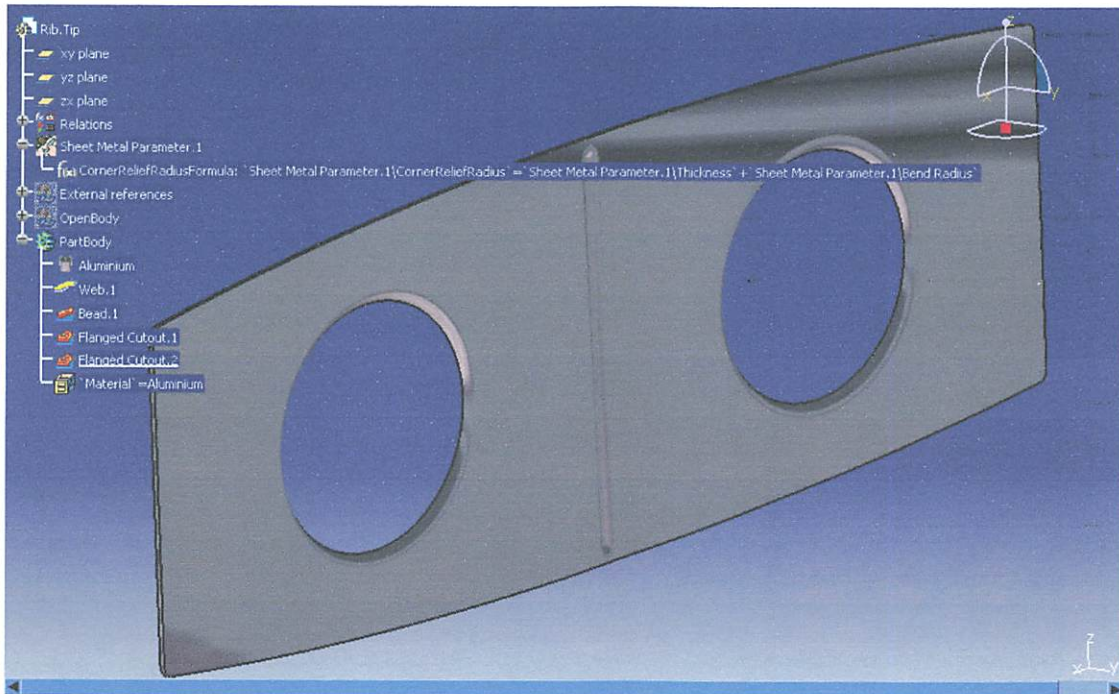
- CPU : Intel P4 2,0
- RAM : 512 MB
- Disk Space : 1GB (hanya untuk beberapa modul standar)
- VGA : 64MB.

Modul didalam CATIA terbagi atas beberapa aplikasi utama seperti *Infrastructure, Mechanical Design, Shape, Analysis & Simulation, Digital Mockup, Machining, Ergonomics Design &*

Analysis, dan beberapa yang lain. Masing-masing aplikasi utama ini terdiri atas kumpulan meja kerja (*WorkBench*) yang sejenis. Untuk tiap *WorkBench* memiliki *toolbars* yang spesifik yang sebagian besar hanya ada pada *WorkBench* itu saja. Gambar 2.4 menunjukkan jenis-jenis aplikasi utama dan *WorkBenches* yang ada pada *Mechanical Design*. Window yang tampak pada gambar 2.4 merupakan window dari *WorkBench Aerospace Sheet Metal Design* dengan macam macal toolbar yang ada di dalamnya. Seorang yang ahli dalam menggunakan CATIA kemungkinan besar tidak akan mampu menguasai seluruh workbench yang tersedia karena sangat banyaknya fungsi dan kemampuan yang diberikan oleh CATIA.

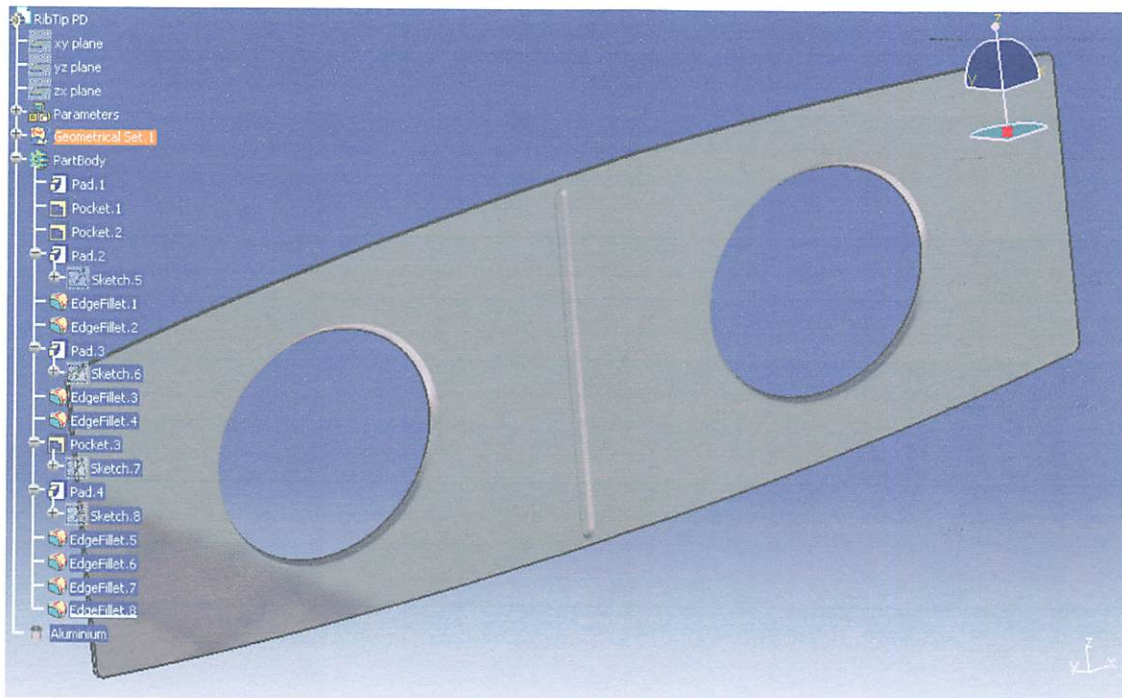


Gambar 2.4 Aplikasi utama dan workbench yang terdapat didalamnya.



Gambar 2.5 Part Rib.Tip menggunakan workbench *Aerospace Sheet Metal Design*

Untuk membuat model solid pada CATIA dapat dilakukan melalui berbagai macam toolbar pada WorkBench yang berbeda, tetapi pengguna perangkat lunak CATIA harus dapat menentukan *WorkBench* mana yang terbaik digunakan untuk produk yang didesainnya. Sebagai contoh desain sebuah rib yang ada pada gambar 2.5 merupakan hasil dari workbench *Aerospace Sheet Metal Design* sedangkan gambar 2.6 dibuat dengan menggunakan dari workbench *Part Design*.



Gambar 2.6 Part Rib.Tip menggunakan workbench *Part Design*

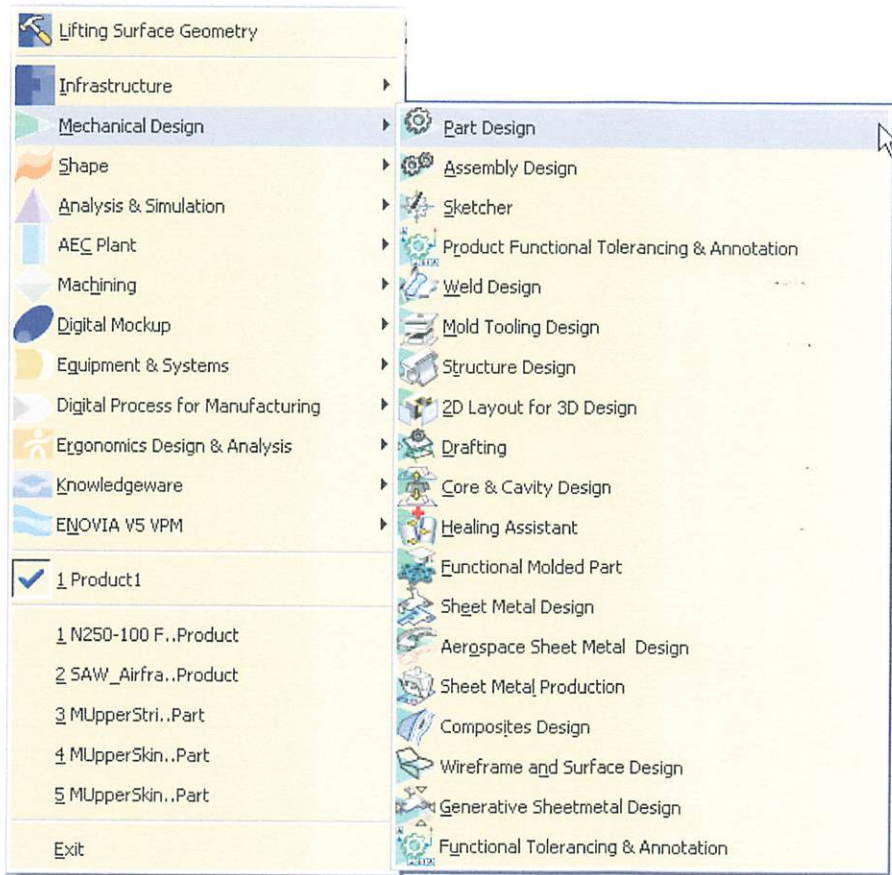
Jika hanya melihat hasil akhirnya saja, maka keduanya memiliki property struktur yang sama, seperti ukuran, massa jenis, dan momen inersianya. Tetapi jika dilihat dari proses mendesainnya sendiri dan bagaimana produk ini akan dimanufaktur maka keduanya adalah desain yang sangat berbeda. Untuk produk pada gambar 2.5 merupakan hasil dari proses pembentukan sebuah pelat tipis sedangkan untuk produk pada gambar 2.6 akan melalui proses permesinan dari sebuah balok aluminum.

Tugas akhir ini dalam mendesain model solid hanya menggunakan beberapa workbench standar yang umum digunakan.

2.4.1 Workbench Mechanical Design

Mechanical Design merupakan salah satu dari 12 aplikasi utama untuk CATIA V5R15 P3 yang digunakan dalam mengerjakan tugas akhir ini. Workbench yang terdapat dalam

Mechanical Design dapat dilihat pada gambar 2.7 dibawah. Akan dijelaskan beberapa workbench didalam Mechanical Design yang berkaitan dengan pengerjaan tugas akhir ini.



Gambar 2.7 Aplikasi utama Mechanical Design dan workbench di dalamnya

2.4.1.a Part Design

Part Design merupakan workbench standar dalam menciptakan desain model solid 3D.

Aplikasi CATIA versi 5 Part Design dibuat sangat mungkin untuk mendisain secara presisi 3D dari suatu part produk dengan *user interface* yang fleksibel dan mudah dimengerti. Aplikasi Part Design akan memungkinkan mengakomodasi kebutuhan disain untuk part dengan kompleksitas yang tinggi, dari sederhana hingga *advance*. Operasi yang terdapat pada Part Design tidak hanya hubungan antar model solid saja tetapi dapat terjadi operasi boolean dengan surface yang berasal dari workbench lain. Untuk memperoleh hasil 2D dari solid model yang sudah jadi dapat diperoleh dengan sangat mudah. Hal ini disebabkan karenan adanya keterkaitan dengan workbench lain.

2.4.1.b Workbench Assembly Design

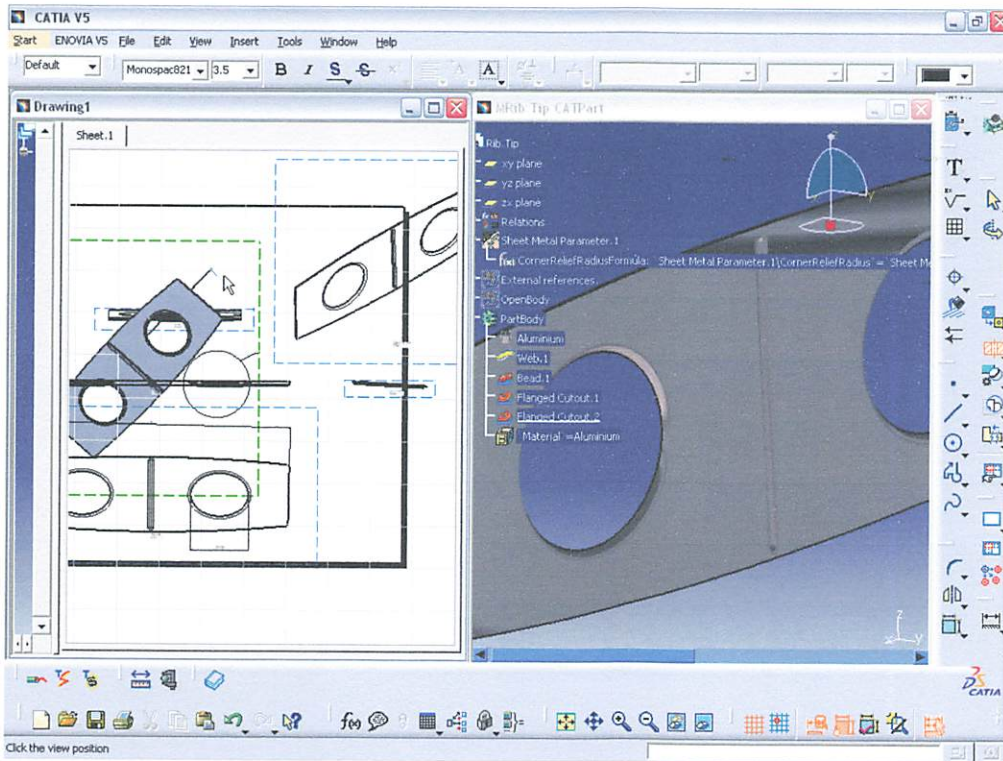
Assembly Design workbench menyediakan sarana buat pengguna untuk merakit suatu desain produk yang berasal dari berbagai jenis part. Definisi dari hubungan yang terjadi antar part dapat dilakukan dengan mudah. Analisis kontak maupun *clash* yang terjadi antar part dapat diperoleh dengan mudah, sehingga pengguna dapat mengetahui kondisi dari part jika terhadap part yang lain jika telah digabungkan menjadi suatu produk. Pada dasarnya Assembly Design workbench memiliki tipe file yang berbeda dengan Part Design, yaitu .CATProduct. Assembly Design hanya menganalisis hubungan antar part. Assembly Design tidak dapat berdiri sendiri seperti halnya .CATPart, ketika file CATProduct dibuka maka akan segera mencari posisi file CATPart yang tersimpan, jika suatu file dari part berubah ataupun dipindah maka produk pada Assembly Design workbench akan kehilangan seluruh data yang berkaitan dengan part tersebut.

2.4.1.c Workbench Sketcher

Aplikasi Sketcher sangat memungkinkan bagi para perancang untuk membuat sket profil 2D yang presisi dan cepat. Berbeda dengan Drafting Workbench yang juga menghasilkan gambar 2D, Sketcher merupakan gambar 2D yang digunakan sebagai dasar membuat model 3D, Sketcher berada dalam lingkungan 3D.

2.4.1.d Workbench Generatif Drafting

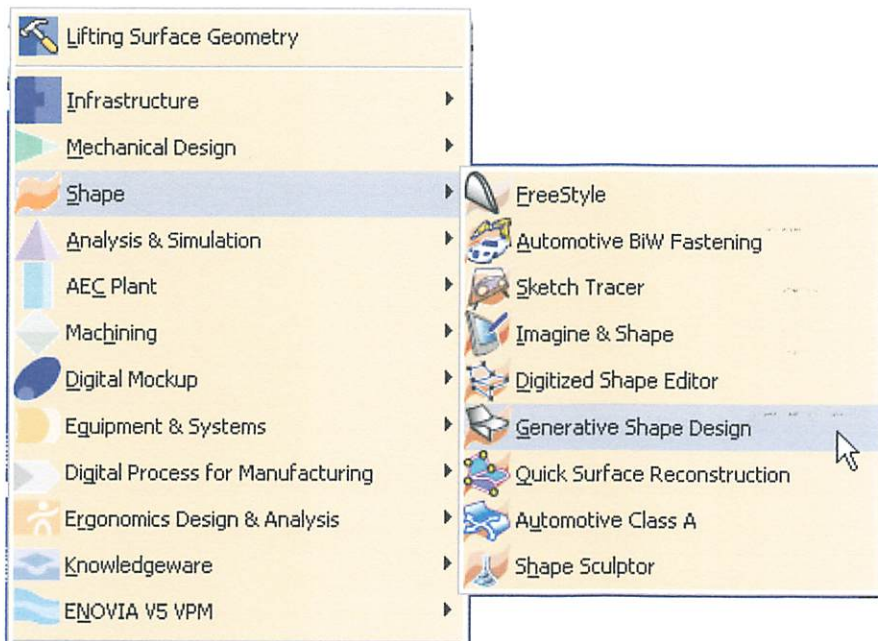
Generatif Drafting Workbench memudahkan bagi perancang untuk membuat gambar 2D dari hasil model 3D yang berasal dari Part Design maupun Assembly Design. Gambar 2.8 dibawah menunjukkan membuat gambar 2D dari model solid 3D dengan mengkombinasi Generatif Drafting Workbench dan Part Design.



Gambar 2.8 Pembuatan gambar 2D (drafting) dari hasil model 3D

2.4.2 Workbench Shape

Aplikasi utama Shape terdiri atas Workbench untuk mendesain model 3D surface atau permukaan. Workbench yang terdapat dalam Shape dapat dilihat pada gambar 2.8 dibawah.

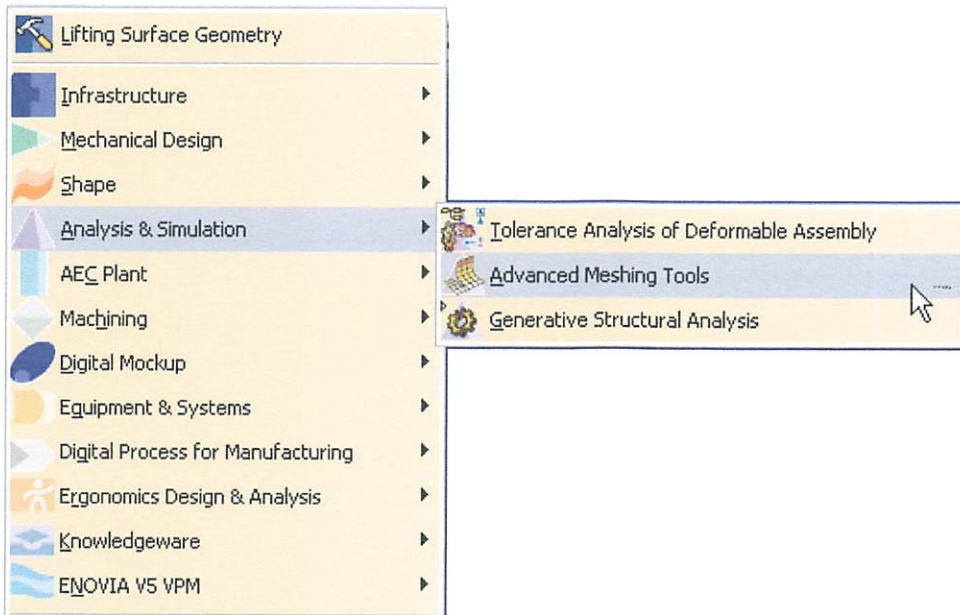


Gambar 2.9 Aplikasi utama Shape dan workbench di dalamnya

Generative Shape Design merupakan salah satu workbench yang digunakan dalam membuat model solid 3D pada tugas akhir ini. Generative Shape Design workbench memberikan sarana yang mudah dan cepat untuk mendesain model 3D *wireframe* maupun *surface* yang mudah maupun rumit. Workbench ini menyediakan banyak sekali tool untuk menciptakan maupun mengedit desain bentuk, dan ketika dikombinasikan dengan Part Design dapat digunakan untuk membentuk model solid maupun dioperasikan dengan model solid yang sudah ada.

2.4.3 Workbench Analysis & Simulation

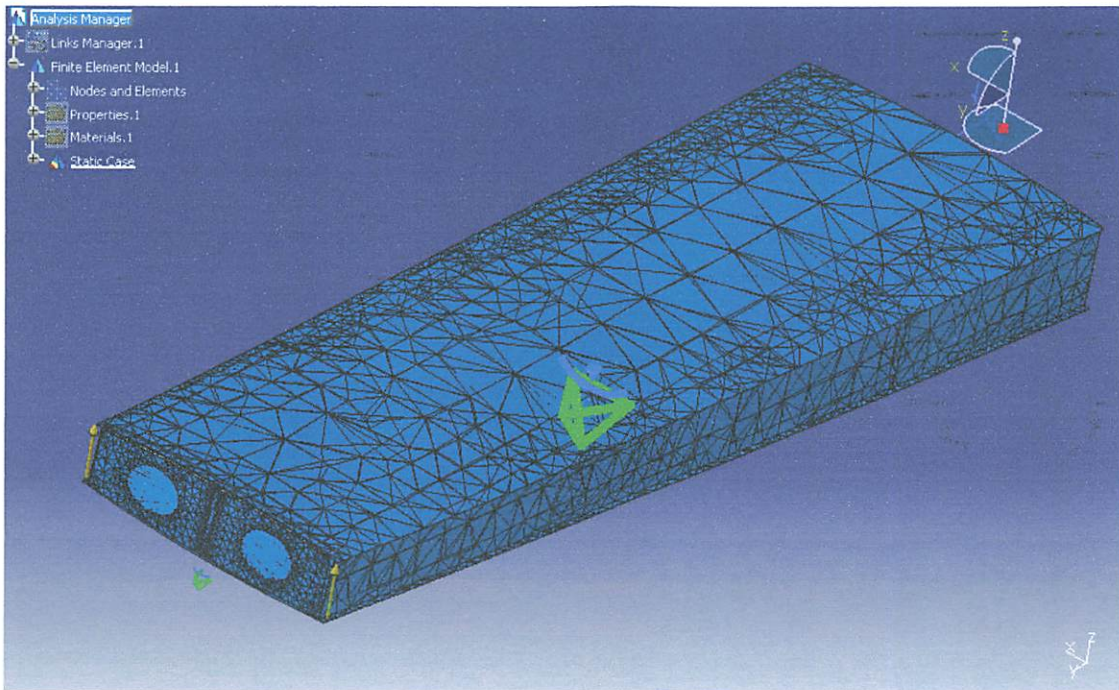
Aplikasi utama ini menyediakan semua tool yang dibutuhkan untuk spesialis dan para perancang yang melibatkan proses analisis. Proses ini meliputi stress, frequency, thermomechanical, buckling dan contact analysis dengan beban berganda. Analisis dapat dilakukan pada part tunggal maupun assembly dengan mencampurkan elemen solid, shell dan beam. Analysis & Simulation memiliki tiga workbench yang dapat dilihat pada gambar 2.10. Pada tugas akhir ini akan digunakan dua dari tiga workbench yang tersedia, yaitu *Advanced Meshing Tools* dan *Generatif Structural Analysis*.



Gambar 2.10 Aplikasi utama Analysis & Simulation dan workbench di dalamnya

2.4.3.a Workbench Advanced Meshing Tools

Advanced Meshing Tools workbench membantu dalam menghasilkan finite element model dengan cepat untuk part yang kompleks baik beam, surface maupun solid. Guna menghasilkan finite element model untuk part surface yang kompleks, pengguna harus menggunakan *FEM Surface product*. Sedangkan *FEM Solid product* untuk part solid yang kompleks.



