

Dokumentasi Coding PSR David untuk Stress Checking

Pendahuluan

Semua *stress checking* di laporan PSR saya dilakukan di kalkulator ini. Kalkulator ini dapat menerima *input* bentuk *airfoil/fuselage* dalam bentuk titik dan garis. Setiap titik bisa didefinisikan luasnya (luas *stringer*). Setiap garis bisa didefinisikan tebalnya. Kalkulator ini utamanya digunakan untuk menghitung *margin of safety* (MoS) dari suatu profil *airfoil/fuselage*. MoS yang dimaksud adalah terhadap kegagalan karena *Von Mises Criterion* dan *buckle*. Dengan demikian, kalkulator ini, bisa digunakan untuk mengiterasi *initial sizing* dengan mudah dan cepat sampai didapat hasil yang optimum.

Berikut adalah perhitungan yang dilakukan di kalkulator:

- Idealisasi luas tembok menjadi titik
- Perhitungan *shear flow* dan *shear stress* karena gaya *shear* dan *torsion*
- Perhitungan *normal stress* karena momen *bending*
- Perhitungan MoS dari sisi *Von Mises Criterion*
- Perhitungan *buckling* lokal karena panel ber-*stringer* yang terkompresi (ESDU 71014)
- Perhitungan *buckling shear* pada panel (ESDU 71005)
- Perhitungan MoS dari sisi *buckling*
- Perhitungan massa per satuan panjang *spanwise* dari profil

Code ini sudah divalidasi kebenaran perhitungannya melalui soal *shear flow* dari tutor dan ppt. Note: buku Niu ada contoh soal yang jawabannya salah. Jadi jangan bandingin ke buku Niu.

Bagian utama code yang digunakan

Code terdiri dari banyak *file*. Yang perlu diedit hanyalah:

- Main_Input.m

Berisi input data geometri struktur. Geometri dibentuk dengan kelas Point, Wall, dan Loop.

- Main.m

Berisi input gaya yang dialami struktur

- Main_Buckle1.m

Berisi penentuan nilai K dari ESDU 71014 (bergantung dari dimensi *stringer*). Nilai K harus diisi manual dari ESDU 71014. Maaf saya belum bisa automasi.

- DaftarMaterial.m

Berisi data material yang digunakan.

Yang perlu di run hanyalah Main.m dan Visualisasi.mlx

Selain code, tersebut, ada 5 file tambahan yang tidak digunakan di code (bisa dihapus) tetapi bisa digunakan untuk menyimpan input supaya tidak susah-susah lagi:

- DaftarInputLoad.m
- DaftarInputDavid1.m
- DaftarInputDavid2.m
- DaftarInputDavid3.m
- DaftarInput.m

Penentuan Material

Material dapat didefinisikan dengan kelas material.

```
classdef Material
    properties
        name string
        E(1,1) double %modulus young dalam GPa
        G(1,1) double %shear modulus dalam GPa
        sigmaY(1,1) double %yield strength dalam MPa
        rho(1,1) double %dalam kg/m^3
    end
    methods
        function obj = Material(name,E,G,sigmaY,rho)
```

Contoh di file DaftarMaterial.m

```
Al7075 = Material('Al7075-T6',71.7,26.9,503,2810);
Al2024 = Material('Al2024-T4',73.1,28,324,2780);
```

Variabel material ini (seperti Al7075) nanti diletakan sebagai input geometri di kelas Point dan Wall.

Penentuan Geometri

Penentuan geometri semua dilakukan di file Main_Input.m melalui kelas Point di variabel p, kelas Wall di variabel w, dan kelas Loop di variabel loop. Point di sini adalah *stringer* dan pertigaan antara *skin* dan *spar*. Wall di sini adalah garis yang menghubungkan 2 point. Loop di sini adalah pendefinisian adanya loop di profil geometri. Setiap loop di geometri harus didefinisikan masing-masing. Hasil geometri bisa dilihat di Visualisasi.mlx atau dengan run file Main_Input.m dan *uncomment* code *figure* di bawah.

Kelas Point

Point(n,pos,material,A)

n adalah nama titik.

pos adalah posisi koordinat titik dalam m. Ekspresikan dalam [x y].

material masukkan kelas material. Contoh Al2024.

A luas dalam mm². Luas bisa diisi nol.

Variabel p berisi kelas-kelas Point yang membentuk geometri. Kalau bisa diurutkan.

Kelas Wall

Wall(n,p1,p2,material,t)

n adalah nama wall.

p1 dan p2 adalah kelas titik yang menghubungkan garis wall.

material masukkan kelas material. Contoh Al2024.

t tebal dalam mm.