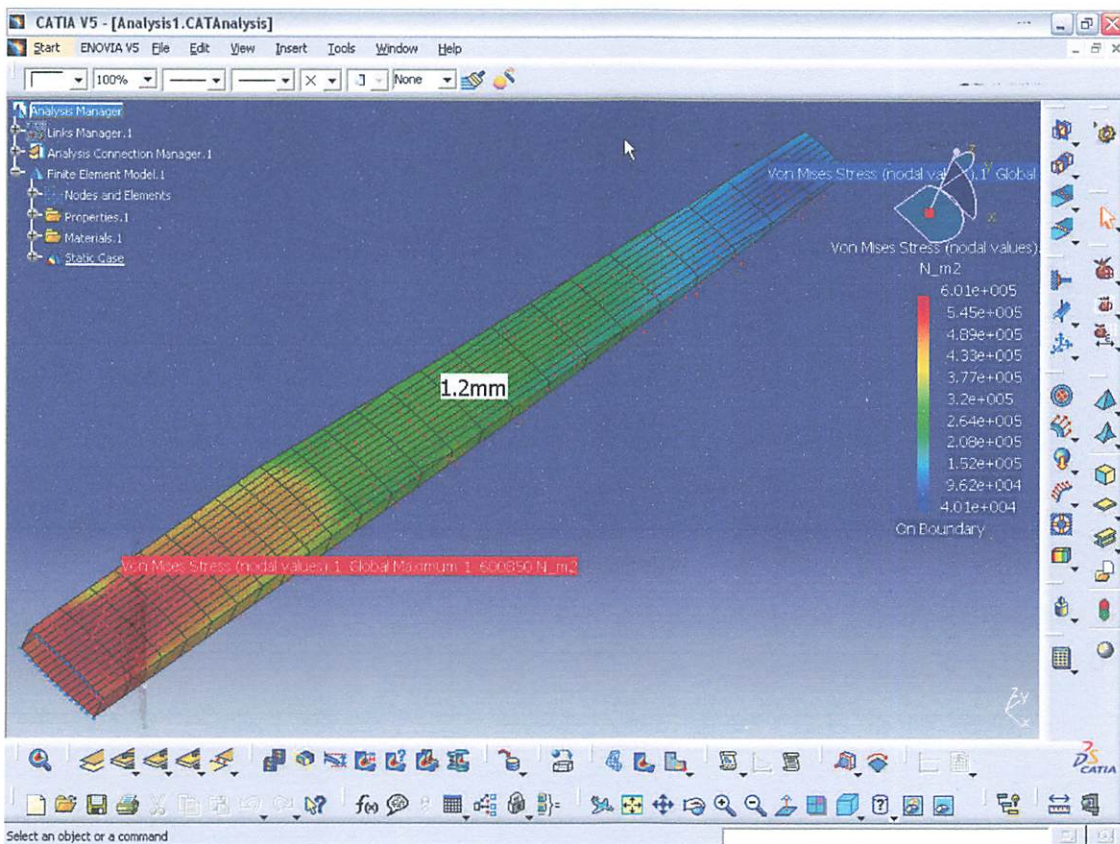
Gambar 2.11 Mesh dari model solid

2.4.3.b Workbench Generatif Struktural Analysis

ELFINI Structural Analysis, Generative Part Structural Analysis and Generative Assembly Structural Analysis membantu dalam menganalisis struktur untuk model 3D dengan cepat dan mudah. Analisis dapat dilakukan dengan model 3D dari Part Design (satu model part baik solid maupun surface) maupun yang berasal dari Assembly Design. Untuk model yang berasal dari Assembly Design maka hubungan antar part (*Constraint*) yang telah didefinisikan pada workbench tersebut sangat penting bagi pendefinisian sifat koneksi antar part pada Generatif Struktural Analysis seperti *slider*, *Contact Connection*, *Bolt*, *Fastened Connection*, *Rigid Connection*, dan masih banyak lagi. Pada prinsipnya pendefinisian *Constraint* pada Generatif Struktural Analysis ini hanya menguatkan dan memperjelas koneksi yang terjadi sehingga diistilahkan sebagai *restraint*. Jika tidak terlebih dahulu didefinisikan koneksi antar part tersebut, maka *restraint* juga tidak terdefinisi. Generatif Struktural Analysis dengan Advanced Meshing Tools memiliki kalitan yang kuat, meskipun pada Generatif Struktural Analysis workbench memiliki tool yang dapat menghasilkan model elemen. Model element pada

Generatif Struktural Analysis workbench bersifat umum, untuk memperoleh elemen yang lebih baik maka peran dari Advanced Meshing Tools sangatlah besar.

Gambar 2.12 menunjukkan window dari workbench Generatif Struktural Analysis dan jenis analisis yang dilakukan adalah static dengan model elemen kombinasi antara shell dengan beam.



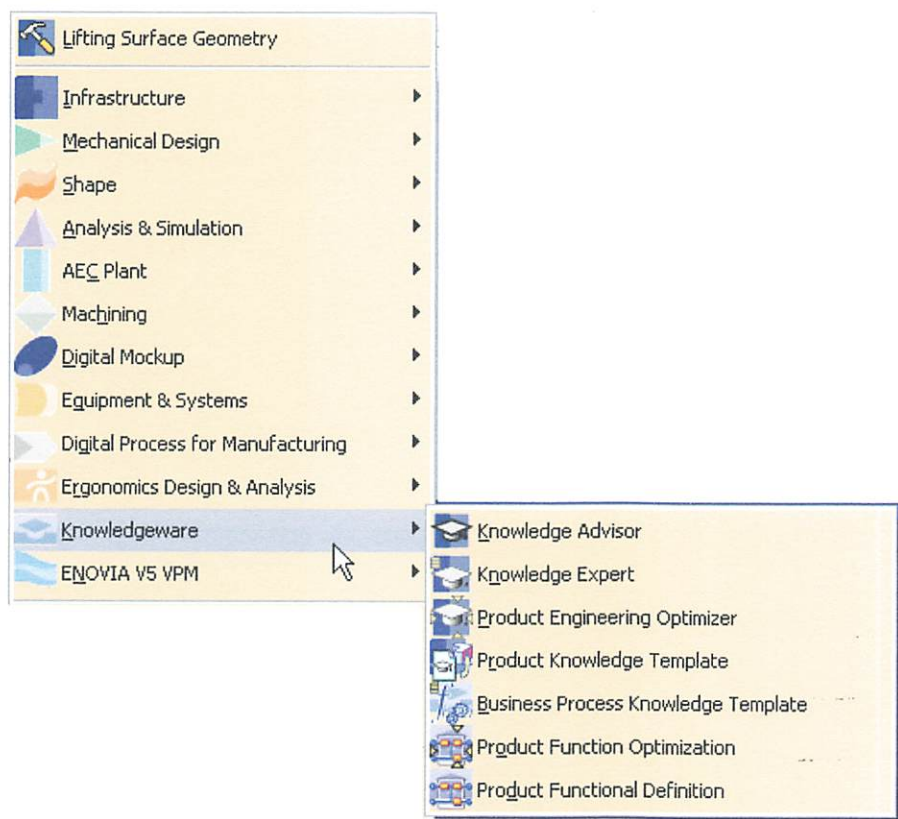
Gambar 2.12 Hasil analisis static dengan finite elemen model

2.5 Pemrograman / Automation pada CATIA

CATIA memiliki aplikasi yang menunjang untuk dilakukan proses automasi. Aplikasi yang telah tersedia sebagai standar modul yang diberikan pada CATIA V5 yaitu, *Knowledgeware*, *Macros* dan *Visual Basic for Application (VBA)* serta aplikasi untuk automasi yang ditambahkan ke CATIA sebagai *embedded software*, CAA / CAA RADE.

2.5.1 Workbench Knowledgeware

Knowledgeware merupakan salah satu aplikasi yang tersedia di CATIA guna melakukan proses optimisasi. Knowledgeware membantu pengguna didalam outomasi desain, mengurangi kesalahan dan untuk maksimum produktifitas. Dengan Knowledgeware para pengguna dapat memasukkan formula atau rumusan, aturan dan cek yang dapat digunakan setiap saat. Knowledgeware bekerja menurut definisi yang telah diberikan oleh pengguna sehingga dapat dengan mudah di akses kembali dan dipahami. Workbench yang tergabung dalam Knowledgeware dapat dilihat pada gambar 2.13 dibawah.



Gambar 2.13 Aplikasi utama Knowledgeware dan workbench di dalamnya

2.5.2 Macros / .CATScripts

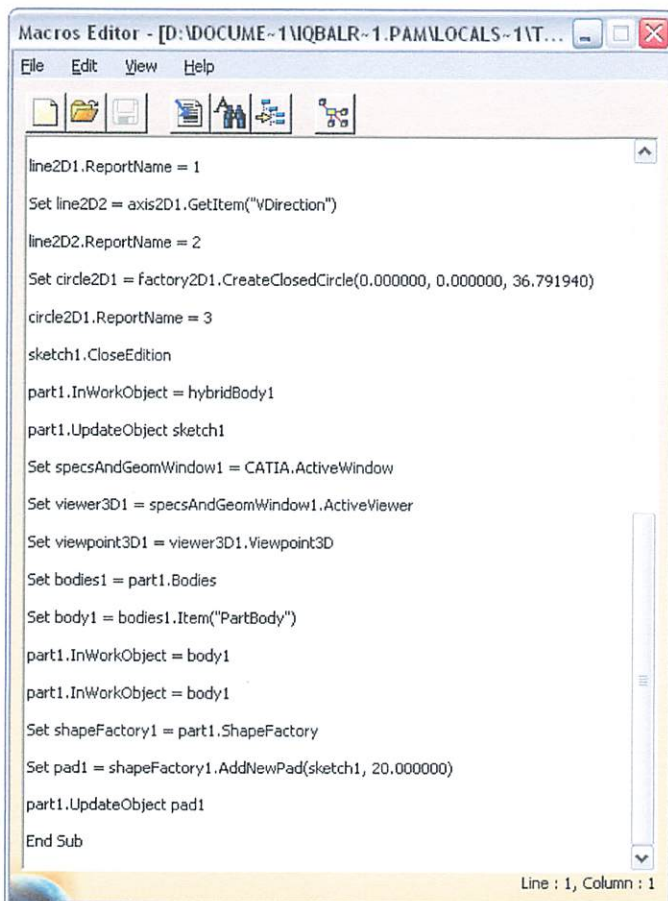
Macros / CATScripts juga merupakan fasilitas untuk melakukan proses automasi pada CATIA. Jika pengguna ingin melaksanakan tugas secara berulang ulang kali, pengguna dapat mengambil keuntungan dari macros tersebut untuk mengautomasiukannya. Macros merupakan suatu deretan fungsi yang ditulis dengan VB Script maupun CATScript.

Sebagai contoh automasi menggunakan macros yaitu;

- Membuat suatu deretan lubang di part.
- Menyaring daftar material dari suatu assembli
- Mencetak (print) serangkaian dokumen.

Untuk membuat macros sangatlah mudah yaitu dengan merekam (*recording*) suatu urutan operasi yang dilakukan ataupun memodifikasi file macros yang sudah ada sesuai dengan kebutuhan. Gambar 2.14 merupakan window *macros editor* hasil dari proses rekaman operasi yang dilakukan, terlihat sederetan fungsi yang ditulis menggunakan VBScript. Di *macros editor* inilah dilakukan pengeditan untuk diperoleh perintah baru yang diinginkan.

Macros memiliki kekurangan dalam hal fleksibilitas untuk melakukan automasi dan proses *debug* –nya sangat sulit.

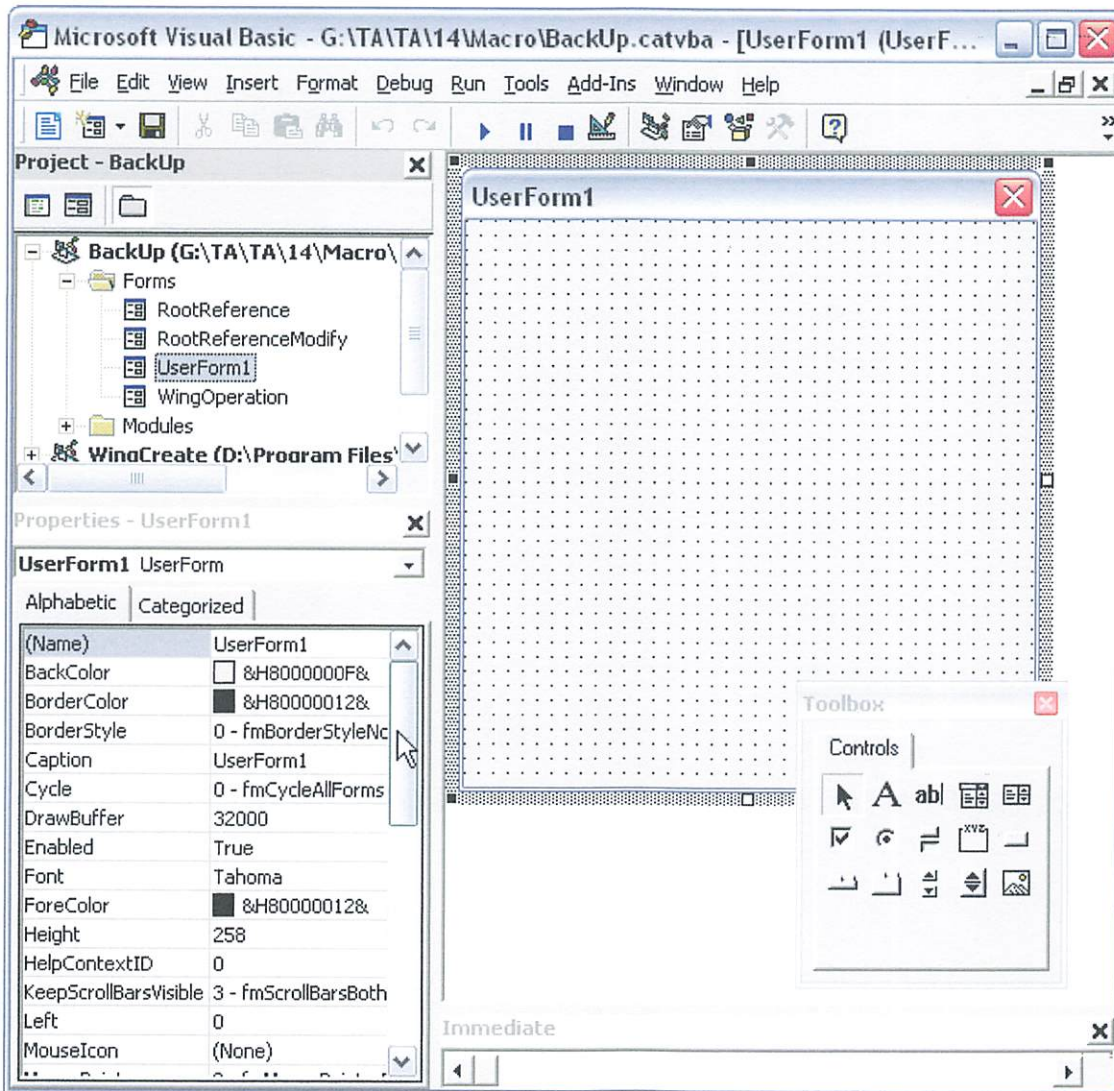


Gambar 2.14 Window Macros editor

2.5.3 Visual basic for Application (VBA) / .CATvba

Visual basic for Application (VBA) / .CATvba merupakan aplikasi yang telah tersedia di CATIA V5 dan digunakan untuk menyusun tugas akhir ini. Jauh berbeda dengan macros, pada VBA, pengguna diberi fasilitas untuk mendesain *user interface (forms)* bagi program automasi yang dibuat. VBA memiliki editor yang menunjang kemampuan *debugging* dan lebih fleksibel dalam melakukan automasi.

Gambar 2.15 menunjukkan window dari *Microsoft Visual Basic Editor*. Visual basic for Application (VBA) juga dapat dibuat dengan proses perekaman dari operasi yang dilakukan dan kemudian dimodifikasi untuk melakukan proses yang diinginkan. Proses perekaman tidak menghasilkan suatu bentuk *UserForm* tertentu, tetapi seperti halnya macros, yang dihasilkan berupa sederetan perintah (*command*) yang ditulis menggunakan bahasa VB. Hasil dari proses perekaman ini tertulis sebagai *module* dari file .CATvba yang telah ditentukan lokasinya terlebih dahulu.



Gambar 2.15 Window VBA editor

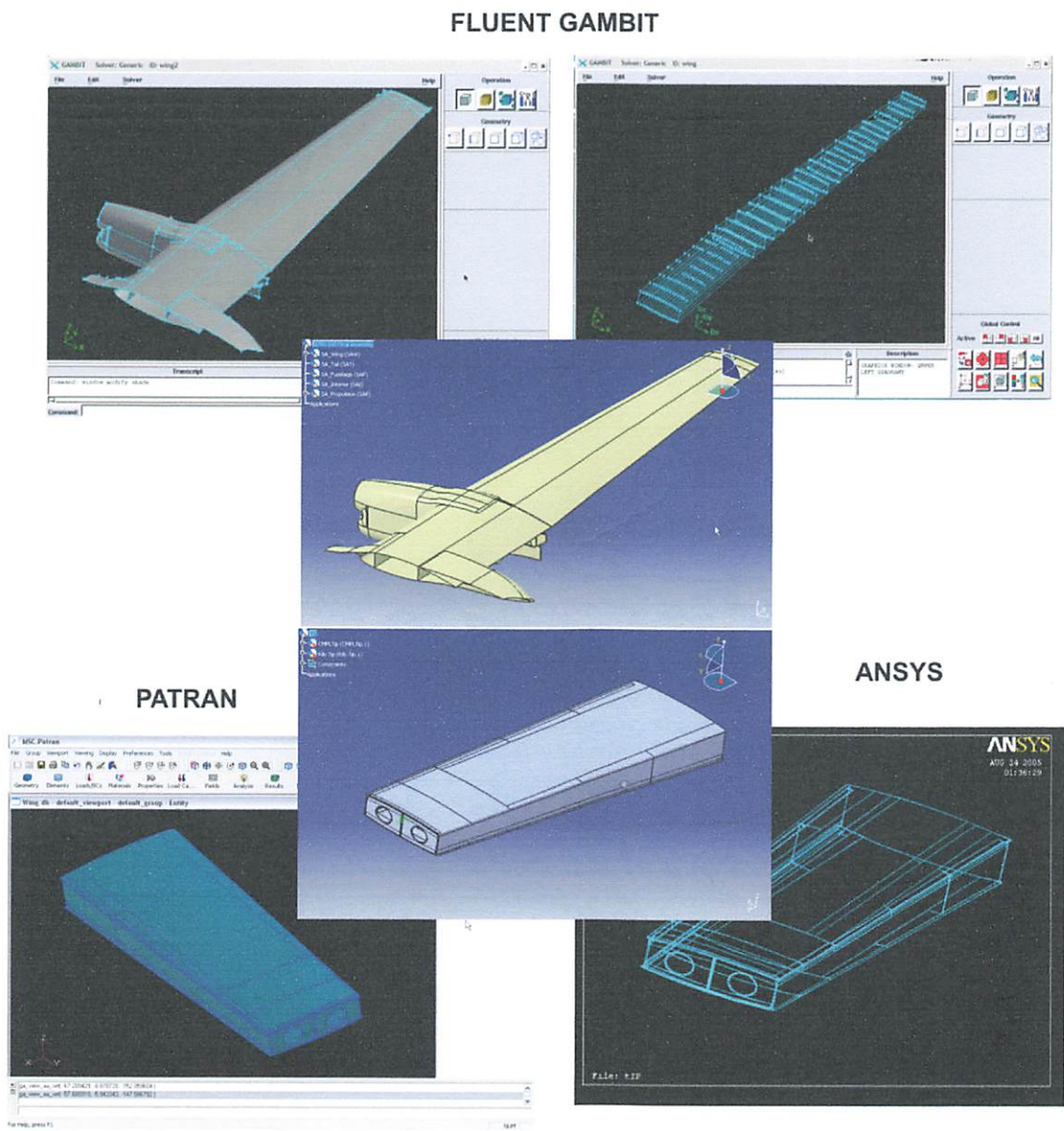
2.5.4 CAA

Satu lagi fasilitas yang dapat digunakan untuk melakukan automasi di CATIA yaitu CAA. CAA merupakan aplikasi tambahan yang terpisah dari CATIA, ketika aplikasi perangkat lunak CAA RADE di *install* kedalam suatu computer maka secara otomatis perangkat lunak ini masuk ke dalam CATIA (*embedded*).

CAA jauh lebih cepat dan *powerful* dibandingkan Visual Basic tetapi lebih sulit untuk dipelajari. Bahasa pemrograman yang digunakan untuk automasi di CATIA oleh CAA adalah bahasa C++.

2.6 Interaksi CATIA dengan software lain

CATIA memiliki kemampuan untuk mengkonversi file yang dia hasilkan menjadi tipe file yang dapat dibaca oleh perangkat lunak yang lain, begitu juga sebaliknya, CATIA mampu membaca tipe file dari perangkat lunak lain. Kemampuan seperti ini yang menyebabkan CATIA menjadi lebih mudah berinteraksi dengan perangkat lunak yang lain.



Gambar 2.16 Model CATIA dapat diterima oleh perangkat lunak lain

BAB 3

Pemrograman Lifting Surface Geometry menggunakan VBA

3.1 Pendahuluan

Dalam perancangan awal (*coceptual design*) konfigurasi pesawat udara, seorang perancang biasanya akan menentukan desain yang optimal, sehingga sering kali dilakukan perubahan. Untuk membuat suatu sayap pesawat terbang secara manual, seorang perancang akan sangat membutuhkan waktu yang lama pada saat memasukkan tipe airfoil disetiap sekmennya. Data airfoil berupa koordinat yang berjumlah sangat banyak, kemudian data tersebut dimasukkan dalam gambar secara manual sebagai *point* dan kemudian point tersebut saling dihubungkan dengan menggunakan objek *spline*. Dapat dibayangkan berapa lama proses ini dilakukan, terlebih jika selalu terjadi perubahan dalam penentuan airfoil maka perancang akan memulai menggambar dari awal lagi. Oleh sebab itu dibutuhkan outomasi untuk mempermudah proses ini.

Kelebihan CATIA yang telah dijelaskan bab dan sub bab sebelumnya, sepertihalnya *update* dan hubungan *parent and children* menyebabkan model yang dibuat tidak hanya sebagai model yang hanya menampilkan keindahan saja tetapi juga sebagai model untuk analisis dan proses proses lain di CATIA. Analisis pada CATIA akan dibahas pada bab berikutnya

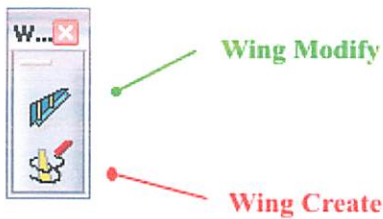
Aplikasi pemodelan ini adalah sebuah aplikasi kecil yang terintegrasi dengan CATIA V5, dia hanya dapat dieksekusi di dalam lingkungan kerja CATIA V5. Automasi yang dilakukan pada tugas akhir ini menggunakan fasilitas *Visual Basic for Aplication* (VBA) yang dimiliki oleh CATIA. Dan diujicoba untuk CATIA V5R12 dan 14, sedangkan tugas akhir ini menggunakan CATIA V5R15.

3.2 Antarmuka Antara Aplikasi Pemodelan dengan CATIA

Aplikasi tambahan ini toolbar dan window dialog supaya interaksi antara user dan aplikasi menjadi lebih mudah (*user friendly*). Lifting Surface Geometry merupakan proses automasi untuk mendisain sayap pesawat terbang. Aplikasi ini digunakan untuk menciptakan maupun memodifikasi sayap pesawat terbang. Modifikasi hanya dapat dilakukan secara otomatis untuk model sayap yang dibuat dengan aplikasi pasangannya. Jika masih dalam operasi menciptakan

model sayap maka modifikasi saat itu juga dapat dilakukan tanpa harus keluar terlebih dahulu. Tetapi jika telah keluar dari operasi *Wing Create* maka proses modifikasi harus dengan *Wing Modify*.

Antarmuka awal aplikasi ini berupa dua buah toolbar tambahan yang dapat dimasukkan ke dalam workbench yang diinginkan ataupun dipisahkan menjadi satu buah workbench tersendiri. Gambar 3.1 merupakan tampilan dari kedua toolbar tersebut

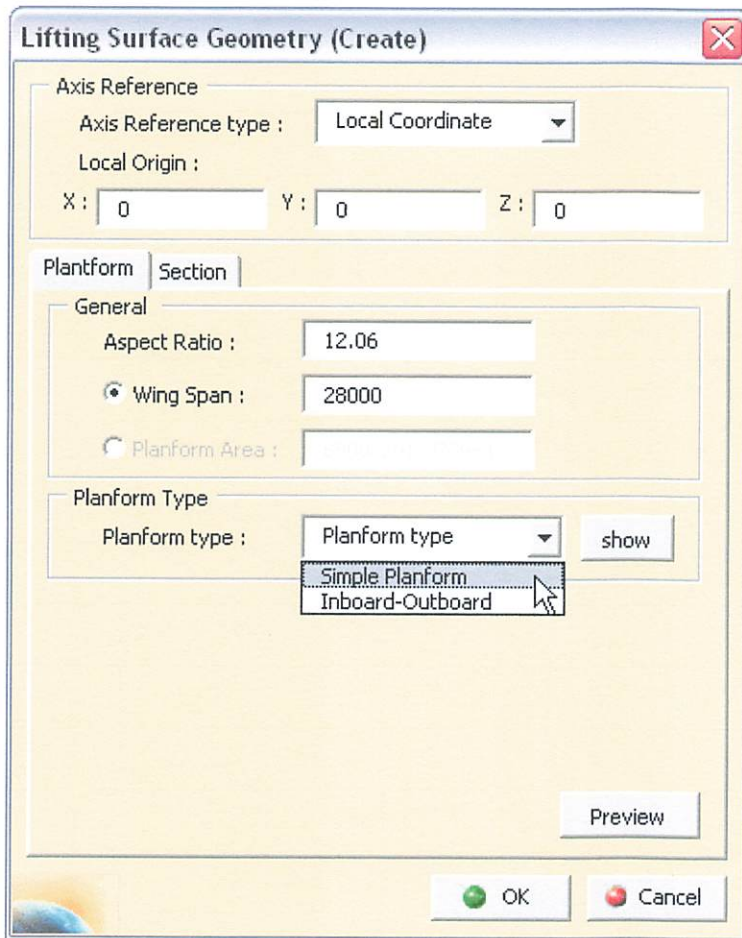


Gambar 3.1 Toolbar

Aplikasi ini mampu mendesain dua tipe sayap, yaitu *simple wing* dan *inboard-outboard wing*. Sayap yang terbentuk hanya sayap sebelah kanan dari pesawat terbang. Posisi tipe airfoil yang dimasukkan bukanlah tepat sejajar pada root, kink, maupun tip tetapi tegak lurus *leading edge* dari masing masing sekmen. Sehingga airfoil yang ada sepanjang *section* merupakan transformasi linear dari airfoil tiap sekmen

3.2.1 Simple Wing

Simple Wing merupakan tipe sayap dengan posisi pemasangan airfoil di Root dan Tip saja. Airfoil sepanjang root menuju tip merupakan perubahan secara linear dari airfoil root dan tip. Gambar 3.2 memperlihatkan *user interface* dari *Lifting Surface Geometry (Create)* pada tab *Platform*.

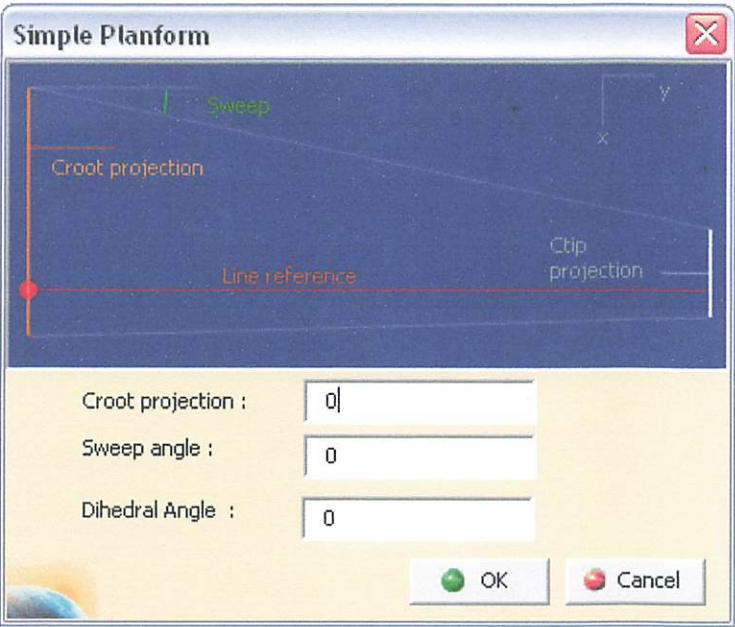


Gambar 3.2 Window dialog lifting serface geometry (Create), tab plantform, dengan pilihan *Simple Planform*

Harga yang terlebih dahulu dimasukkan adalah lokasi *Local coordinate* untuk sayap. Posisi *Local Coordinate* ini berada pada *leadingedge* dari root.

Pada tab *Plantform* user memasukkan input berupa besar *Aspect Ratio*, panjang sayap (*Wing Span*) ataupun dapat memilih luas penampang dari sayap (*Plantform Area*). Ketika harga panjang bentang sayap dimasukkan, secara otomatis harga dari luas penampang sayap akan muncul. Tombol *preview* pada tab *plantform* merupakan tombol untuk melakukan proses pembuatan model dari *plantform* sayap yang didesain. Tetapi terlebih dahulu user harus menentukan tipe dari sayap yang didesain. Jika telah ditentukan tipe dari sayap maka akan muncul *user interface* baru sesuai dengan tipe tersebut.

Gambar 3.3 merupakan gambar dari window dialog *Simple Plantform* yang berisikan data data baru yang bersesuaian dengan tipe ini dan harus dimasukkan oleh user.

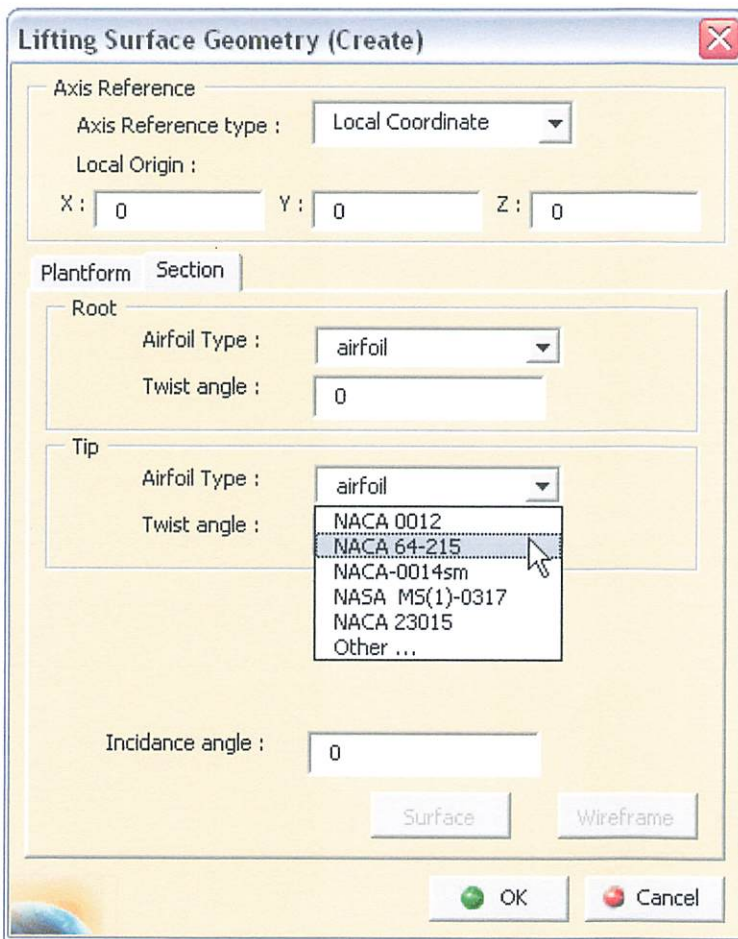


Gambar 3.3 Window dialog simple plantform

Data data yang perlu dimasukkan di dalam window dialog yang perlu dimasukkan adalah data yang dibutuhkan dalam mendesain sayap bertipe *Simple*, yaitu proyeksi panjang *chord* di root (*Croot projection*), sudut *sweep*, dan sudut *dihedral*.

Jika model plantform sayap yang didesain telah terbentuk (dengan menekan tombol *preview*) maka selanjutnya membuka tab *Section*. Perlu diperhatikan bahwa tombol *wireframe* dan *surface* tidak aktif atau dengan kata lain pembuatan model *wireframe* maupun *surface* tidak dapat dilakukan jika model *plantform* tidak dibuat terlebih dahulu.

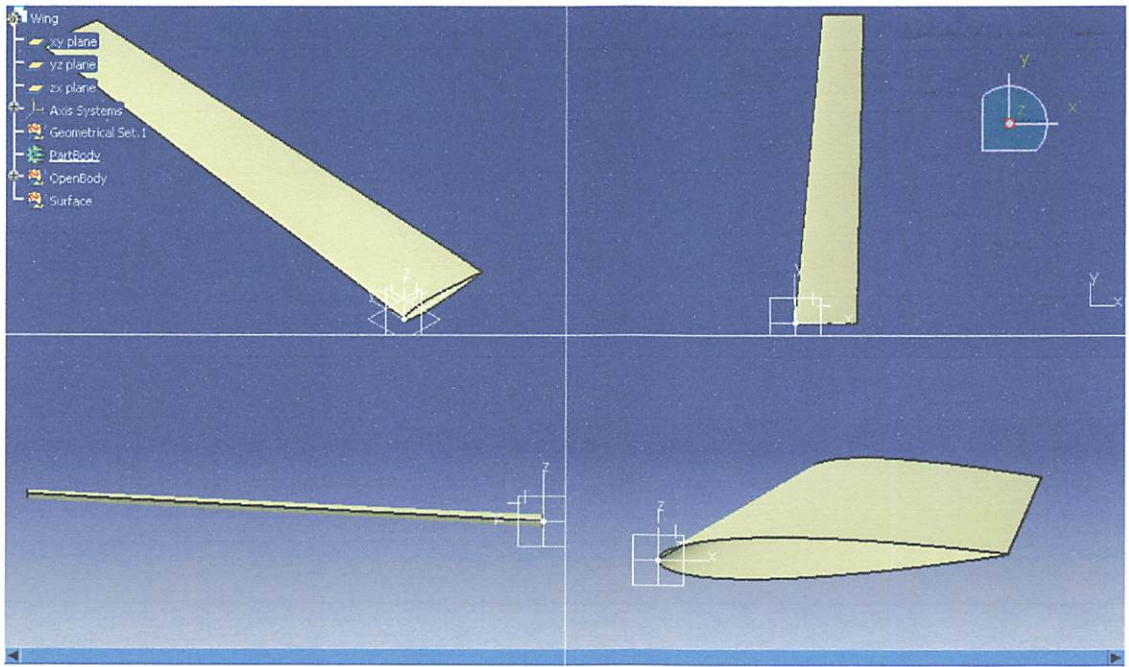
Tampilan dari tab *Section* dapat dilihat dari gambar 3.4. data yang perlu dimasukkan dalam tab *Section* adalah tipe airfoil root dan tip serta sudut twist masing-masing. Disamping itu ada data sudut pemasangan sayap (*incidence angle*) yang juga perlu dimasukkan.



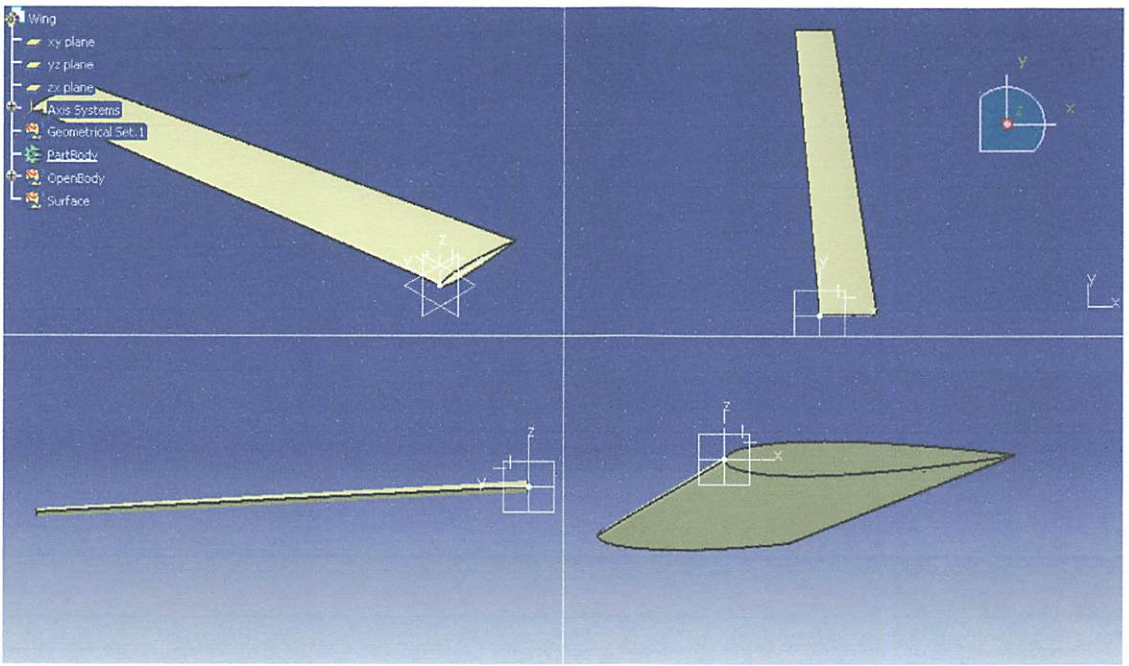
Gambar 3.4 Window dialog lifting surface geometry (Create), tab- *Section*, tipe Simple Planform

Jika semua data yang dibutuhkan sudah ada maka model kedua yang perlu di bentuk ialah *Wireframe* dengan menekan tombolnya. Tombol *surface* masih tidak aktif hingga proses pembentukan model *wireframe* selesai. kemudian tekan tombol *Surface* untuk pembentukan model *surface*. User diperkenankan keluar dari proses ketika hanya sampai pembentukan *planform* ataupun *surface*. Jika ingi meneruskan model yang baru setengah jadi (*planform* ataupun *surface*) lagi, maka operasi yang dipilih adalah *Wing Create*.

Gambar 3.5 adalah contoh dari hasil operasi yang telah dilakukan untuk sayap dengan sudut *sweep* dan *dihedral positif*. Dan gambar 3.6 adalah contoh dari hasil operasi yang telah dilakukan untuk sayap dengan sudut *sweep* dan *dihedral negatif*



Gambar 3.5 Sayap dengan sudut *sweep* dan *dihedral positif*

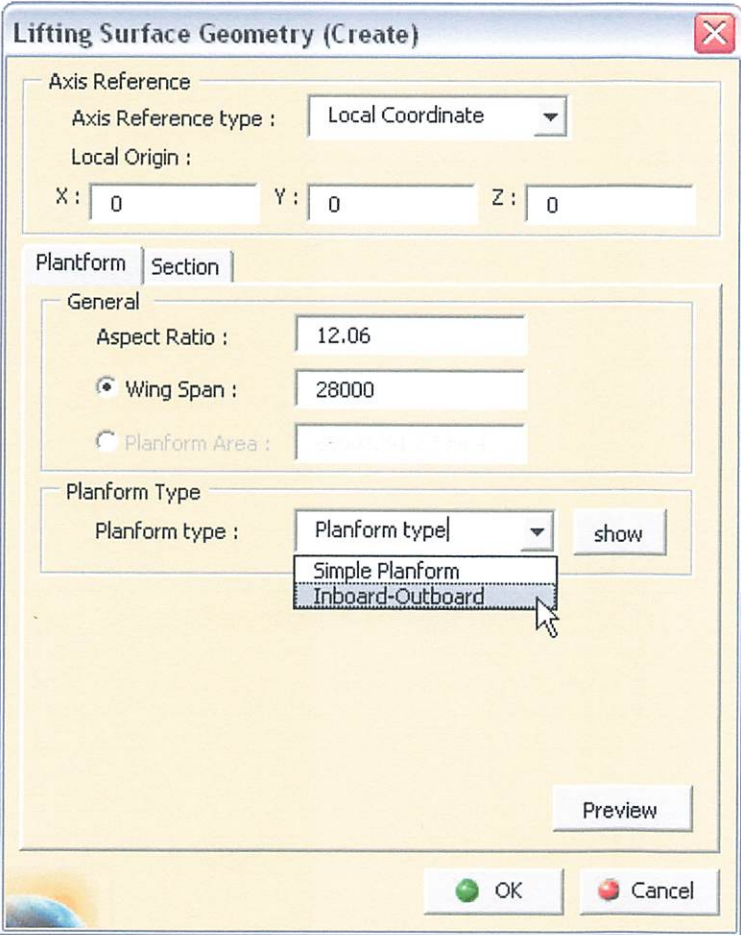


Gambar 3.6 Sayap dengan sudut *sweep* dan *dihedral negative*

3.2.2 Inboard-Outboard Wing

Inboard-Outbiard Wing merupakan tipe sayap dengan posisi pemasangan airfoil di Root Kink dan Tip. Airfoil sepanjang root menuju kink merupakan perubahan secara linear dari airfoil root dan kink, begitupula dari kink ke tip.

Gambar 3.7 memperlihatkan *user interface* dari *Lifting Surface Geometry (Create)* pada tab *Plantform*.

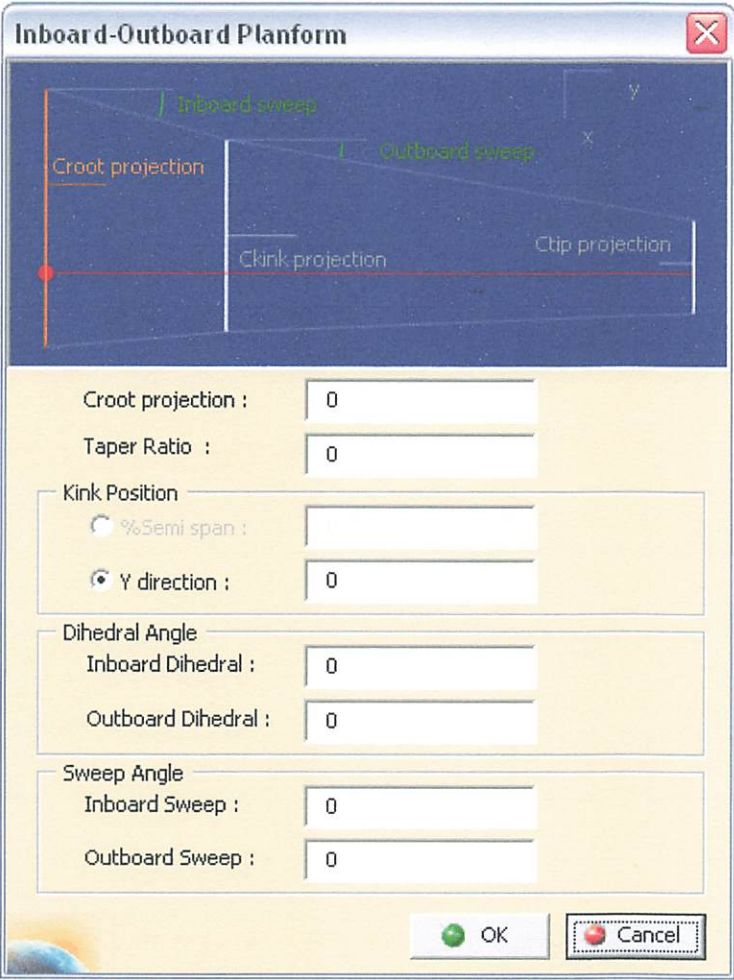


Gambar 3.7 Window dialog lifting serface geometry (Create), tab plantform, dengan pilihan *Inboard-Outboard*

Harga yang terlebih dahulu dimasukkan adalah lokasi *Local coordinate* untuk sayap. Posisi *Local Coordinate* ini berada pada *leadingedge* dari root.

Pada tab *Plantform* user memasukkan input berupa besar *Aspect Ratio*, panjang sayap (*Wing Span*) ataupun dapat memilih luas penampang dari sayap (*Plantform Area*). Ketika harga panjang bentang sayap dimasukkan, secara otomatis harga dari luas penampang sayap akan muncul. Tombol *preview* pada tab *plantform* merupakan tombol untuk melakukan proses pembuatan model dari *plantform* sayap yang didesain. Tetapi terlebih dahulu user harus menentukan tipe dari sayap yang didesain. Jika telah ditentukan tipe dari sayap maka akan muncul *user interface* baru sesuai dengan tipe tersebut.

Gambar 3.8 merupakan gambar dari window dialog *Inboard-Outboard Plantform* yang berisikan data data baru yang bersesuaian dengan tipe ini dan harus dimasukkan oleh user.

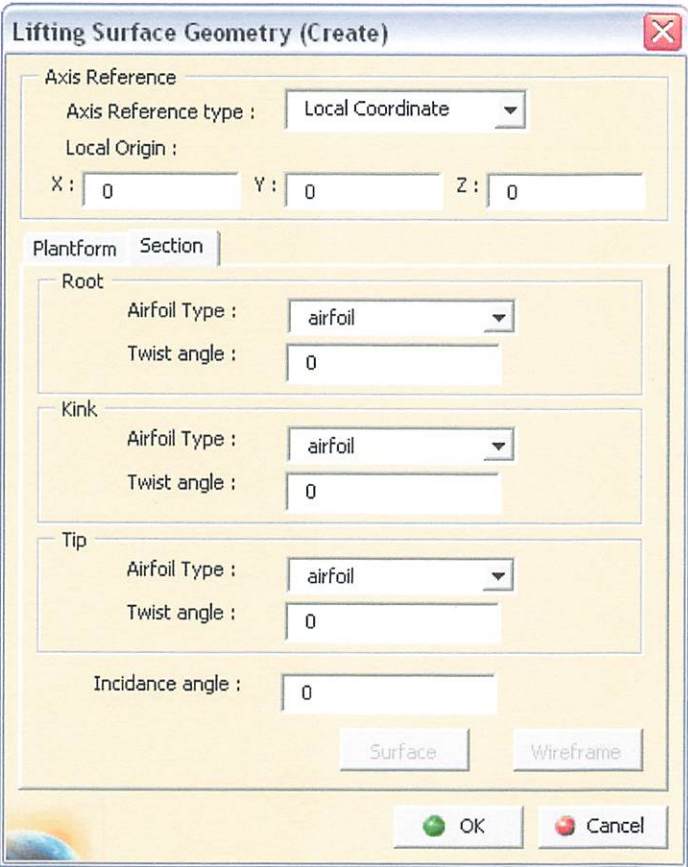


Gambar 3.8 Window dialog Inboard-Outboard plantform

Data data yang perlu dimasukkan di dalam window dialog adalah data yang dibutuhkan dalam mendesain sayap bertipe *Simple*, yaitu proyeksi panjang *chord* di root (*Croot projection*), *Taper Ratio*, posisi kink (*kink position*) yang dapat berupa persentasi dari panjang setengah bentang sayap atau jarak pada arah sumbu Y dari root (*Y direction*) sudut *sweep* untuk inboard maupun outboard, dan sudut *dihedral* untuk inboard maupun outboard juga.

Jika model plantform sayap yang didesain telah terbentuk (dengan menekan tombol *preview*) maka selanjutnya membuka tab *Section*. Perlu diperhatikan bahwa tombol *wireframe* dan *surface* tidak aktif atau dengan kata lain pembuatan model *wireframe* maupun *surface* tidak dapat dilakukan jika model *plantform* tidak dibuat terlebih dahulu.

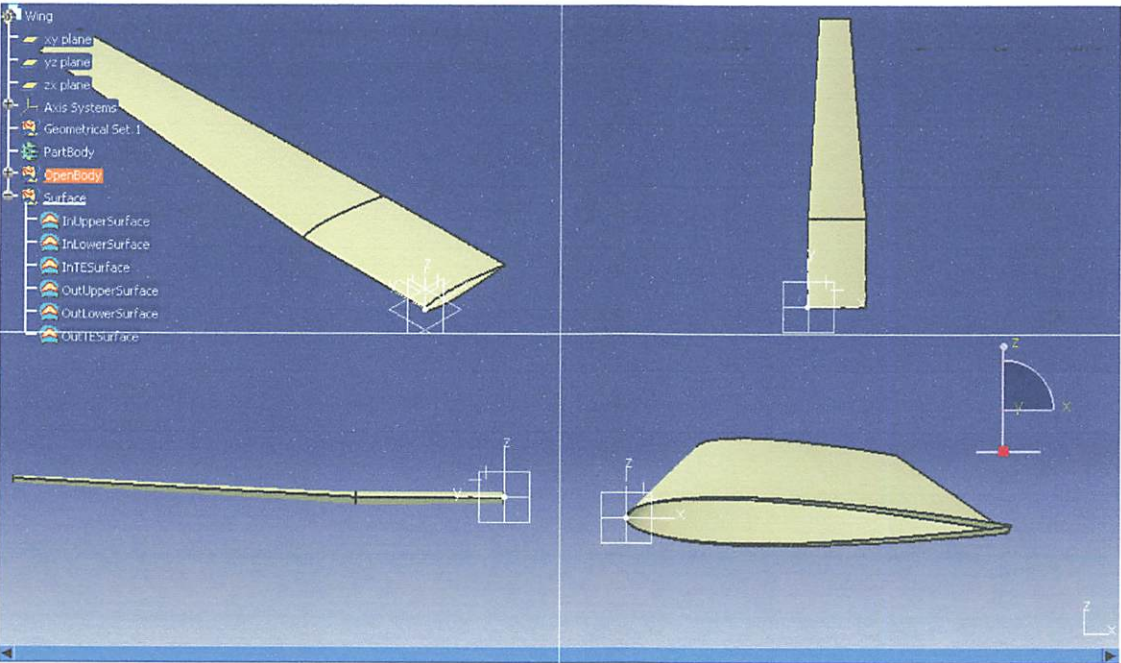
Tampilan dari tab *Section* dapat dilihat dari gambar 3.9. Data yang perlu dimasukkan dalam tab *Section* adalah tipe airfoil root, kink dan tip serta sudut twist masing-masing. Disamping itu ada data sudut pemasangan sayap (*incidence angle*) yang juga perlu dimasukkan.



Gambar 3.9 Tab- *Section*, tipe Inboard-Outboard

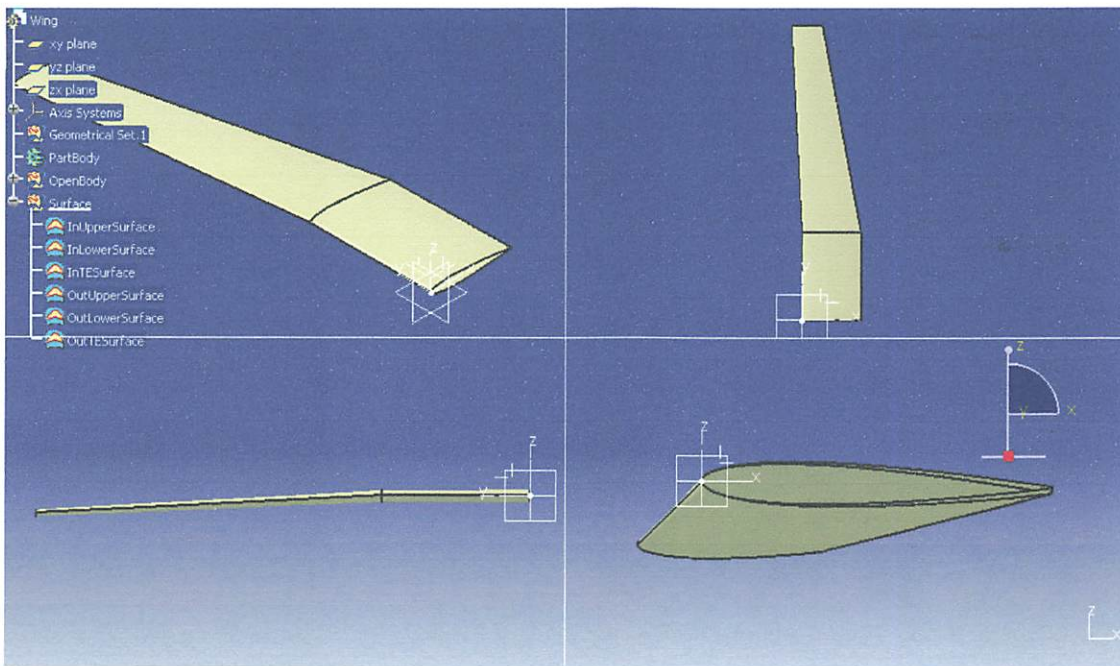
Jika semua data yang dibutuhkan sudah ada maka model kedua yang perlu di bentuk ialah *Wireframe* dengan menekan tombolnya. Tombol *surface* masih tidak aktif hingga proses pembentukan model *wireframe* selesai. kemudian tekan tombol *Surface* untuk pembentukan model *surface*. User diperkenankan keluar dari proses ketika hanya sampai pembentukan *planform* ataupun *surface*. Jika ingi meneruskan model yang baru setengah jadi (*planform* ataupun *surface*) lagi, maka operasi yang dipilih adalah *Wing Create*.

Gambar 3.10 adalah contoh dari hasil operasi yang telah dilakukan untuk sayap N250-100PN. Dan gambar 3.11 adalah contoh dari hasil operasi yang telah dilakukan untuk sayap dengan sudut *sweep outboard* dan *dihedral outboard negatif*



Gambar 3.10 Sayap N250-100PN

Specification tree yang dihasilkan dari operasi aplikasi ini telah dikelompokkan kedalam *OpenBody* yang berisi *wireframe* dan *Surface*.



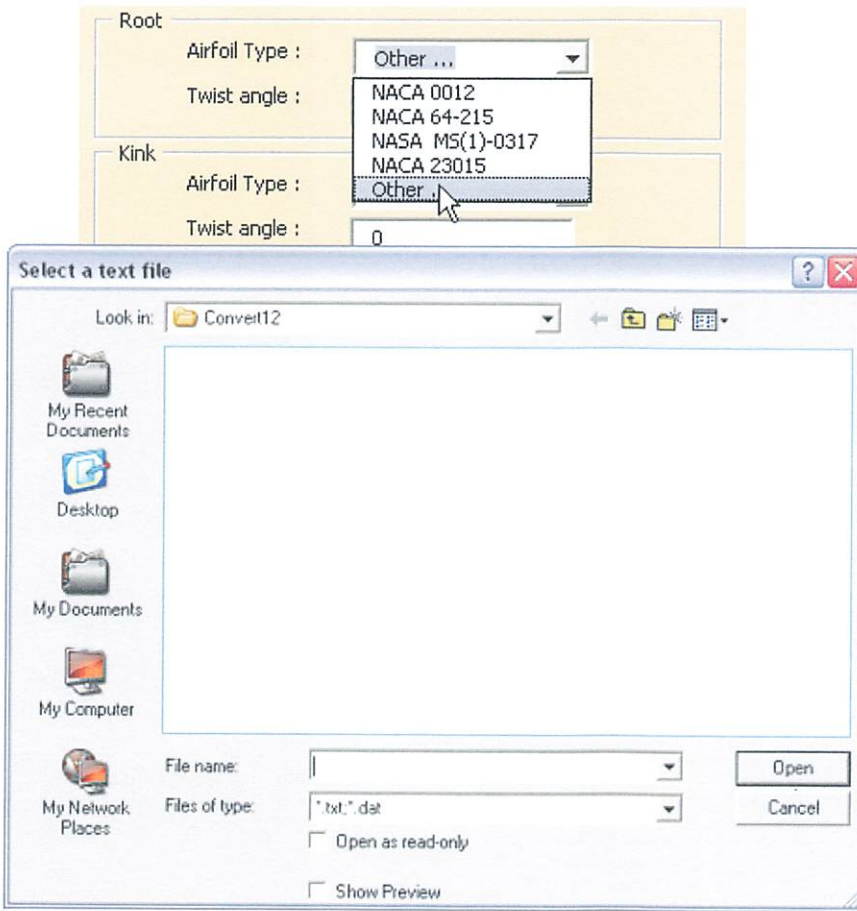
Gambar 3.11 Sayap denagn sudut *sweep outboard* dan *dihedral outboard negative*

Pada Operasi *Wing Modify* memiliki user interface yang sama dengan *Wing Create*, hanya saja data dari dokumen yang dimodifikasi telah masuk kedalam kolom kolom yang seharusnya diisi ketika melakukan operasi *Wing Create*.

3.3 Airfoil database

User dari aplikasi ini jika ingin menggunakan data airfoil lain karena tidak terdapat dalam database yang tersedia di aplikasi, maka user dapat menggunakan pilihan “other...” yang telah tersedia di daftar pilihan tipe airfoil. Kemudian akan muncul window dialog select a text file yang digunakan pengguna untuk mencari letak database airfoil baru miliknya.

Tipe file yang dapat di baca oleh aplikasi ini adalah text dengan extension .DAT ataupun .txt. Gambar 3.12 memperlihatkan cara untuk mencari dan membaca file data base airfoil baru.

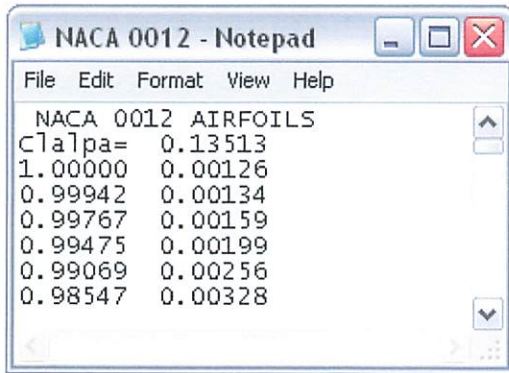


Gambar 3.12 window dialog select a text file

Untuk dapat dibaca baik oleh aplikasi , maka databesa airfoil yang dibuat haruslah mengikuti aturan format yang ditetapkan, yaitu :

- Baris pertama merupakan nama dari tipe airfoil.
- Baris kedua adalah harga dari koefisien gaya angkat airfoil, besarnya gaya angkat dapat dimasukkan ataupun tidak.
- Baris ketiga hingga terakhir merupakan data koordinat airfoil X dan Y dengan skala 0 hingga 1.
- Jumlah angka penyusun koordinat airfoil tiap sumbunya sebanyak 7 termasuk titik.
- Jarak antara angka terakhir dari koordinat sumbu X dengan angka awal sumbu Y sebesar 2 spasi.
- Jumlah data koordinat airfoil masing masing 100 buah untuk Upper maupun Lower.

Contoh dari format data airfoil dapat dilihat pada gambar 3.13.



Gambar 3.13 Contoh format data airfoil

3.3 Model matematika

Dalam penulisan program aplikasi ini banyak menggunakan persamaan matematika geometry dalam ruang, yaitu: *plane equation*, *coordinate transformation equation*,

Line equation, *line-plane intersection*, sebagainya.keempat persamaan yang telah disebutkan dapat dilihat pada gambar 3.14 hingga 3.17.

Persamaan-persamaan ini terletak di dalam program dengan kondisi tidak aktif sehingga tidak mempengaruhi *debugging*. Peletakan persamaan tersebut dimaksudkan agar mempermudah memahami dan mengedit sintaknya.

```

*****
PLANE EQUATION

normal Vector a = <x2 - x1,y2 - y1,z2 - z1>
An equation of the plane through P1(x1,y1,z1) with normal Vector a = <a1,a2,a3> is
    a1(x - x1)+a2(y - y1)+a3(z - z1) = 0
or
    ax + by + cz + d = 0 ,with normal Vector <a,b,c>
if the plane through P1(x1,y1,z1),P2(x2,y2,z2),P3(x3,y3,z3) then
    ==> Vector P1P2 = <(x2-x1),(y2-y1),(z2-z1)>
    ==> Vector P1P3 = <(x3-x1),(y3-y1),(z3-z1)>
    ==> P1P2xP1P3 = | (x2-x1) (y2-y1) (z2-z1) |
                   | (x3-x1) (y3-y1) (z3-z1) |

then the normal Vector solution :
    ==> a = [(y2-y1)*(z3-z1)]-[(y3-y1)*(z2-z1)]
    ==> b = -[(x2-x1)*(z3-z1)]-[(x3-x1)*(z2-z1)]
    ==> c = [(x2-x1)*(y3-y1)]-[(x3-x1)*(y2-y1)]

for were RootVector1:
    1 = origin()
    2 = KinkLEPointPlanformDihedralP()
    3 = RTEPointTwist()
for were RootVector_2:
    1 = origin()
    2 = KinkLEPointPlanformDihedralP()
    3 = RootVector_1() = RootVector1 + origin()
*****

```

Gambar 3.14 plane equation

```

*****
INBOARD COORDINATES TRANSFORMATION EQUATION
'
Axis direction at Local Coordinate relative to Global Coordinate :
Axis      Vector
'
X      =      a1      b1      c1
'
Y      =      a2      b2      c2
'
Z      =      a3      b3      c3
'
Point in Local Coordinates : P (x,y,z)
Point coordinates relative to Global Coordinate : P' (x',y',z')
'
x'      =      a1*x + a2*y + a3*z + Xlocal origin
'
y'      =      b1*x + b2*y + b3*z + Ylocal origin
'
z'      =      c1*x + c2*y + c3*z + Zlocal origin
'
*****

```

Gambar 3.15 Coordinate transformation equation

```

*****
LINE EQUATION
LINE THROUGH TWO POINTS
P1(x1, y1, z1) and P2(x2, y2, z2).
We need to find a vector parallel to the line by using the vector between the two points:
P1P2 = i[x2 - x1] + j[y2 - y1] + k[z2 - z1], or
a = a1i + a2j + a3k
a = <a1,a2,a3>
Now use the components of a to express the equation of the line (using P1):
Parametric:
====> x = x1 + a1t ;
      y = y1 + a2t ;
      z = z1 + a3t

Symmetric: (Solve for t in each of the above)

====> t = (x-x1)/a1 = (y-y1)/a2 = (z-z1)/a3
*****

```

Gambar 3.16 Line equation

```

*****
LINE - PLANE INTERSECTION
POINT of INTERSECTION of LINE and PLANE:
Find the point of intersection, A,
of the line
      L: x = x1 + a1t, y = y1 + a2t, z = z1 + a3t
with the plane
      P: ax + by + cz = d
We note that at the point A, the coordinates must satisfy both the equation of the
For the line at A, we have
====> xA = x1 + a1tA, yA = y1 + a2tA, zA = z1 + a3tA
      |
For the plane at A, we have
====> axA + byA + czA = d

Now combine these to find tA:
====> a*[x1 + a1tA] + b*[y1 + a2tA] + c*[z1 + a3tA] = d
      a*x1 + a*a1tA + b*y1 + b*a2tA + c*z1 + c*a3tA = d
      tA*[a*a1 + b*a2 + c*a3] = d - a*x1 - b*y1 - c*z1, or

====> tA = (d - a*x1 - b*y1 - c*z1) / [a*a1 + b*a2 + c*a3]

Now use this value to find the coordinates of A:
====> xA = x1 + a1*tA,
      yA = y1 + a2*tA,
      zA = z1 + a3*tA
*****

```

Gambar 3.17 Line-plane intersection equation

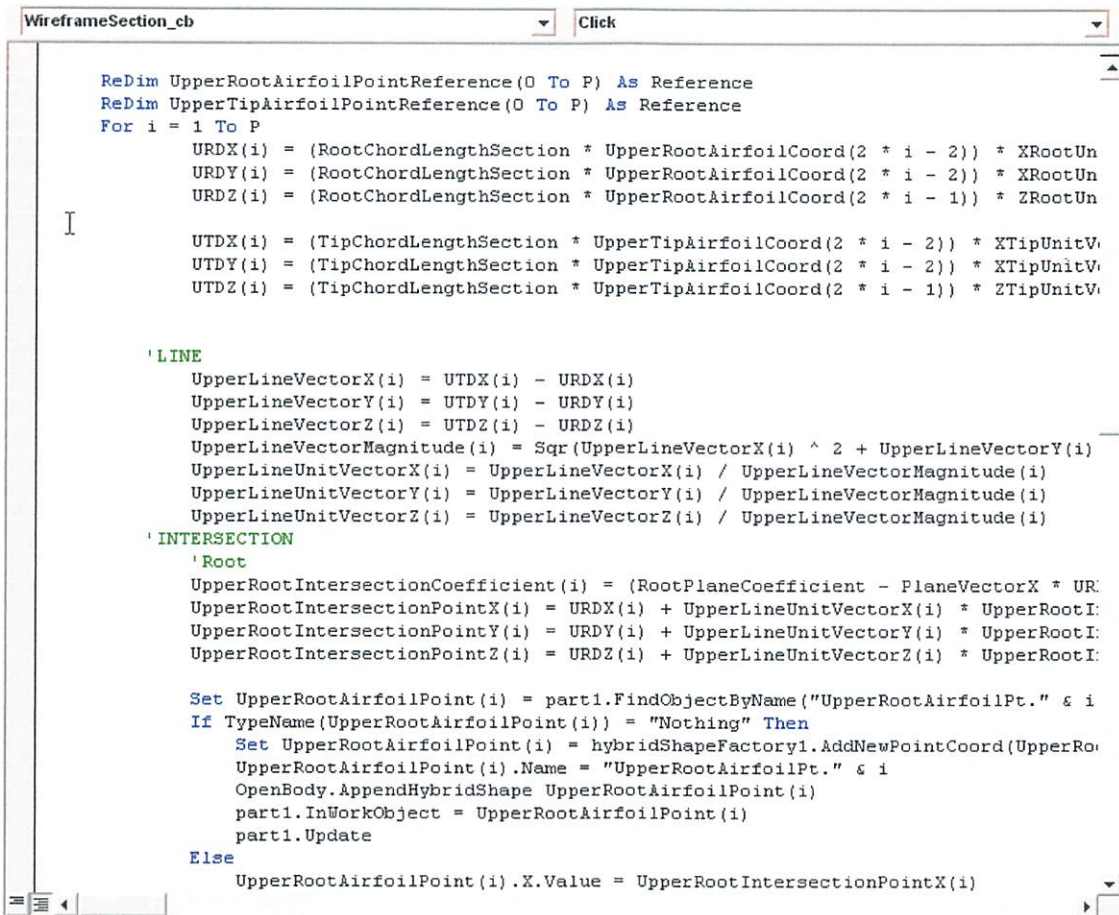
3.4 Penyusunan titik koordinat airfoil.

Aplikasi tambahan ini beroperasi cukup lama dan berat karena dipengaruhi oleh banyaknya iterasi yang dilakukan, terutama iterasi dalam proses penempatan koordinat airfoil.

Koordinat airfoil yang berada pada bidang XZ ditransformasikan kebidang yang tegak lurus dengan leading edge dengan menggunakan *Coordinate transformation equation* sehingga diperoleh koordinat airfoil yang baru. Point baru yang terbentuk di root, di kink dan di tip saling dihubungkan (root dengan kink dan kink dengan tip) sehingga diperoleh persamaan garis (dengan menggunakan *line equation*) dalam ruang. Dengan menggunakan *line-plane intersection equation* maka diperoleh koordinat titik perpotongan antara garis (hasil dari penggunaan line equation) dengan bidang yang sejajar bidang XZ, yang berada di root, tip maupun kink.

Titik perpotongan ini yang kemudian dijadikan model point yang muncul di CATIA.

Contoh proses iterasi dapat dilihat dari gambar 3.18



```

WireframeSection_cb Click
ReDim UpperRootAirfoilPointReference(0 To P) As Reference
ReDim UpperTipAirfoilPointReference(0 To P) As Reference
For i = 1 To P
    URDX(i) = (RootChordLengthSection * UpperRootAirfoilCoord(2 * i - 2)) * XRootUn
    URDY(i) = (RootChordLengthSection * UpperRootAirfoilCoord(2 * i - 2)) * XRootUn
    URDZ(i) = (RootChordLengthSection * UpperRootAirfoilCoord(2 * i - 1)) * ZRootUn

    UTDX(i) = (TipChordLengthSection * UpperTipAirfoilCoord(2 * i - 2)) * XTipUnitV
    UTDY(i) = (TipChordLengthSection * UpperTipAirfoilCoord(2 * i - 2)) * XTipUnitV
    UTDZ(i) = (TipChordLengthSection * UpperTipAirfoilCoord(2 * i - 1)) * ZTipUnitV

    'LINE
    UpperLineVectorX(i) = UTDX(i) - URDX(i)
    UpperLineVectorY(i) = UTDY(i) - URDY(i)
    UpperLineVectorZ(i) = UTDZ(i) - URDZ(i)
    UpperLineVectorMagnitude(i) = Sqr(UpperLineVectorX(i) ^ 2 + UpperLineVectorY(i) ^ 2 + UpperLineVectorZ(i) ^ 2)
    UpperLineUnitVectorX(i) = UpperLineVectorX(i) / UpperLineVectorMagnitude(i)
    UpperLineUnitVectorY(i) = UpperLineVectorY(i) / UpperLineVectorMagnitude(i)
    UpperLineUnitVectorZ(i) = UpperLineVectorZ(i) / UpperLineVectorMagnitude(i)

    'INTERSECTION
    'Root
    UpperRootIntersectionCoefficient(i) = (RootPlaneCoefficient - PlaneVectorX * URDX(i) - PlaneVectorY * URDY(i) - PlaneVectorZ * URDZ(i)) / (PlaneVectorX * UpperLineUnitVectorX(i) + PlaneVectorY * UpperLineUnitVectorY(i) + PlaneVectorZ * UpperLineUnitVectorZ(i))
    UpperRootIntersectionPointX(i) = URDX(i) + UpperLineUnitVectorX(i) * UpperRootIntersectionCoefficient(i)
    UpperRootIntersectionPointY(i) = URDY(i) + UpperLineUnitVectorY(i) * UpperRootIntersectionCoefficient(i)
    UpperRootIntersectionPointZ(i) = URDZ(i) + UpperLineUnitVectorZ(i) * UpperRootIntersectionCoefficient(i)

    Set UpperRootAirfoilPoint(i) = part1.FindObjectByName("UpperRootAirfoilPt." & i)
    If TypeName(UpperRootAirfoilPoint(i)) = "Nothing" Then
        Set UpperRootAirfoilPoint(i) = hybridShapeFactory1.AddNewPointCoord(UpperRootIntersectionPointX(i), UpperRootIntersectionPointY(i), UpperRootIntersectionPointZ(i))
        UpperRootAirfoilPoint(i).Name = "UpperRootAirfoilPt." & i
        OpenBody.AppendHybridShape UpperRootAirfoilPoint(i)
        part1.InWorkObject = UpperRootAirfoilPoint(i)
        part1.Update
    Else
        UpperRootAirfoilPoint(i).X.Value = UpperRootIntersectionPointX(i)
    End If
Next i

```

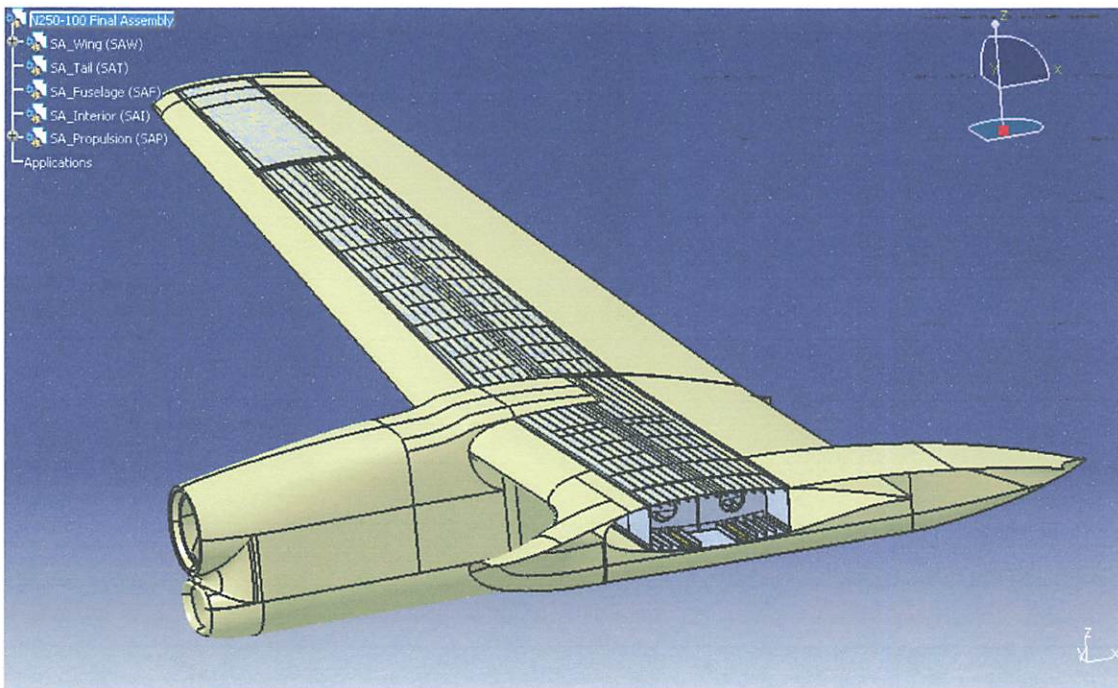
Gambar 3.18 Contoh program untuk proses iterasi

BAB 4

PEMODELAN TIGA DIMENSI (3D) UNTUK STRUKTUR WINGBOX

Pada tugas akhir ini dilakukan pemodelan 3 dimensi sayap N250-100PN. Pemodelan dimulai dari pembentukan surface dari sayap, kemudian pembentukan model solid dari wingboxnya, yang meliputi *spar cap*, *spar web*, *skin*, *stringer*, *rib cap*, dan *rib web*. Part dari *upper skin* dan *lower skin* di buat mengikuti bentuk surface dari sayap

Model sayap N250-100PN merupakan proses *assembly* dari sejumlah besar part yang berupa model solid maupun surface. System *assembly* yang dilakukan akan dibahas lebih terperinci di bagian assembly dari bab ini.

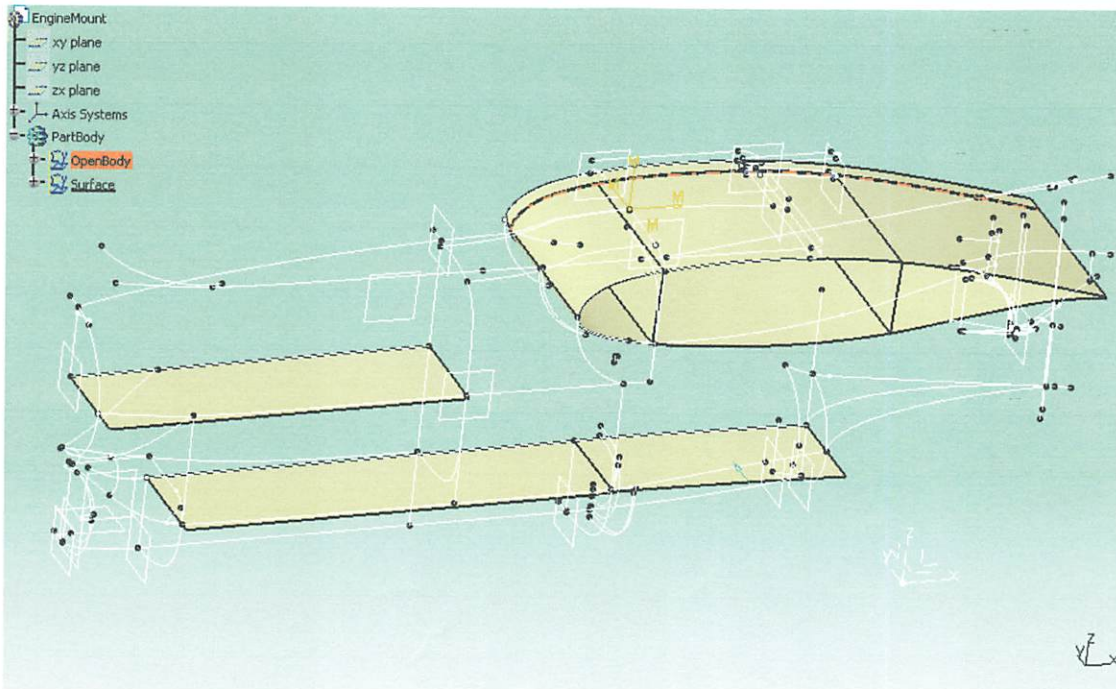


Gambar 4.1 Assembly dari seluruh sayap secara keseluruhan

Metoda umum yang digunakan untuk pembuatam model tiga dimensi yaitu *wire-frame modeling*, *surface modeling*, dan *solid modeling*. Di CATIA metoda tersebut dapat dilakukan dengan banyak sekali *tool*, masing workbench memiliki tool yang berbeda untuk melakukan metoda yang sama

4.1 Wire-frame modeling

Model *wire-frame* merepresentasikan batas-batas permukaan suatu obyek dengan garis, lingkaran, dan kurva lainnya. Suatu model tiga dimensi ditampilkan dengan representasi batas permukaan yang berupa garis-garis yang menyerupai kawat.



Gambar 4.2 Wireframe

4.2 Surface modeling

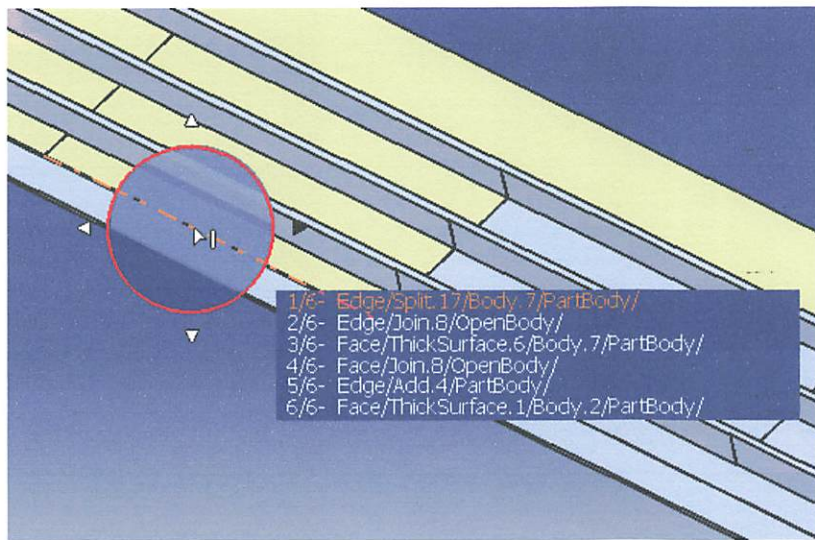
Yang dimaksud dengan *surface* adalah model matematika permukaan dalam ruang 3D. Pemodelan *surface* digunakan untuk menangani bentuk-bentuk geometri yang kompleks dan rumit seperti kontur permukaan.

Surface dari sayap N250-100PN dibangun dengan terlebih dahulu membuat *wireframe* berupa *spline* airfoil di root, kink dan tip. Kemudian tiap leading edge dan trailing edge dari airfoil tersebut saling dihubungkan.

Wireframe tersebut yang digunakan sebagai batasan untuk model surface.

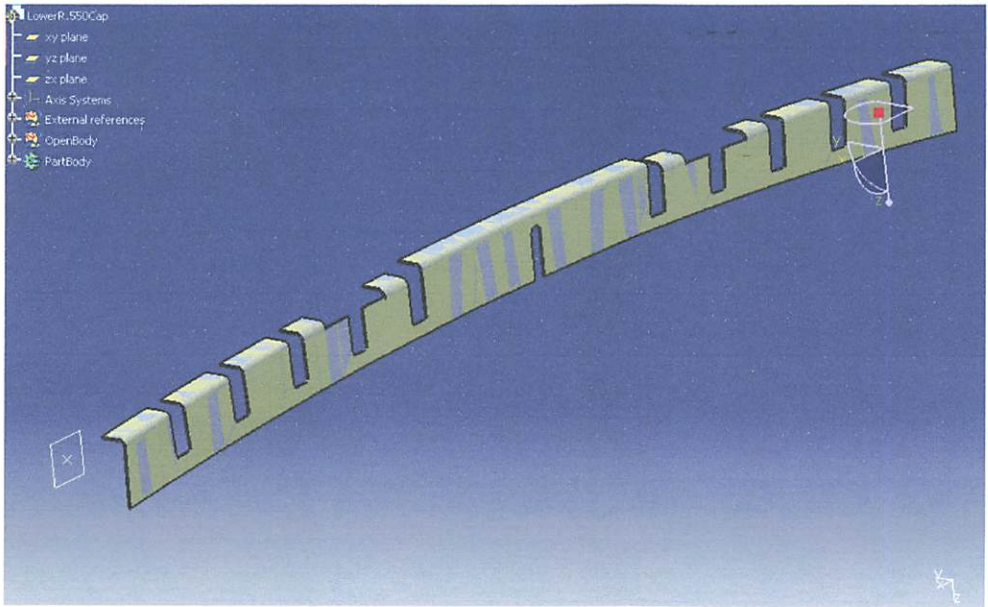
4.3 Solid modeling

Pemodelan ini menggunakan *entity 3D solid*. Model yang dibangun dapat direpresentasikan sebagai sebuah obyek tunggal melalui operasi *boolean* (dengan perintah *union* beberapa *entity 3D solid* dapat disatukan menjadi sebuah *entity 3D solid*), sehingga model yang dibangun dapat lebih terintegrasi dengan unjuk kerja yang lebih baik, karena beberapa karakteristik penting seperti volum, pusat masa, dan momen inersia dapat diketahui.



Gambar 4.3 informasi data yang jelas

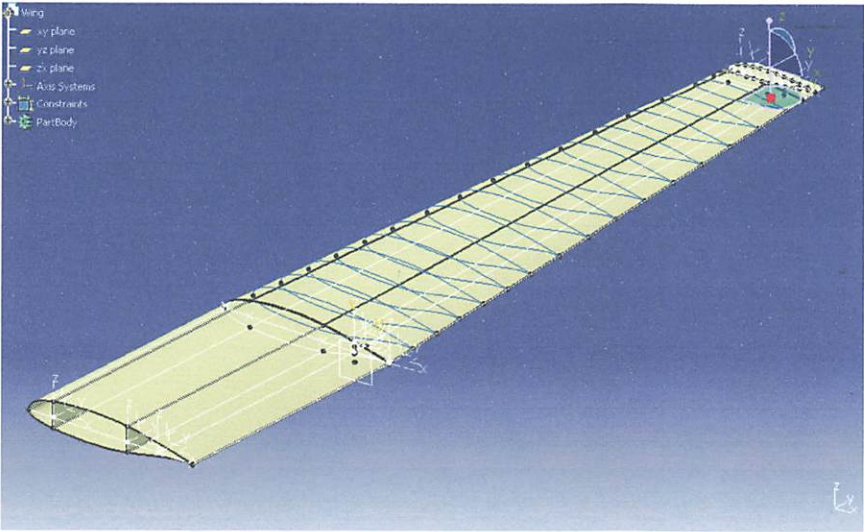
Setiap model yang terbentuk memiliki informasi yang sangat jelas. Terlihat pada gambar 4.3 , informasi dari sebuah model solid dapat berupa *vertex*, *edge*, maupun *face*. Informasi tersebut dapat digunakan untuk membentuk model yang lain. Untuk *vertex* dari sebuah solid maupun surface dapat digunakan untuk membentuk model *point*. Informasi *edge* dari solid dan surface dapat digunakan untuk membentuk line, ataupun spline. Dan informasi *face* dari model solid dapat digunakan untuk membentuk model surface.



Gambar 4.4 Thick Surface dari rip cap

Gambar 4.4 diatas melihatkan salah satu operasi pembentukan model solid yang berasal dari model surface, dengan cara pemberian ketebalan pada model surface. Model surface yang tertutup dapat pula menjadi model solid.

Ada banyak sekali operasi yang dapat dilakukan di CATIA. Garis biru yang berbentuk deretan airfoil pada gambar 4.5 merupakan hasil dari operasi *intersection* antara surface sayap dengan *plane* yang berderet sepanjang outerwing.

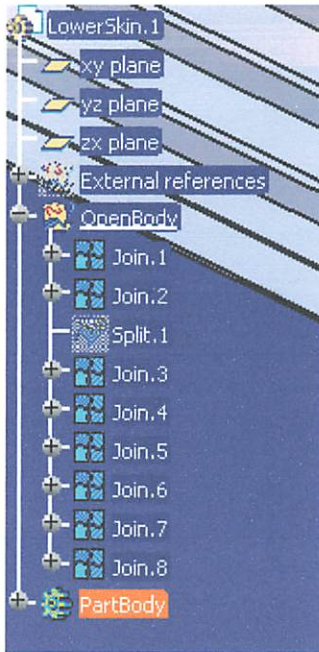


Gambar 4.5 Intersection

4.4 Hierarki parent-children

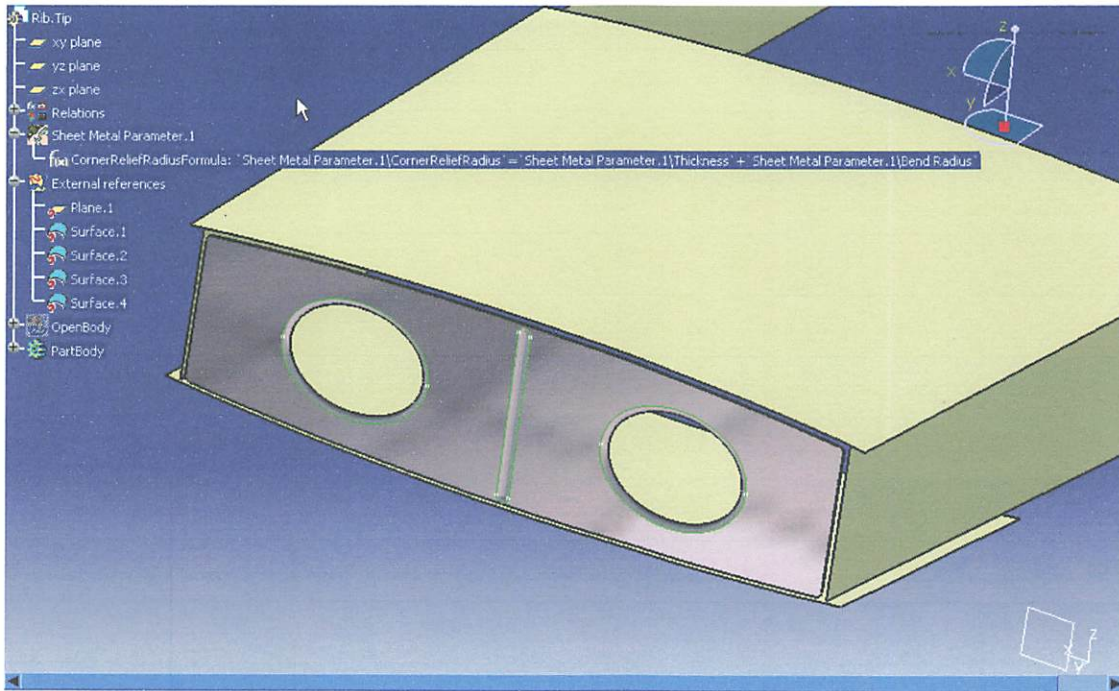
Setiap part yang terbentuk akan memiliki hierarki bagaimana part tersebut dibuat dan model apa saja yang menyusun part tersebut. Hierarki dari seluruh aplikasi yang dilakukan di CATIA seperti part design, assembly design, drafting analysis, Digital MockUp (DMU), human analysis dan lainnya berada pada *specifications tree*. *Specifications tree* adalah catatan dari seluruh operasi yang dilakukan. Proses pengeditan dari suatu operasi yang telah terjadi dapat dilakukan dengan memilihnya di *specifications tree* ini. Proses pengeditan yang terjadi mengakibatkan semua proses yang terbentuk setelahnya dan terkait dengan proses tersebut akan ikut berubah secara otomatis. Hubungan ini yang disebut sebagai hubungan *parent and children*. Di *specifications tree* ini dapat terlihat dengan jelas hubungan *parent and children* antar operasi yang telah dilakukan. Hubungan *parent and children* tidak hanya terjadi antar proses dalam satu part, tetapi dapat terjadi antar part yang menyusun sebuah produk. Gambar 4.6 memperlihatkan suatu *specifications tree* dari workbench part design. Bagian *External references* yang ada pada *specifications tree* merupakan penghubung part LowerSkin.1 pada gambar 4.6 dengan part yang lain, dengan kata lain, LowerSkin.1 sebagai children. External references maupun OpenBody terdiri atas sekumpulan model *wireframe* dan *surface*, model solid masuk didalam PartBody. Model *wireframe* dan *surface* dapat juga dimasukkan kedalam PartBody, tetapi mengelompokkan model kedalam External references ataupun OpenBody akan menjadikan lebih mudah dalam pencarian data operasi dan terlebih pada pengeditan. Sebagai contoh suatu operasi telah dilakukan pada PartBody, kemudian muncul data baru berupa model surface di External references. Operasi yang telah dilakukan terlebih dahulu tersebut ingin dilakukan perubahan pada *boundary condition* –nya dengan mengganti ke data yang baru masuk, maka proses ini akan berjalan.

Jika tidak ada pengelompokan, Join.3 pada OpenBody dapat diedit dengan menjadikan Join.1 dan Join.2 masuk ke dalamnya (Join.3 sebagai children), tetapi proses tidak akan berlangsung jika memasukkan Join.4. Hal ini disebabkan di Joint.4 belum terdefinisi ketika Join.3 dibentuk.



Gambar 4.6 Specification tree

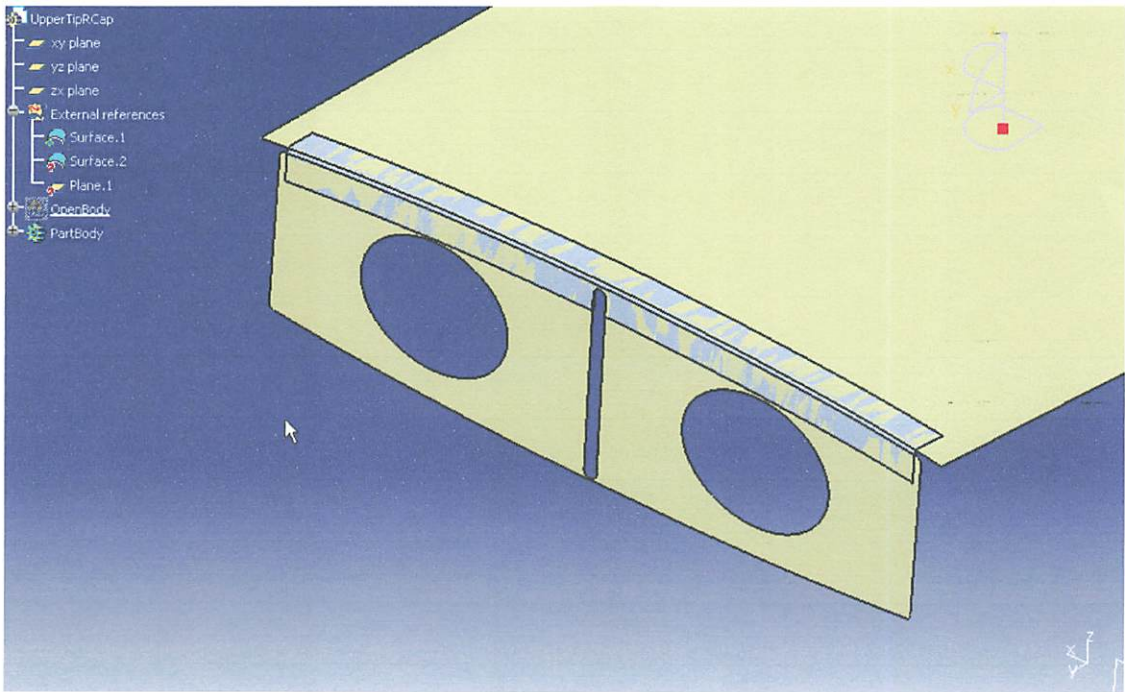
Antar part penyusun model sayap pada tugas akhir ini dibangun dengan memiliki hubungan *parent and children*, sehingga jika terjadi perubahan pada *parent* maka secara otomatis part yang sebagai *children*-nya akan ikut berubah jika telah mengalami proses *update*. Proses *link* dan *update* ini yang mengakibatkan tidak perlu lagi dilakukan proses menggambar ulang dari part tersebut. Perubahan pada *children* dapat melalui proses pengeditan ataupun tanpa perlu diedit. Perlu tidaknya dilakukan pengeditan dapat diketahui melalui *report error* / *peringatan* muncul pada layer. Untuk pendefinisian *parent* dan *children*-ya harus dilakukan dengan benar benar, jangan sampai muncul proses *looping* antar pasangan part akibat salah satunya menjadi *parent* maupun *children* dari part pasangannya.



Gambar 4.7 External references 1

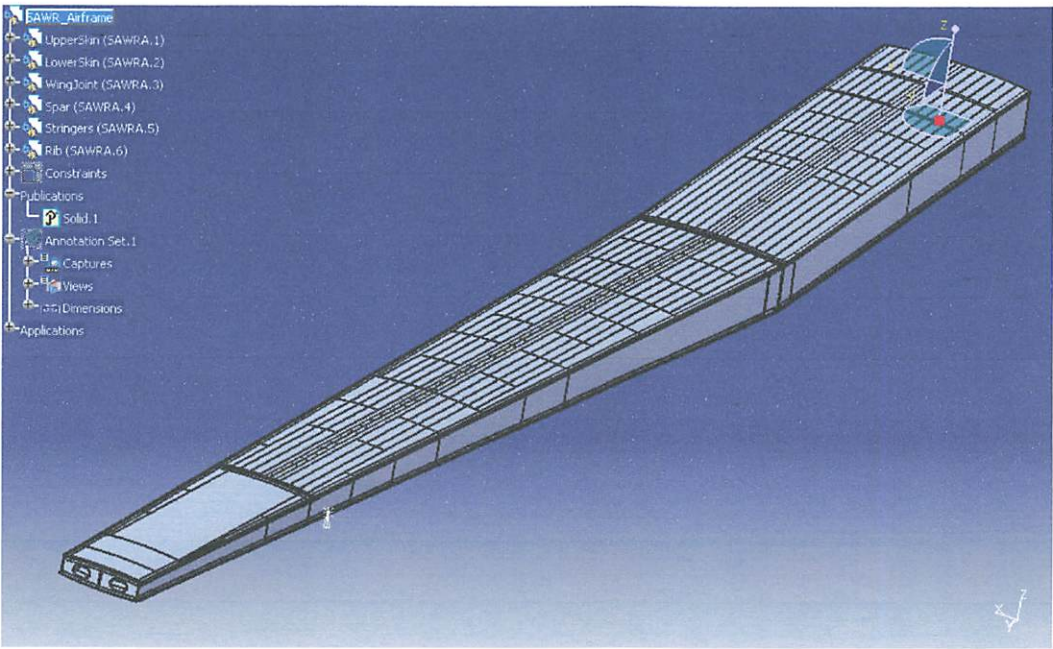
Dari gambar 4. 7 di atas terlihat bahwa part Rib.Tip memiliki referensi berupa surface milik upper skin, lower skin rear spar dan front spar. Ketika terjadi perubahan pada salah satu model referensinya, maka secara otomatis part Rib.Tip tersebut juga mengalami perubahan.

Dari gambar 4. 8 di bawah merupakan gambar dari UpperTipRCap dengan external references surface dari Rib.Tip dan UpperSkin. Jika terjadi perubahan pada UpperSkin maka Rib.Tip dan UpperTipRCap akan berubah secara otomatis pula.



Gambar 4.8 external references 2

4.5 Assembly dari model solid wingbox

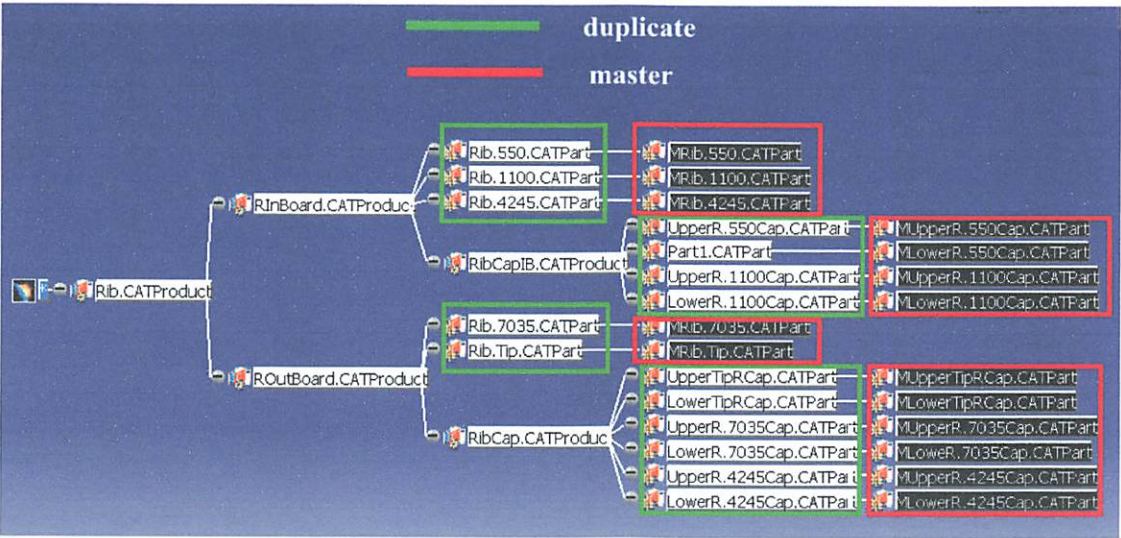


Gambar 4.9 WingBox

Assembly merupakan workbench yang digunakan dalam menyusun part-part menjadi sebuah produk. Dokumen dari assembly berbeda dari part design dan memiliki tipe file .CATProduct. Pada penyusunan produk dari sayap N250-100PN ini dikelompokkan menjadi beberapa *sub-assembly* dan didalam *sub-assembly* terdapat *sub-assembly* yang lebih kecil lagi hingga menjadi *sub-assembly* yang hanya memiliki part. Pengelompokan produk dan part pada tugas akhir ini berdasarkan kesamaan posisi. Gambar 3.9 menunjukkan *sub-assembly* dari salah satu *sub-assembly* Wing. Metoda ini dilakukan guna memudahkan dalam manajemen part.

Pada gambar 4.10, salah satu sub-assembly Rib memiliki dua buah sub-assembly yaitu, RInBoard (rip pada lokasi inboard dari wing) dan ROutBoard (rip pada lokasi outboard dari wing). RInBoard maupun ROutBoard merupakan sub-assembly akhir yang memiliki anggota berupa part.

Ketika membuka suatu file assembly maka seluruh sub_assembly didalamnya hingga part yang dimiliki akan muncul.

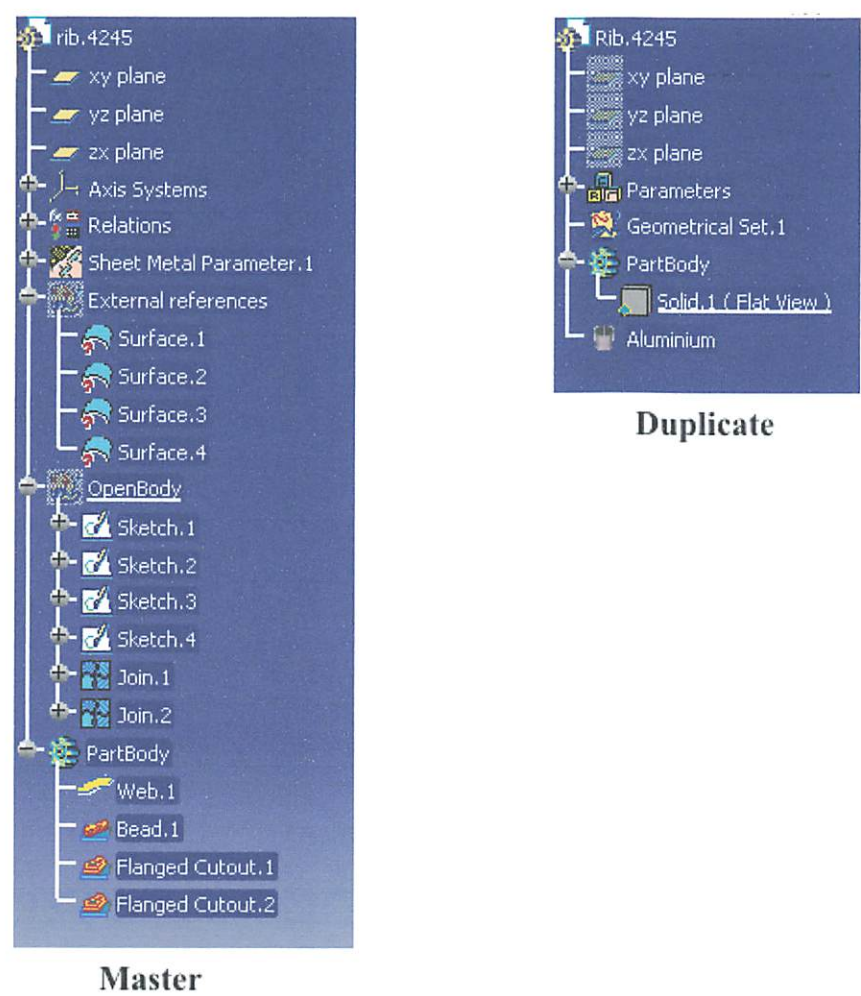


Gambar 4.10 Desk (link antara produk ke part dan master)

Gambar 4.10 juga menerangkan metoda lain yang digunakan dalam membentuk produk dari sayap. Metoda ini bertujuan untuk meringankan kinerja computer ketika membuka dokumen produk, semakin banyak part yang terkandung didalamnya maka semakin lama dokumen produk akan terbuka sehingga semakin berat kerja computer. Untuk mengatasi masalah ini,

part yang dibuka oleh *assembly*-nya merupakan tiruan (*duplicate*) dari part yang dapat dilakukan proses pengeditan jika terjadi perubahan, yang disebut sebagai *master*.

Data dari *master* lebih lengkap dibandingkan data tiruannya sehingga ukuran filenya lebih besar pula. Ukuran file *duplicate* dapat berkurang hingga 20% dari *master*-nya. Hal ini yang mengakibatkan produk semakin berat dan lama untuk dibuka.

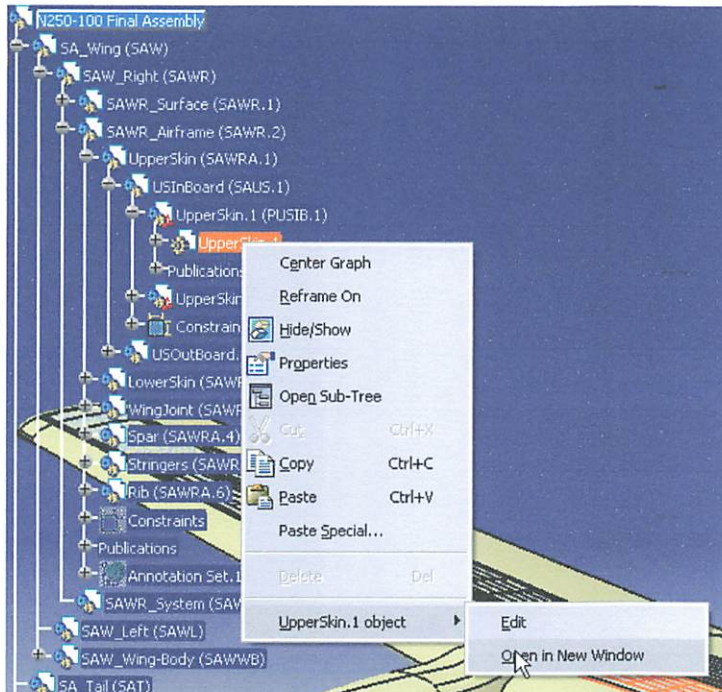


Gambar 4.11 Master- duplicate

Perbedaan *master* dan tiruannya dapat dilihat dari *specifications tree* keduanya. Master berisikan seluruh operasi yang dilakukan untuk membuat part tersebut sedangkan *duplicate*

hanya berisi hasil akhir dari part, yang dapat berupa solid maupun surface. Perbedaannya dapat dilihat dari gambar 4.11.

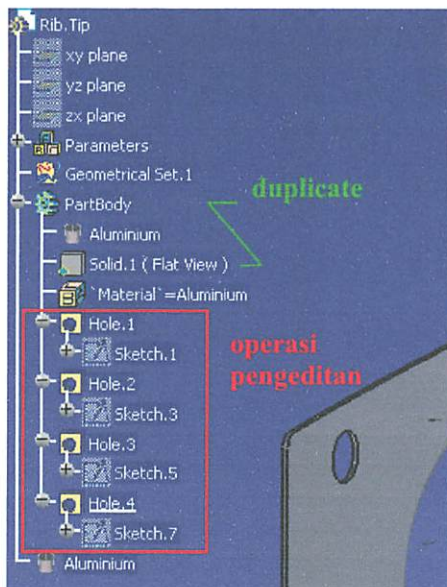
Metoda *master-duplicate* tidak berpengaruh pada kondisi *parent and children*. Dari gambar 4.10 juga menunjukkan hubungan antara assembly dengan part serta part *master* dengan *duplicate*-nya. Ketika part *master* mengalami perubahan, maka *duplicate* -nya juga ikut berubah. Dapat diartikan bahwa *duplicate* merupakan *children* dari part *master*. Keuntungan lain dari cara ini adalah melindungi data part dari perubahan yang tidak diinginkan. Untuk desain pesawat terbang yang memiliki tingkat ketelitian yang sangat tinggi, sangat berisiko besar jika terjadi kesalahan dalam mendesain, walaupun hanya sebuah part yang salah. Setiap perancang sebuah part menjadi sangat bertanggung jawab terhadap desain yang ia hasilkan. Dokumen dari suatu produk dapat dilihat oleh banyak sekali orang, baik yang berkaitan langsung dengan proses desain maupun yang secara tidak langsung berhubungan. Untuk membuka dokumen part dari dokumen produk sangatlah mudah, hanya dengan memberikan *double click* ataupun klik kanan untuk part yang dipilih kemudian pilih *edit* ataupun *Open in New Window* modifikasinya akan dapat dilakukan (gambar 4.12).



Gambar 4.12 Open part in new window

Jika produk tersusun atas part *master*, seseorang selain perancangnya dengan sengaja atau tidak akan sangat mudah untuk merubah ataupun menambahkan suatu operasi pada part tersebut. Sangat sulit ditemukan hasil modifikasi yang tidak diinginkan jika hanya merubah nilai dari suatu operasi, sebagai contoh merubah diameter dari lubang *rivet* atau ketebalan part sebesar 0.1 mm, dan bagaimana juga jika jumlah operasi yang dilakukan untuk membentuk part tersebut sangat banyak.

Jika part yang terbuka adalah *duplicate*, maka sangatlah mudah mencari modifikasi yang terjadi di *specification tree* dan untuk memodifikasinya hanya bisa dengan menambahkan operasi saja. Gambar 4.13 menunjukkan kondisi yang terjadi jika dilakukan modifikasi pada *part duplicate*.



Gambar 4.13 Modifikasi pada part duplicate

Hubungan *parent and children* antara *master* dan *duplicate* dapat diputus ataupun disambung kembali oleh perancangnya.

BAB 5

PEMAKAIAN WORKBENCH ANALYSIS & SIMULATION UNTUK WINGBOX.

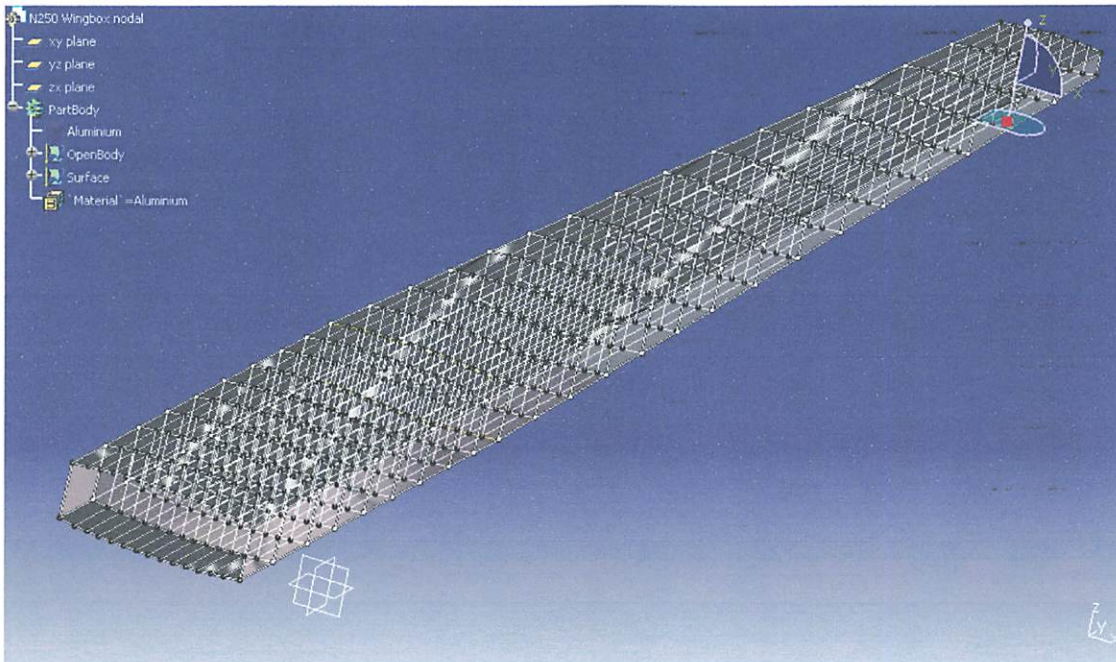
5.1 Pendahuluan

Telah dijelaskan mengenai workbench *analysis and simulation* pada BAB 2 yang lalu maka pada BAB 5 ini akan lebih dijelaskan untuk studi kasus Airframe (*wingbox*) dari sayap N250-100PN hasil rekonstruksi. Model FEM didapat dari contoh yang diberikan PT DI kemudian dipetakan ke model wireframe dan surface. Studi kasus ini bukan melakukan analisa kekuatan struktur, melainkan hanya contoh penggunaan. Ada dua studi untuk analisis static pada tugas akhir ini, yaitu analisis dengan *Finite Elemen Model* yang tersusun atas elemen *beam* dan *Shell*, dan analisis dengan elemen model solid.

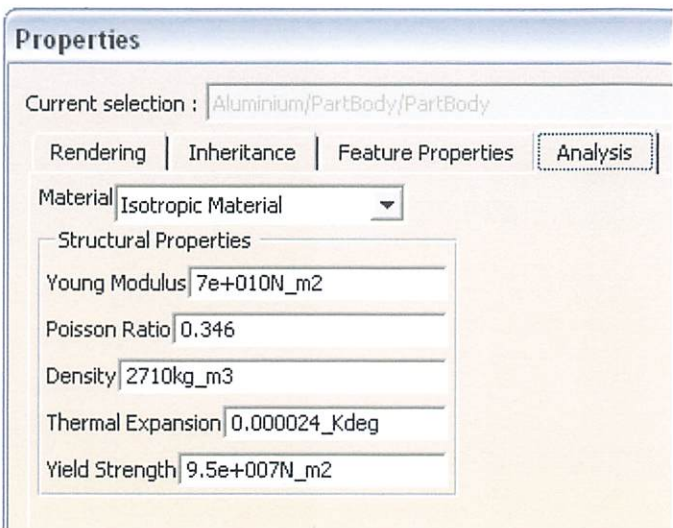
5.2 Pembentukan model

Pembentukan model dari Finite Elemen Model yang tersusun atas elemen *Beam* dan *Shell* hanya berupa *wireframe* dan *surface* saja. Model dibuat dalam workbench *Generatif Shape design*, kemudian model yang telah jadi diberi material aluminium.

Gambar 5.1 merupakan gambar dari *finite elemen model*. Karena model hanya berupa wireframe dan surface, maka hanya jenis materialnya saja yang dapat diberikan. Untuk jenis elemen yang dipakai dan ketebalan dari shell serta luar area dari penampang beam diberikan ketika membuat mesh pada workbench *advanced meshing tools*.



Gambar 5.1 Finite element model (bar dan shell)

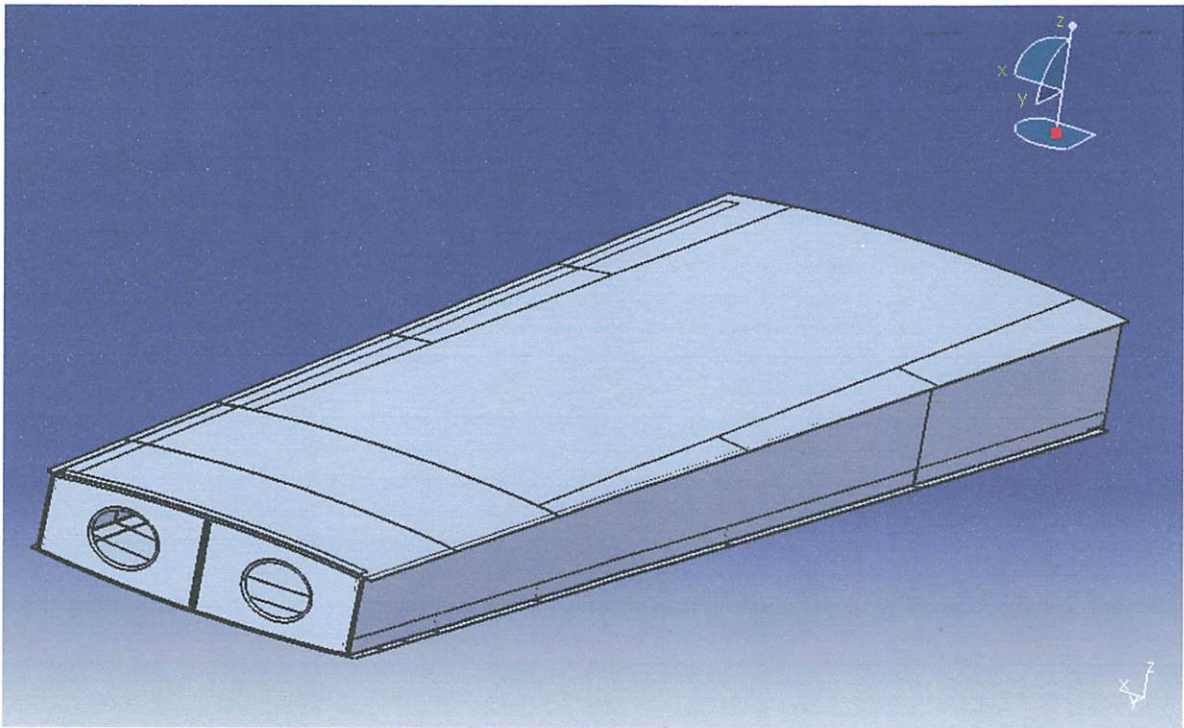


Gambar 5.2 Material properties

Sifat material aluminium yang dipakai sesuai pada gambar 5.2

Untuk model elemen solid yang digunakan adalah bagian sheet metal dari model assembly wingbox yang sudah ada.

Gambar 5.3 Merupakan gambar model solid yang digunakan.



Gambar 5.3 Solid model

5.3 Elemen

Jenis elemen yang digunakan dalam workbench analisis di CATIA adalah : (ref.2)

- *Linear Triangle Sheel*
- *Parabolic Triangle Sheel*
- *Linear Quadrangle Sheel*
- *Lineat Tetrahedron*
- *Parabolic Tetrahedron*
- *Beam*
- *Spring*
- *Contact Rod*, dan masih banyak lagi.

Jenis element yang dipakai dalam analisis pada tugas akhir ini adalah *beam*, *linear quadrangle sheel*, dan *linear tetrahedron*, dengan sifat-sifat dari elemennya sebadai berikut :

Linear Quadrangle Shell

Linear Quadrangle Shell adalah sebuah elemen sheel dengan empat buah nodal yang berdasarkan Reissner/Mindlin.

Type	surface element
Physical property	shell
Mesh connectivity	parabolic quadrangle
Number of nodes	4
Degrees of freedom (per node)	6 (3 translations and 3 rotations)
Type of behavior	elastic



Elemen ini memiliki empat *gauss points*:

P1 ($-\sqrt{2}/2$; $-\sqrt{2}/2$)	P2 ($\sqrt{2}/2$; $-\sqrt{2}/2$)
P3 ($\sqrt{2}/2$; $\sqrt{2}/2$)	P4 ($-\sqrt{2}/2$; $\sqrt{2}/2$)

Beam

Beam adalah elemen beam nodal lurus dengan transverse shear yang berdasarkan pada teori Timoshenko.

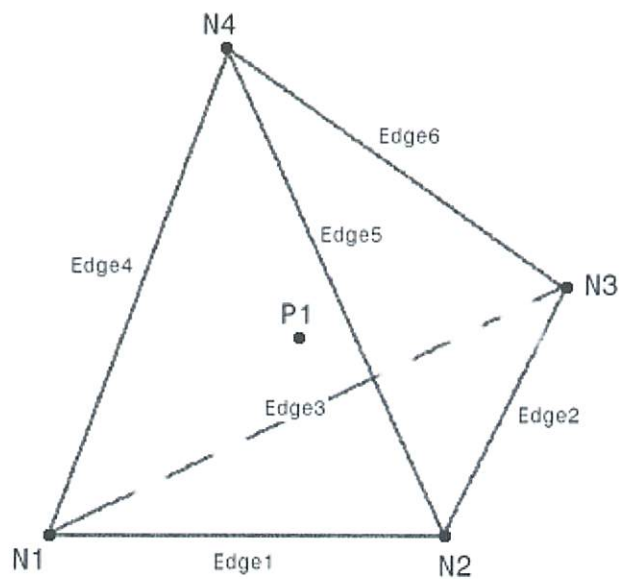
Type	lineic element
Physical property	beam
Mesh connectivity	rod
Number of nodes	2
Degrees of freedom (per node)	6 (3 translations and 3 rotations)
Type of behavior	elastic



Linear Tetrahedron

Linear Tetrahedron adalah empat nodal *isoparametric* elemen solid.

Type	solid element
Physical property	solid
Mesh connectivity	linear tetrahedron
Number of nodes	4
Degrees of freedom (per node)	3 (translations)
Type of behavior	elastic



Elemen ini hanya memiliki sebuah *gauss point*: pusat grafitasi (P1) dari tetrahedron.

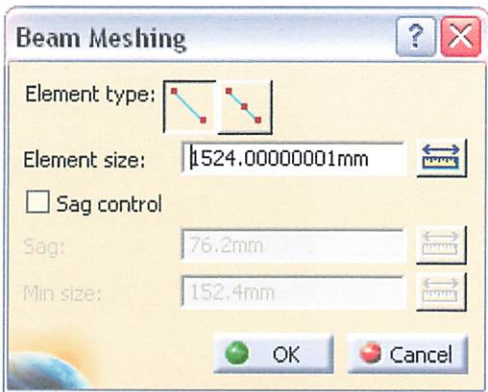
Hanya memiliki tiga buah translasi.

5.3 Pembentukan mesh

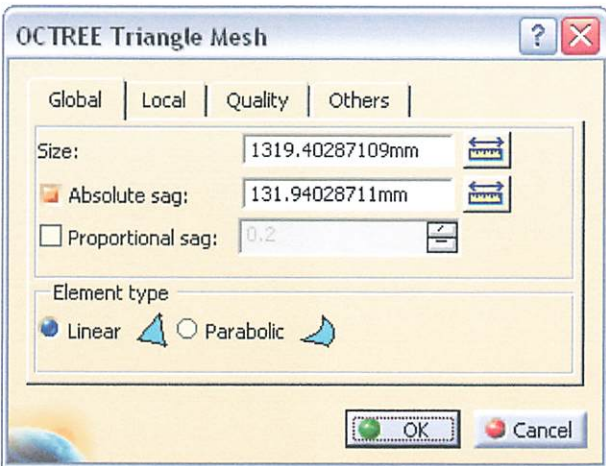
Untuk finite element model , model elemen yang dipilih adalah Cylindrical beam dan linear quadrangle shell.

Terlebih dahulu yang dilakukan meshing adalah beam yang dilanjutkan dengan shell, jika perlakuannya terbalik maka tidak akan terjadi ikatan antara nodal milik beam dengan shell, sehingga ketika dilakukan analisis maka akan terjadi kesalahan.

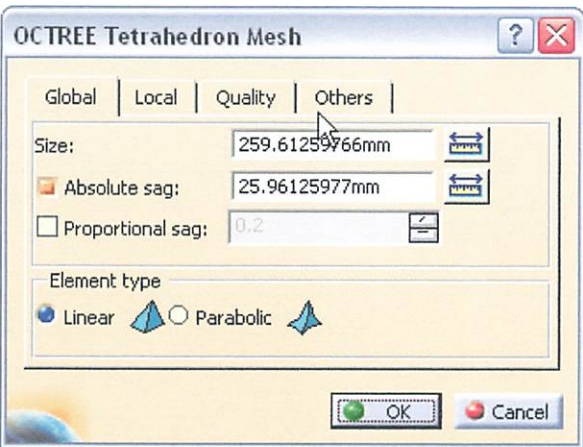
Pembentukan mesh dapat pula dilakukan secara otomatis pada workbench *generatif structural analysis* tetapi yang dihasilkan adalah mesh standar.



Gambar 5.4 window dialog beam meshing

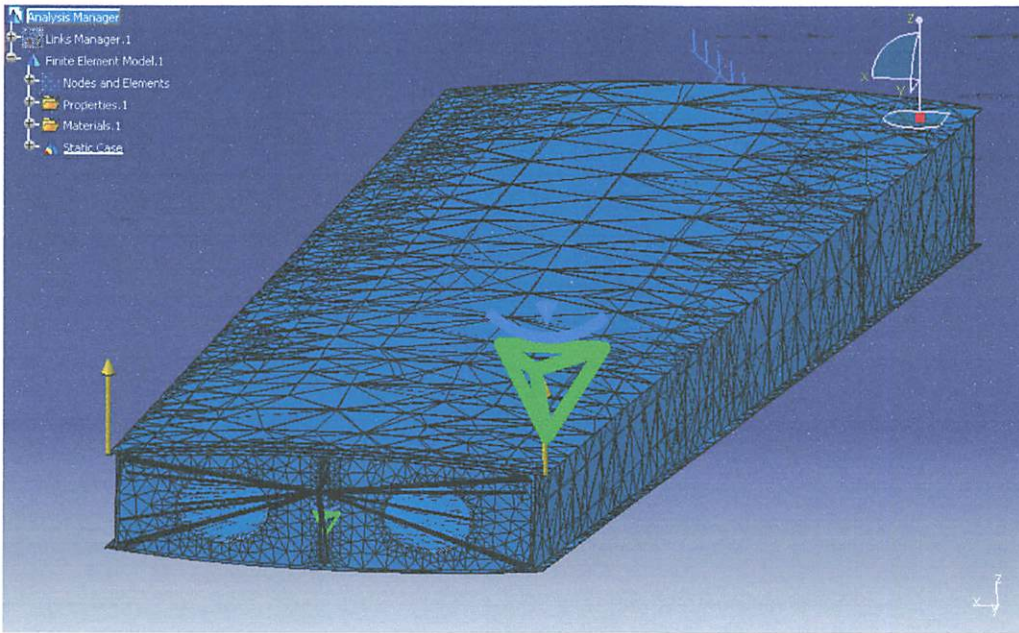


Gambar 5.5 window dialog octree triangle mesh



Gambar 5.6 window dialog octree tetrahedron mesh

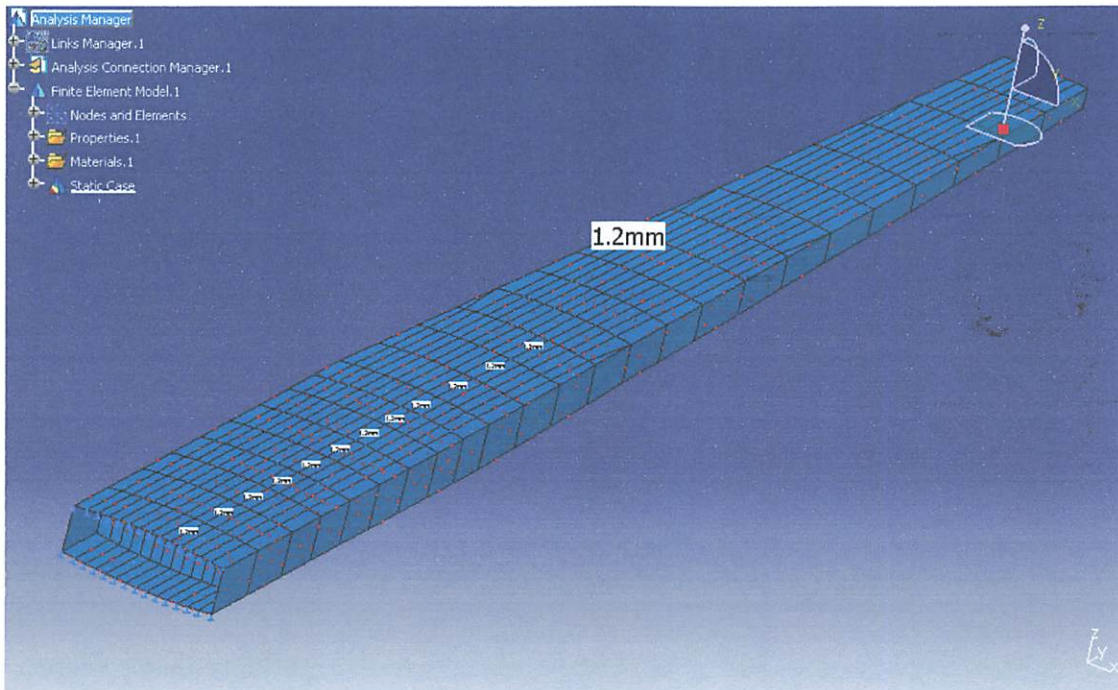
Gambar 5.4 hingga 5.6 adalah window dialog pendefinisian pertama kali untuk model beam, shell dan solid. Toolbar dari widow dialog ini juga terdapat di workbench generative structural analysis, tetapi tidak dapat dilakukan proses pengeditan untuk menjadi yang diinginkan. Secara lengkap mengenai pemakaian window dialog ini ini dapat dilihat pada referensi [2].



Gambar 5.7 Solid mesh dengan element linear tetrahedron

Gambar diatas adalah hasil dari proses meshing instant pada workbench generatif structural analysis. Tampak terlihat jelas jika mesh yang dihasilkan sangat tidak teratur. Jumlah nodal yang diperoleh adalah 5631 dan jumlah elemennya 16003. Lama waktu yang digunakan untuk membangun mesh adalah 3 menit.

Gambar 5.8 adalah hasil dari proses meshing pada workbench advanced meshing tools dengan proses pengeditan yang tersedia.



Gambar 5.8 mesh dari finite element model (bar dan shell).

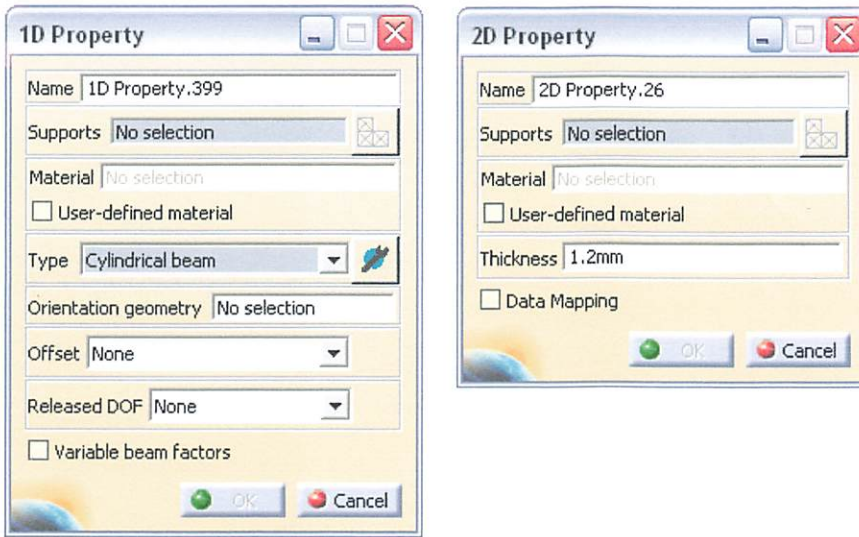
Jumlah nodal yang diperoleh adalah 1960 dan jumlah elemennya 3898.

5.4 Penentuan properties, contact, restraint, dan load.

Pendefinisian *property* dari model harus diberikan untuk dapat dilakukan analisis. Untuk finite elemen bar dan shell, window dialog untuk mendefinisikannya dapat dilihat dari gambar 5.9.

Pada contoh kasus pada tugas akhir ini, 1 D properti dipakai adalah tipe *Cylindrical Beam* dengan luas penampang 5mm dan harga thickness untuk 2D property-nya adalah 1.2 mm.

3D property untuk solid model telah terdefinisi sendiri ketika masuk workbench generative structural analysis.



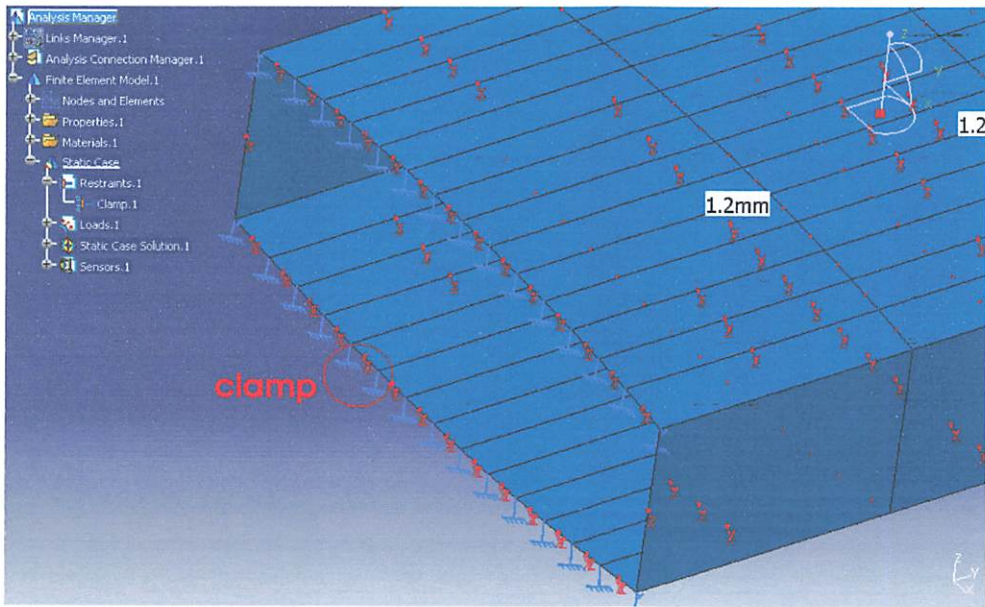
Gambar 5.9 Window dialog 1D dan 2D properties

Beam tersusun atas wirefram berupa line yang terputus putus, maka untuk dapat di analisis terlebih dahulu ditentukan tipe conection antar beam. Tipe yang dipilih untuk analisis kasus ini adalah *nodes to nodes conection*. Untuk mendefinisikannya dibutuhkan waktu yang lama sebab harus dipilih satu persatu pasangan titik yang ingin dihubungkan.

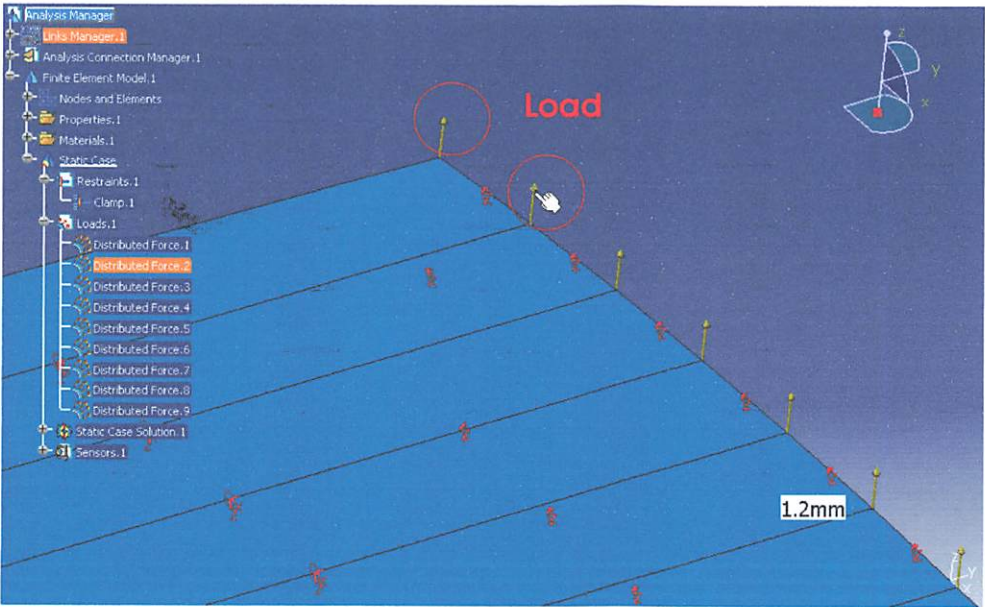
Restraint adalah istilah untuk constraint diworkbench generative structural analysis. Restraint yang diberikan untuk kasus finite element model dari wing box adalah *Clamp*. Clamp diletakkan pada beam yang berada di root.

Beban diletakkan pada beam yang berada di tip. Tipe load yang diberikan adalah *distributed force* yang diletakkan pada tiap nodal dari beam, besar distributed force berkurang dari 10 N di front spar hingga 2 N di rear spar.

Symbol clamp dan load dapat dilihat pada gambar 5.10 dan 5.11.



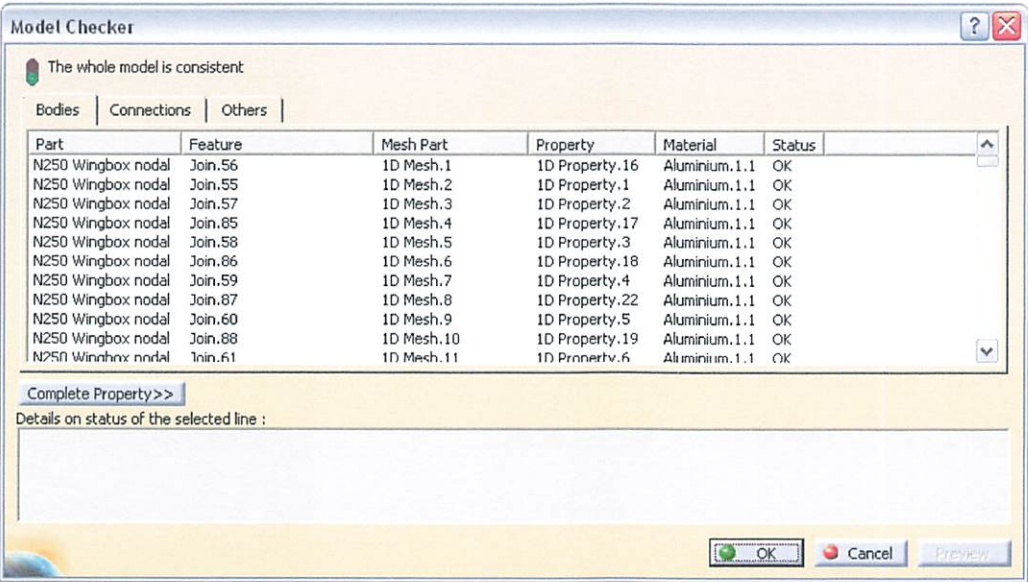
Gambar 5.10 Restrain berupa Clamp.



Gambar 5.11 Load yang berupa Distributed force

Setelah semua didefinisikan, maka sebelum *running* dapat dicek terlebih dahulu status dari model, apakah terdapat kesalahan atau tidak.

Gambar 5.12 merupakan window dialog untuk melihat status dari model. Jika lampu yang berada pada pojok kiri atas dari window dialog ini, maka seleruh status dalam keadaan OK. Dan model siap dilakukan analisis.



Gambar 5.12 Window dialog dari Model Checker

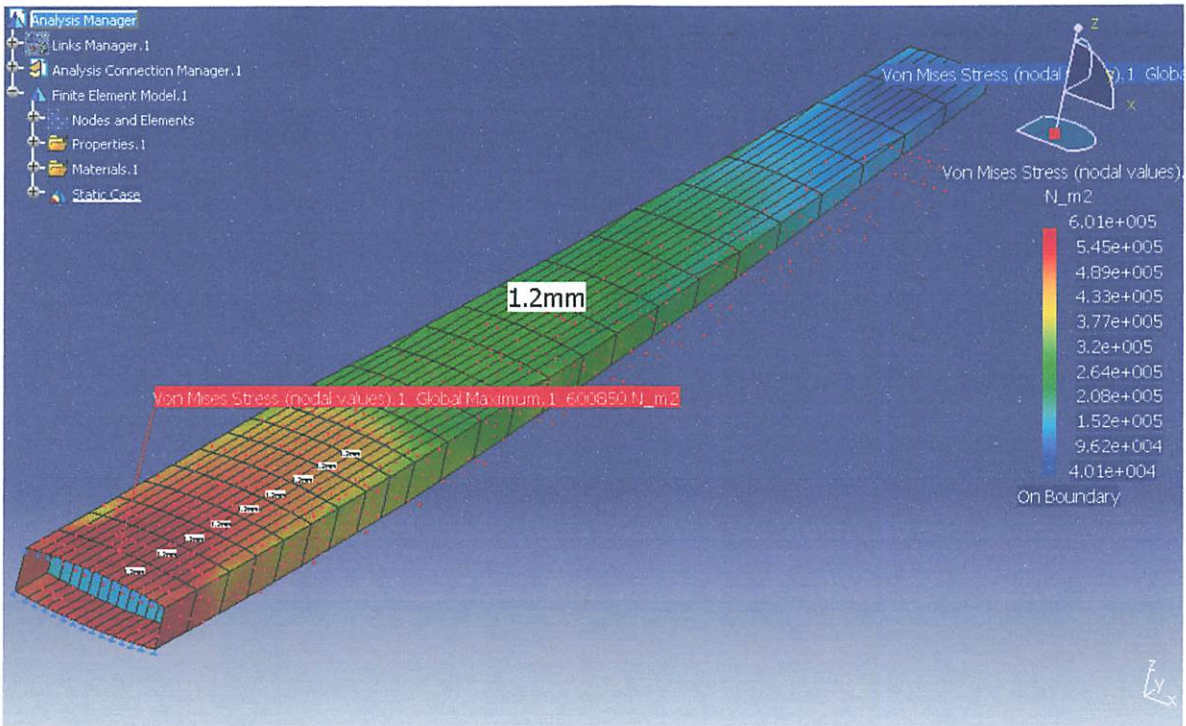
Analisis model kasus beam dan shell membutuhkan waktu selama 5 menit. Hasil dapat dilihat pada gambar 5.13. solusi dapat berupa *Deformation* , *Von Mises Stress*, maupun *Displacement*.

Maksimum Von Misses Stress adalah $6,01 \cdot 10^5 \text{ Nm}^2$,lebih kecil dibandingkan Yield Strength bahan sebesar $9.5 \cdot 10^7 \text{ Nm}^2$.

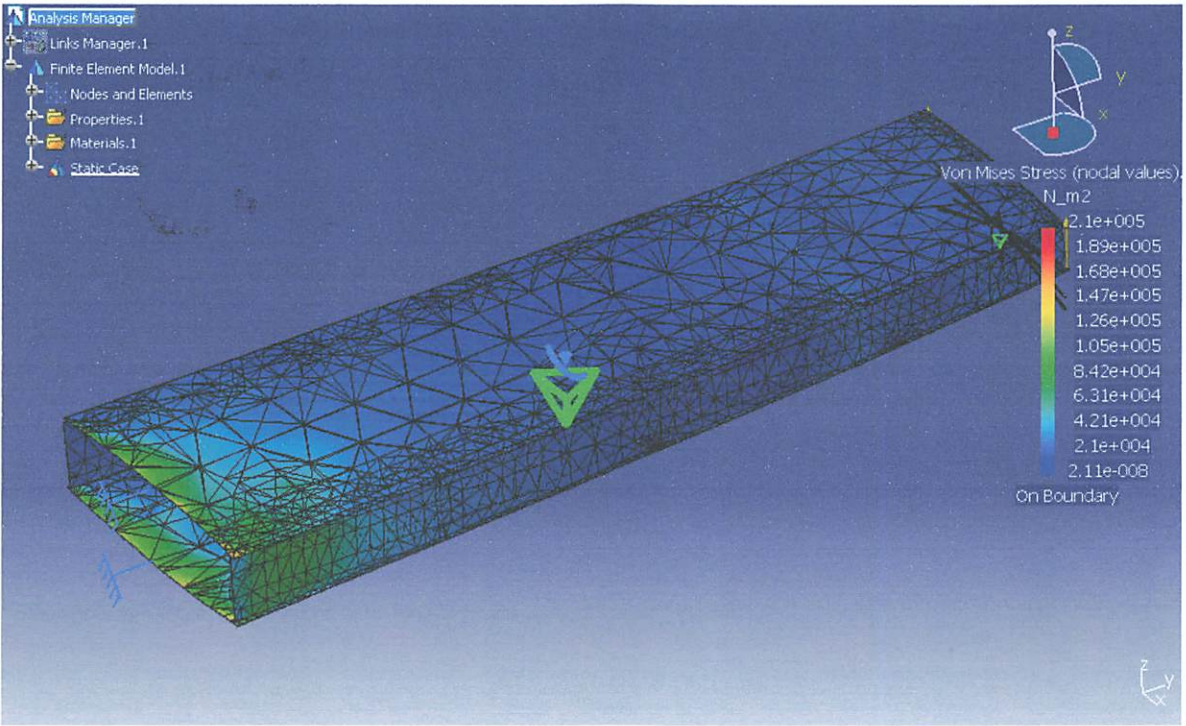
Pada analisis menggunakan model solid dilakukan hal yang sama yakni menggunakan restrain berupa clamp, menggunakan material yang sama, tetapi load yang dimasukkan seragam dari front spar ke rear spar sebesar 10 N.

Hasil analisis dengan menggunakan model solid dapat dilihat pada gambar 5.14.

Maksimum Von Misses Stress adalah $2.1 \cdot 10^5 \text{ Nm}^2$,lebih kecil dibandingkan Yield Strength bahan sebesar $9.5 \cdot 10^7 \text{ Nm}^2$.



Gambar 5.13 Hasil analisis Von Mises Stres untuk model beam dan shell.



Gambar 5.14 Hasil analisis Von Mises Stres untul model solid

BAB 6

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Dari hasil pemikiran dan pengerjaan semua tugas akhir ini, dapat ditarik beberapa kesimpulan sebagai berikut :

1. CATIA V5 merupakan perangkat lunak yang sangat bagus, dengan CATIA, model 3D tidak hanya sekedar sebagai model yang menarik untuk dilihat saja, tetapi dengan satu buah model 3D dapat dilakukan berbagaimacam proses, dari analisis hingga manufaktur.
2. Skech merupakan gambar 2D awal untuk membentuk model 3D, sedangkan drafting merupakan gambar 2D yang siap dicetak ke kertas dan berasal dari model 3D.
3. Ketika membuat suatu part, buatlah kelopmok yang tersusun atas model wireframe saja, surface saja atau solid saja. Untuk menjaga-jaga adanya modifikasi
4. Keberadaan update, link dan parent-children menyebabkan gambar menjadi sangat mudah untuk dimonifikasi.
5. Sebuah model 3D di CATIA dapat digunakan sebagai *boundary* untuk lebih dari satu analisis.
6. Dengan VBA, maka window dialog untuk aplikasi yang dibuat dapat didesain dengan mudah.

6.2 Saran

Untuk pengembangan lebih lanjut terhadap pemodelan tiga dimensi, penulis memberikan beberapa rekomendasi berikut :

1. Pemrograman selanjutnya dapat dilakukan pada bagian bagian lain dari pesawat udara, sebagai contoh bagian nose pesawat terbang, intersection antara wing dengan fuselage.
2. Jika ingin tetap berkonsentrasi pada bagian sayap pesawat terbang maka dibuat aplikasi yang lebih fleksibel dan untuk model yang lebih komplek
3. Mencoba untuk menggunakan bahasa pemrograman yang lebih handal dibandingkan VBA, sebagai contoh menggunakan CAA .
4. Jika ingin tetap menggunakan aplikasi yang sudah tersedia di CATIA, coba memaadukan VBA dengan *Knowlegeware* agar didapatkan aplikasi yang lebih *powerful* dan efisien
5. Kembangkan aplikasi yang tidak hanya berputar putar di sekitar model saja tetapi coba untuk menggabungkannya dengan poses analisis maupun proses-proses lain yang tersedia di CATIA.
6. Mempelajari CATIA lebih dalam lagi.

REFERENSI

- [1] Ibrahim Zeid, G., Mastering CAD/CAM, McGRAW.HILL INTERNATIONAL EDITION 2005
- [2] CATIA V5R12 DOCUMENTASION
- [3] Michael Halvorson, Microsoft Visual Basic 6.0 Professional, tep by Step, Elex Media Komputindo 2003
- [4] Dassault Systèmes Homepage, www.3ds.com
- [5] Google search, www.google.com

LAMPIRAN A

CONTOH PROGRAM PEMODELAN

```
Private Sub PreviewPlanform_cb_Click()
'Dim PartName As String
'PartName = part1.Name
'MsgBox "part :" & PartName
If AxisSystemType = "Global Coordinate" Then
    Dim XYZOrigin(0 To 2) As Double
    XYZOrigin(0) = 0
    XYZOrigin(1) = 0
    XYZOrigin(2) = 0
    Origin = XYZOrigin
Else
    'Find LocalOrigin
    Dim LocalOrigin As HybridShapePointCoord

    Set LocalOrigin = part1.FindObjectByName("LocalOrigin")
    'Dim XYZOrigin(0 To 2) As Double
    If TypeName(LocalOrigin) = "Nothing" Then
        AxisSystemType = "Local Coordinate"
        XYZOrigin(0) = CDBl(X_tb.Text)
        XYZOrigin(1) = CDBl(Y_tb.Text)
        XYZOrigin(2) = CDBl(Z_tb.Text)
        Origin = XYZOrigin
        'create Origin

        Set LocalOrigin = hybridShapeFactory1.AddNewPointCoord(Origin(0), Origin(1),
Origin(2))
        LocalOrigin.Name = "LocalOrigin"
        OpenBody.AppendHybridShape LocalOrigin
        part1.InWorkObject = LocalOrigin
        part1.Update
    Else
        XYZOrigin(0) = CDBl(X_tb.Text)
        XYZOrigin(1) = CDBl(Y_tb.Text)
        XYZOrigin(2) = CDBl(Z_tb.Text)
        Origin = XYZOrigin
        LocalOrigin.X.Value = Origin(0)
```

```

        LocalOrigin.Y.Value = Origin(1)
        LocalOrigin.Z.Value = Origin(2)
    part1.Update
End If

End If
'create Reference for axisSystem1
'Dim LocalOriginReference As Reference
Set LocalOriginReference = part1.CreateReferenceFromObject(LocalOrigin)

Dim axisSystem1 As AxisSystem
Set axisSystem1 = part1.FindObjectByName("Wing Reference")
If TypeName(axisSystem1) = "Nothing" Then
'create axisSystem1
Set axisSystem1 = axisSystems1.Add()
axisSystem1.Name = "Wing Reference"

axisSystem1.OriginType = catAxisSystemOriginByPoint
axisSystem1.OriginPoint = LocalOriginReference

axisSystem1.XAxisType = catAxisSystemAxisByCoordinates

Dim arrayOfVariantOfDouble1(2)
arrayOfVariantOfDouble1(0) = 1#
arrayOfVariantOfDouble1(1) = 0#
arrayOfVariantOfDouble1(2) = 0#
Set axisSystem1 Variant = axisSystem1
axisSystem1 Variant.PutXAxis arrayOfVariantOfDouble1

axisSystem1.YAxisType = catAxisSystemAxisByCoordinates

Dim arrayOfVariantOfDouble2(2)
arrayOfVariantOfDouble2(0) = 0#
arrayOfVariantOfDouble2(1) = 1#
arrayOfVariantOfDouble2(2) = 0#
Set axisSystem1 Variant = axisSystem1
axisSystem1 Variant.PutYAxis arrayOfVariantOfDouble2

axisSystem1.ZAxisType = catAxisSystemAxisByCoordinates

Dim arrayOfVariantOfDouble3(2)
arrayOfVariantOfDouble3(0) = 0#
arrayOfVariantOfDouble3(1) = 0#
arrayOfVariantOfDouble3(2) = 1#
Set axisSystem1 Variant = axisSystem1

```

```

axisSystem1.Variant.PutZAxis arrayOfVariantOfDouble3

part1.UpdateObject axisSystem1
axisSystem1.IsCurrent = False
part1.Update

Else

    axisSystem1.OriginPoint = LocalOriginReference
    part1.Update
End If

'SemiSpan
SemiSpan = WingSpan / 2
PlanformArea = Round((WingSpan ^ 2 / AspectRatio), 6)
SemiPlanformArea = PlanformArea / 2
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

If PlanformType_cbx.Value = "Simple Planform" Then
    'Dim CtipProjection As Double
    Dim CrootProjection As Double
    CrootProjection = CDBl(SimplePlanform.SimpleCRLength_tb.Value)
    CtipProjection = ((2 * SemiPlanformArea) - (CrootProjection * SemiSpan)) / SemiSpan
    'definition TraillingEdge's point at Root
    Dim RTEPointPlanform(0 To 2) As Double
    RTEPointPlanform(0) = Origin(0) + CrootProjection
    RTEPointPlanform(1) = Origin(1)
    RTEPointPlanform(2) = Origin(2)
    RTEPointPlanformP = RTEPointPlanform
    'create TraillingEdge's point at Root
    Dim RootTEPoint As HybridShapePointCoord
    Set RootTEPoint = part1.FindObjectByName("RootTEPoint")
    If TypeName(RootTEPoint) = "Nothing" Then
        Set RootTEPoint = hybridShapeFactory1.AddNewPointCoord(RTEPointPlanform(0),
RTEPointPlanform(1), RTEPointPlanform(2))
        RootTEPoint.Name = "RootTEPoint"
        OpenBody.AppendHybridShape RootTEPoint
        part1.InWorkObject = RootTEPoint
        part1.Update
    Else
        RootTEPoint.X.Value = RTEPointPlanform(0)
        RootTEPoint.Y.Value = RTEPointPlanform(1)
        RootTEPoint.Z.Value = RTEPointPlanform(2)
        part1.Update
    End If
    'create RootChord's Line from LeadingEdge's point and TraillingEdge's point
    'Dim RChordLine As HybridShapeLinePtPt

```

```

Set RChordLine = part1.FindObjectByName("RChordLine")
If TypeName(RChordLine) = "Nothing" Then
    Set RChordLine = hybridShapeFactory1.AddNewLinePtPt(LocalOrigin, RootTEPoint)
    RChordLine.Name = "RChordLine"
    OpenBody.AppendHybridShape RChordLine
    part1.InWorkObject = RChordLine
    part1.Update
Else
    RChordLine.PtOrigine = LocalOrigin
    RChordLine.PtExtremity = RootTEPoint
    part1.Update
End If

```

```

'definition Angle
Dim SweepPlanform, SweepPlanform_rad As Double
    SweepPlanform = CDBl(SimplePlanform.SimpleSweepAngle_tb.Value)
    SweepPlanform_rad = SweepPlanform * rad
Dim DihedralAngle, DihedralAngle_rad As Double
    DihedralAngle = CDBl(SimplePlanform.SimpleDihedralAngle_tb.Value)
    DihedralAngle_rad = DihedralAngle * rad

```

```

Dim delXTipPlanform As Double 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx!perlu dijelaskan
    delXTipPlanform = (SemiSpan) * Tan(SweepPlanform_rad)

```

```

'definition LeadingEdge Planform's point at Tip
Dim TipLEPointPlanform(0 To 2) As Double
    TipLEPointPlanform(0) = Origin(0) + delXTipPlanform
    TipLEPointPlanform(1) = Origin(1) + (SemiSpan)
    TipLEPointPlanform(2) = Origin(2)
    TipLEPointPlanformP = TipLEPointPlanform

```

```

'definition TrailingEdge Planform's point at Tip
Dim TipTEPointPlanform(0 To 2) As Double
    TipTEPointPlanform(0) = TipLEPointPlanform(0) + CtipProjection
    TipTEPointPlanform(1) = TipLEPointPlanform(1)
    TipTEPointPlanform(2) = TipLEPointPlanform(2)
    TipTEPointPlanformP = TipTEPointPlanform

```

'DIHEDRAL

```

'definition Tip's point with dihedral
Dim Dihedral As Double
    Dihedral = SemiSpan * Tan(DihedralAngle_rad)

```

```

'definition LeadingEdge Planform's point with dihedral at Tip
Dim TipLEPointPlanformDihedral(0 To 2) As Double
    TipLEPointPlanformDihedral(0) = TipLEPointPlanform(0)

```

```

    TipLEPointPlanformDihedral(1) = TipLEPointPlanform(1)
    TipLEPointPlanformDihedral(2) = TipLEPointPlanform(2) + Dihedral
    TipLEPointPlanformDihedralP = TipLEPointPlanformDihedral
'create LeadingEdge Planform's point with dihedral at Tip
Dim TipLEPoint As HybridShapePointCoord
Set TipLEPoint = part1.FindObjectByName("TipLEPoint")
If TypeName(TipLEPoint) = "Nothing" Then
    Set TipLEPoint =
hybridShapeFactory1.AddNewPointCoord(TipLEPointPlanformDihedral(0),
TipLEPointPlanformDihedral(1), TipLEPointPlanformDihedral(2))
    TipLEPoint.Name = "TipLEPoint"
    OpenBody.AppendHybridShape TipLEPoint
    part1.InWorkObject = TipLEPoint
    part1.Update
Else
    TipLEPoint.X.Value = TipLEPointPlanformDihedral(0)
    TipLEPoint.Y.Value = TipLEPointPlanformDihedral(1)
    TipLEPoint.Z.Value = TipLEPointPlanformDihedral(2)
    part1.Update
End If
'Dim TipLEPointReference As Reference
Set TipLEPointReference = part1.CreateReferenceFromObject(TipLEPoint)

'definition TrailingEdge Planform's point with dihedral at Tip
Dim TipTEPointPlanformDihedral(0 To 2) As Double
    TipTEPointPlanformDihedral(0) = TipTEPointPlanform(0)
    TipTEPointPlanformDihedral(1) = TipTEPointPlanform(1)
    TipTEPointPlanformDihedral(2) = TipTEPointPlanform(2) + Dihedral
    TipTEPointPlanformDihedralP = TipTEPointPlanformDihedral
'create TrailingEdge Planform's point with dihedral at Tip
Dim TipTEPoint As HybridShapePointCoord
Set TipTEPoint = part1.FindObjectByName("TipTEPoint")
If TypeName(TipTEPoint) = "Nothing" Then
    Set TipTEPoint =
hybridShapeFactory1.AddNewPointCoord(TipTEPointPlanformDihedral(0),
TipTEPointPlanformDihedral(1), TipTEPointPlanformDihedral(2))
    TipTEPoint.Name = "TipTEPoint"
    OpenBody.AppendHybridShape TipTEPoint
    part1.InWorkObject = TipTEPoint
    part1.Update
Else
    TipTEPoint.X.Value = TipTEPointPlanformDihedral(0)
    TipTEPoint.Y.Value = TipTEPointPlanformDihedral(1)
    TipTEPoint.Z.Value = TipTEPointPlanformDihedral(2)
    part1.Update
End If

```

```

'create TipChord's Line from "TipLEPoint" and "TipTEPoint"
'Dim TChordLine As HybridShapeLinePtPt
Set TChordLine = part1.FindObjectByName("TChordLine")
If TypeName(TChordLine) = "Nothing" Then
    Set TChordLine = hybridShapeFactory1.AddNewLinePtPt(TipLEPoint, TipTEPoint)
    TChordLine.Name = "TChordLine"
    OpenBody.AppendHybridShape TChordLine
    part1.InWorkObject = TChordLine
    part1.Update
Else
    TChordLine.PtOrigine = TipLEPoint
    TChordLine.PtExtremity = TipTEPoint
    part1.Update
End If

'create LeadingEdge's Line from "LocalOrigin" and "TipLEPoint"
'Dim LELinePlanform As HybridShapeLinePtPt
Set LELinePlanform = part1.FindObjectByName("LELine")
If TypeName(LELinePlanform) = "Nothing" Then
    Set LELinePlanform = hybridShapeFactory1.AddNewLinePtPt(LocalOrigin,
TipLEPoint)
    LELinePlanform.Name = "LELine"
    OpenBody.AppendHybridShape LELinePlanform
    part1.InWorkObject = LELinePlanform
    part1.Update
Else
    LELinePlanform.PtOrigine = LocalOrigin
    LELinePlanform.PtExtremity = TipLEPoint
    part1.Update
End If
'Dim LELinePlanformReference As Reference
Set LELinePlanformReference = part1.CreateReferenceFromObject(LELinePlanform)

'create TrailingEdge's Line from "RootTEPoint" and "TipTEPoint"
'Dim TELinePlanform As HybridShapeLinePtPt
Set TELinePlanform = part1.FindObjectByName("TELine")
If TypeName(TELinePlanform) = "Nothing" Then
    Set TELinePlanform = hybridShapeFactory1.AddNewLinePtPt(RootTEPoint,
TipTEPoint)
    TELinePlanform.Name = "TELine"
    OpenBody.AppendHybridShape TELinePlanform
    part1.InWorkObject = TELinePlanform
    part1.Update
Else
    TELinePlanform.PtOrigine = RootTEPoint

```

```

    TELinePlanform.PtExtremity = TipTEPoint
    part1.Update
End If
'Dim TELinePlanformReference As Reference
Set TELinePlanformReference = part1.CreateReferenceFromObject(TELinePlanform)

    Set oSelection = CATIA.ActiveDocument.Selection
    oSelection.Add LocalOrigin
    oSelection.Add TipLEPoint
    oSelection.Add RootTEPoint
    oSelection.Add TipTEPoint
    oSelection.VisProperties.SetShow 1
    oSelection.Clear
    Set oSelection = Nothing
Else
'#####
'#####
'#####
'#####
If PlanformType_cbx.Value = "Inboard-Outboard" Then
'MsgBox "SemiPlanformArea" & SemiPlanformArea
'TaperRatio
TaperRatio = CDBl(InboardOutboardPlanform.TaperRatio_tb.Text)
'Dim CrootProjection As Double
    CrootProjection = CDBl(InboardOutboardPlanform.InOutCRLength_tb.Text)
'Dim RootToKink As Double
    RootToKink = CDBl(InboardOutboardPlanform.RootToKink) '4245 '
'Dim KinkToTip As Double
    KinkToTip = SemiSpan - RootToKink
'Dim CtipProjection As Double
    CtipProjection = TaperRatio * CrootProjection
'Dim CkinkProjection As Double
    CkinkProjection = ((2 * SemiPlanformArea) - (CrootProjection * RootToKink) -
(CtipProjection * KinkToTip)) / (SemiSpan)

'definition Angle
Dim InboardSweepPlanform, InboardSweepPlanform_rad As Double
    InboardSweepPlanform = CDBl(InboardOutboardPlanform.InOutInboardSweep_tb.Text)
    InboardSweepPlanform_rad = InboardSweepPlanform * rad
Dim OutboardSweepPlanform, OutboardSweepPlanform_rad As Double
    OutboardSweepPlanform =
CDBl(InboardOutboardPlanform.InOutOutboardSweep_tb.Text)
    OutboardSweepPlanform_rad = OutboardSweepPlanform * rad
Dim RootDihedralAngle, RootDihedralAngle_rad As Double
    RootDihedralAngle = CDBl(InboardOutboardPlanform.InOutInboardDihedral_tb.Text)

```

```

    RootDihedralAngle_rad = RootDihedralAngle * rad
Dim KinkDihedralAngle, KinkDihedralAngle_rad As Double
    KinkDihedralAngle = CDBl(InboardOutboardPlanform.InOutOutboardDihedral_tb.Text)
    KinkDihedralAngle_rad = KinkDihedralAngle * rad

Dim delXKinkPlanform As Double 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    delXKinkPlanform = (RootToKink) * Tan(InboardSweepPlanform_rad)
'Dim delXTipPlanform As Double 'xxxxxxxxxxxxxxxxxxxxxxxxxxxx
    delXTipPlanform = (KinkToTip) * Tan(OutboardSweepPlanform_rad)

'definition TraillingEdge's point at Root
'Dim RTEPointPlanform(0 To 2) As Double
    RTEPointPlanform(0) = Origin(0) + CrootProjection
    RTEPointPlanform(1) = Origin(1)
    RTEPointPlanform(2) = Origin(2)
    RTEPointPlanformP = RTEPointPlanform
'create TraillingEdge's point at Root
'Dim RootTEPoint As HybridShapePointCoord
Set RootTEPoint = part1.FindObjectByName("RootTEPoint")
If TypeName(RootTEPoint) = "Nothing" Then
    Set RootTEPoint = hybridShapeFactory1.AddNewPointCoord(RTEPointPlanform(0),
RTEPointPlanform(1), RTEPointPlanform(2))
    RootTEPoint.Name = "RootTEPoint"
    OpenBody.AppendHybridShape RootTEPoint
    part1.InWorkObject = RootTEPoint
    part1.Update
Else
    RootTEPoint.X.Value = RTEPointPlanform(0)
    RootTEPoint.Y.Value = RTEPointPlanform(1)
    RootTEPoint.Z.Value = RTEPointPlanform(2)
    part1.Update
End If

'create RootChord's Line from LeadingEdge's point and TraillingEdge's point
'Dim RChordLine As HybridShapeLinePtPt
Set RChordLine = part1.FindObjectByName("RChordLine")
If TypeName(RChordLine) = "Nothing" Then
    Set RChordLine = hybridShapeFactory1.AddNewLinePtPt(LocalOrigin, RootTEPoint)
    RChordLine.Name = "RChordLine"
    OpenBody.AppendHybridShape RChordLine
    part1.InWorkObject = RChordLine
    part1.Update
Else
    RChordLine.PtOrigine = LocalOrigin
    RChordLine.PtExtremity = RootTEPoint
    part1.Update

```

End If

'definition LeadingEdge Planform's point at Kink

Dim KinkLEPointPlanform(0 To 2) As Double

KinkLEPointPlanform(0) = Origin(0) + delXKinkPlanform

KinkLEPointPlanform(1) = Origin(1) + (RootToKink)

KinkLEPointPlanform(2) = Origin(2)

KinkLEPointPlanformP = KinkLEPointPlanform

'definition TrailingEdge Planform's point at Kink

Dim KinkTEPointPlanform(0 To 2) As Double

KinkTEPointPlanform(0) = KinkLEPointPlanform(0) + CkinkProjection

KinkTEPointPlanform(1) = KinkLEPointPlanform(1)

KinkTEPointPlanform(2) = KinkLEPointPlanform(2)

KinkTEPointPlanformP = KinkTEPointPlanform

'definition LeadingEdge Planform's point at Tip

'Dim TipLEPointPlanform(0 To 2) As Double

TipLEPointPlanform(0) = KinkLEPointPlanform(0) + delXTipPlanform

TipLEPointPlanform(1) = KinkLEPointPlanform(1) + (KinkToTip)

TipLEPointPlanform(2) = KinkLEPointPlanform(2)

TipLEPointPlanformP = TipLEPointPlanform

'definition TrailingEdge Planform's point at Tip

'Dim TipTEPointPlanform(0 To 2) As Double

TipTEPointPlanform(0) = TipLEPointPlanform(0) + CtipProjection

TipTEPointPlanform(1) = TipLEPointPlanform(1)

TipTEPointPlanform(2) = TipLEPointPlanform(2)

TipTEPointPlanformP = TipTEPointPlanform

'DIHEDRAL

Dim RootDihedral, KinkDihedral As Double

RootDihedral = RootToKink * Tan(RootDihedralAngle_rad)

KinkDihedral = KinkToTip * Tan(KinkDihedralAngle_rad)

'definition LeadingEdge Planform's point with dihedral at Kink

Dim KinkLEPointPlanformDihedral(0 To 2) As Double

KinkLEPointPlanformDihedral(0) = KinkLEPointPlanform(0)

KinkLEPointPlanformDihedral(1) = KinkLEPointPlanform(1)

KinkLEPointPlanformDihedral(2) = KinkLEPointPlanform(2) + RootDihedral

KinkLEPointPlanformDihedralP = KinkLEPointPlanformDihedral

'create LeadingEdge Planform's point with dihedral at Tip

Dim KinkLEPoint As HybridShapePointCoord

Set KinkLEPoint = part1.FindObjectByName("KinkLEPoint")

If TypeName(KinkLEPoint) = "Nothing" Then

Set KinkLEPoint =

hybridShapeFactory1.AddNewPointCoord(KinkLEPointPlanformDihedral(0),

KinkLEPointPlanformDihedral(1), KinkLEPointPlanformDihedral(2))

KinkLEPoint.Name = "KinkLEPoint"

```

    OpenBody.AppendHybridShape KinkLEPoint
    part1.InWorkObject = KinkLEPoint
    part1.Update
Else
    KinkLEPoint.X.Value = KinkLEPointPlanformDihedral(0)
    KinkLEPoint.Y.Value = KinkLEPointPlanformDihedral(1)
    KinkLEPoint.Z.Value = KinkLEPointPlanformDihedral(2)
    part1.Update
End If
'Dim KinkLEPointReference As Reference
Set KinkLEPointReference = part1.CreateReferenceFromObject(KinkLEPoint)

'definition TrailingEdge Planform's point with dihedral at Kink
Dim KinkTEPointPlanformDihedral(0 To 2) As Double
    KinkTEPointPlanformDihedral(0) = KinkTEPointPlanform(0)
    KinkTEPointPlanformDihedral(1) = KinkTEPointPlanform(1)
    KinkTEPointPlanformDihedral(2) = KinkTEPointPlanform(2) + RootDihedral
    KinkTEPointPlanformDihedralP = KinkTEPointPlanformDihedral
'create LeadingEdge Planform's point with dihedral at Tip
Dim KinkTEPoint As HybridShapePointCoord
Set KinkTEPoint = part1.FindObjectByName("KinkTEPoint")
If TypeName(KinkTEPoint) = "Nothing" Then
    Set KinkTEPoint =
hybridShapeFactory1.AddNewPointCoord(KinkTEPointPlanformDihedral(0),
KinkTEPointPlanformDihedral(1), KinkTEPointPlanformDihedral(2))
    KinkTEPoint.Name = "KinkTEPoint"
    OpenBody.AppendHybridShape KinkTEPoint
    part1.InWorkObject = KinkTEPoint
    part1.Update
Else
    KinkTEPoint.X.Value = KinkTEPointPlanformDihedral(0)
    KinkTEPoint.Y.Value = KinkTEPointPlanformDihedral(1)
    KinkTEPoint.Z.Value = KinkTEPointPlanformDihedral(2)
    part1.Update
End If

'create KinkChord's Line from "KinkLEPoint" and "KinkTEPoint"
'Dim KChordLine As HybridShapeLinePtPt
Set KChordLine = part1.FindObjectByName("KChordLine")
If TypeName(KChordLine) = "Nothing" Then
    Set KChordLine = hybridShapeFactory1.AddNewLinePtPt(KinkLEPoint, KinkTEPoint)
    KChordLine.Name = "KChordLine"
    OpenBody.AppendHybridShape KChordLine
    part1.InWorkObject = KChordLine
    part1.Update
Else

```

```

KChordLine.PtOrigine = KinkLEPoint
KChordLine.PtExtremity = KinkTEPoint
part1.Update
End If

```

```

'definition LeadingEdge Planform's point with dihedral at Tip
'Dim TipLEPointPlanformDihedral(0 To 2) As Double
    TipLEPointPlanformDihedral(0) = TipLEPointPlanform(0)
    TipLEPointPlanformDihedral(1) = TipLEPointPlanform(1)
    TipLEPointPlanformDihedral(2) = TipLEPointPlanform(2) + KinkDihedral
    TipLEPointPlanformDihedralP = TipLEPointPlanformDihedral
'create LeadingEdge Planform's point with dihedral at Tip
'Dim TipLEPoint As HybridShapePointCoord
Set TipLEPoint = part1.FindObjectByName("TipLEPoint")
If TypeName(TipLEPoint) = "Nothing" Then
    Set TipLEPoint =
hybridShapeFactory1.AddNewPointCoord(TipLEPointPlanformDihedral(0),
TipLEPointPlanformDihedral(1), TipLEPointPlanformDihedral(2))
    TipLEPoint.Name = "TipLEPoint"
    OpenBody.AppendHybridShape TipLEPoint
    part1.InWorkObject = TipLEPoint
    part1.Update
Else
    TipLEPoint.X.Value = TipLEPointPlanformDihedral(0)
    TipLEPoint.Y.Value = TipLEPointPlanformDihedral(1)
    TipLEPoint.Z.Value = TipLEPointPlanformDihedral(2)
    part1.Update
End If

```

```

'definition TrailingEdge Planform's point with dihedral at Tip
'Dim TipTEPointPlanformDihedral(0 To 2) As Double
    TipTEPointPlanformDihedral(0) = TipTEPointPlanform(0)
    TipTEPointPlanformDihedral(1) = TipTEPointPlanform(1)
    TipTEPointPlanformDihedral(2) = TipTEPointPlanform(2) + KinkDihedral
    TipTEPointPlanformDihedralP = TipTEPointPlanformDihedral
'create TrailingEdge Planform's point with dihedral at Tip
'Dim TipTEPoint As HybridShapePointCoord
Set TipTEPoint = part1.FindObjectByName("TipTEPoint")
If TypeName(TipTEPoint) = "Nothing" Then
    Set TipTEPoint =
hybridShapeFactory1.AddNewPointCoord(TipTEPointPlanformDihedral(0),
TipTEPointPlanformDihedral(1), TipTEPointPlanformDihedral(2))
    TipTEPoint.Name = "TipTEPoint"
    OpenBody.AppendHybridShape TipTEPoint
    part1.InWorkObject = TipTEPoint

```

```

    part1.Update
Else
    TipTEPoint.X.Value = TipTEPointPlanformDihedral(0)
    TipTEPoint.Y.Value = TipTEPointPlanformDihedral(1)
    TipTEPoint.Z.Value = TipTEPointPlanformDihedral(2)
    part1.Update
End If
'create TipChord's Line from "TipLEPoint" and "TipTEPoint"
'Dim TChordLine As HybridShapeLinePtPt
Set TChordLine = part1.FindObjectByName("TChordLine")
If TypeName(TChordLine) = "Nothing" Then
    Set TChordLine = hybridShapeFactory1.AddNewLinePtPt(TipLEPoint, TipTEPoint)
    TChordLine.Name = "TChordLine"
    OpenBody.AppendHybridShape TChordLine
    part1.InWorkObject = TChordLine
    part1.Update
Else
    TChordLine.PtOrigine = TipLEPoint
    TChordLine.PtExtremity = TipTEPoint
    part1.Update
End If

'LEADINGEDGE AND TRAILINGEDGE LINE
'create InboardLeadingEdge's Line from "LocalOrigin" and "KinkLEPoint"
'Dim InboardLELinePlanform As HybridShapeLinePtPt
Set InboardLELinePlanform = part1.FindObjectByName("InboardLELinePlanform")
If TypeName(InboardLELinePlanform) = "Nothing" Then
    Set InboardLELinePlanform = hybridShapeFactory1.AddNewLinePtPt(LocalOrigin,
KinkLEPoint)
    InboardLELinePlanform.Name = "InboardLELinePlanform"
    OpenBody.AppendHybridShape InboardLELinePlanform
    part1.InWorkObject = InboardLELinePlanform
    part1.Update
Else
    InboardLELinePlanform.PtOrigine = LocalOrigin
    InboardLELinePlanform.PtExtremity = KinkLEPoint
    part1.Update
End If
'Dim LELinePlanformReference As Reference
Set InLELinePlanformReference =
part1.CreateReferenceFromObject(InboardLELinePlanform)

'create InboardTrailingEdge's Line from "RootTEPoint" and "KinkTEPoint"
'Dim InboardTELinePlanform As HybridShapeLinePtPt
Set InboardTELinePlanform = part1.FindObjectByName("InboardTELinePlanform")
If TypeName(InboardTELinePlanform) = "Nothing" Then

```

```

    Set InboardTELinePlanform = hybridShapeFactory1.AddNewLinePtPt(RootTEPoint,
KinkTEPoint)
    InboardTELinePlanform.Name = "InboardTELinePlanform"
    OpenBody.AppendHybridShape InboardTELinePlanform
    part1.InWorkObject = InboardTELinePlanform
    part1.Update
Else
    InboardTELinePlanform.PtOrigine = RootTEPoint
    InboardTELinePlanform.PtExtremity = KinkTEPoint
    part1.Update
End If

'create OutboardLeadingEdge's Line from "KinkLEPoint" and "TipLEPoint"
'Dim OutboardLELinePlanform As HybridShapeLinePtPt
Set OutboardLELinePlanform = part1.FindObjectByName("OutboardLELinePlanform")
If TypeName(OutboardLELinePlanform) = "Nothing" Then
    Set OutboardLELinePlanform = hybridShapeFactory1.AddNewLinePtPt(KinkLEPoint,
TipLEPoint)
    OutboardLELinePlanform.Name = "OutboardLELinePlanform"
    OpenBody.AppendHybridShape OutboardLELinePlanform
    part1.InWorkObject = OutboardLELinePlanform
    part1.Update
Else
    OutboardLELinePlanform.PtOrigine = KinkLEPoint
    OutboardLELinePlanform.PtExtremity = TipLEPoint
    part1.Update
End If
'Dim LELinePlanformReference As Reference
Set OutLELinePlanformReference =
part1.CreateReferenceFromObject(OutboardLELinePlanform)

'create OutboardTrailingEdge's Line from "RootTEPoint" and "TipTEPoint"
'Dim OutboardTELinePlanform As HybridShapeLinePtPt
Set OutboardTELinePlanform = part1.FindObjectByName("OutboardTELinePlanform")
If TypeName(OutboardTELinePlanform) = "Nothing" Then
    Set OutboardTELinePlanform = hybridShapeFactory1.AddNewLinePtPt(KinkTEPoint,
TipTEPoint)
    OutboardTELinePlanform.Name = "OutboardTELinePlanform"
    OpenBody.AppendHybridShape OutboardTELinePlanform
    part1.InWorkObject = OutboardTELinePlanform
    part1.Update
Else
    OutboardTELinePlanform.PtOrigine = KinkTEPoint
    OutboardTELinePlanform.PtExtremity = TipTEPoint
    part1.Update
End If

```

End If

```
Set oSelection = CATIA.ActiveDocument.Selection
oSelection.Add LocalOrigin
oSelection.Add RootTEPoint
oSelection.Add KinkLEPoint
oSelection.Add KinkTEPoint
oSelection.Add TipTEPoint
oSelection.Add TipLEPoint
oSelection.VisProperties.SetShow 1
oSelection.Clear
Set oSelection = Nothing
```

End If

```
part1.InWorkObject = OpenBody
part1.Update
```

```
WireframeSection_cb.Locked = Fals
WireframeSection_cb.ForeColor = &H80000012
.....
```