Variabel w berisi kelas-kelas Wall yang membentuk geometri. Kalau bisa diurutkan dari *skin* angka rendah sampai angka tinggi, lalu dilanjutkan dengan *spar* di bawah. Anda bisa pakai fungsi `AutoWall(p,material,t)` untuk otomatis membuat *wall skin* dari point yang ada (contoh penggunaan di contoh *advance*).

Kelas Loop

Loop (angka, p, w)

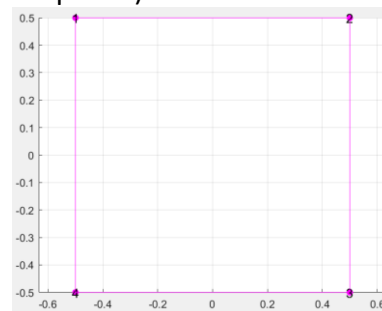
angka diisi indeks-indeks point yang membentuk loop (harus berurutan). Indeks ini sesuai urutan kelas Point di variabel p. Karena itu, variabel p sebaiknya isinya berurutan supaya tidak pusing. Contoh pengisiannya ada di bawah.

p dan w jangan diubah.

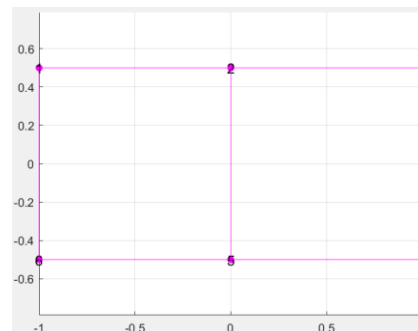
Variabel loop berisi daftar kelas Loop.

Contoh pengisian sederhana (bisa dilihat di file DaftarInput.m)

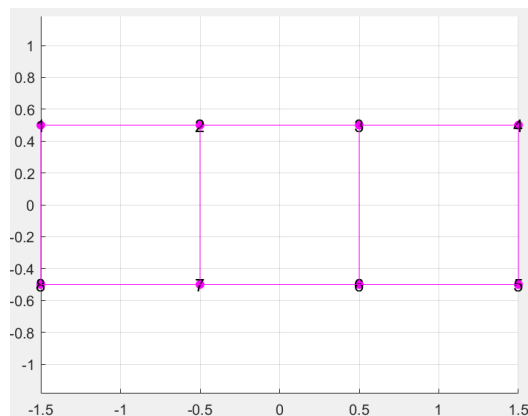
```
%% 1 cell
p = [Point('1',[-0.5 0.5],Al2024,0);
     Point('2',[0.5 0.5],Al2024,0);
     Point('3',[0.5 -0.5],Al2024,0);
     Point('4',[-0.5 -0.5],Al2024,0)];
w = [Wall('1_2',p(1),p(2),Al2024,1);
     Wall('2_3',p(2),p(3),Al2024,1);
     Wall('3_4',p(3),p(4),Al2024,1);
     Wall('4_1',p(4),p(1),Al2024,1)];
loop = [Loop([1 2 3 4],p,w)];
```



```
%% 2 cell
p = [Point('1',[-1 0.5],Al2024,0);
     Point('2',[0 0.5],Al2024,0);
     Point('3',[1 0.5],Al2024,0);
     Point('4',[1 -0.5],Al2024,0);
     Point('5',[0 -0.5],Al2024,0);
     Point('6',[-1 -0.5],Al2024,0)];
w = [Wall('1_2',p(1),p(2),Al2024,1);
     Wall('2_3',p(2),p(3),Al2024,1);
     Wall('3_4',p(3),p(4),Al2024,1);
     Wall('4_5',p(4),p(5),Al2024,1);
     Wall('5_6',p(5),p(6),Al2024,1);
     Wall('6_1',p(6),p(1),Al2024,1);
     Wall('2_5',p(2),p(5),Al2024,1)];
loop = [Loop([1 2 5 6],p,w);
        Loop([2 3 4 5],p,w)];
```

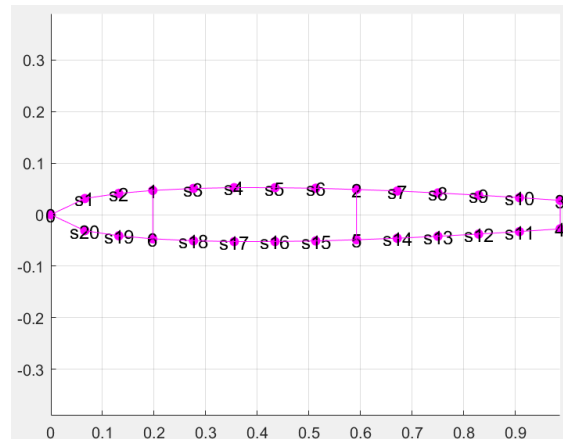


```
%% 3 cell
p = [Point('1',[-1.5 0.5],Al2024,0);
     Point('2',[-0.5 0.5],Al2024,0);
     Point('3',[0.5 0.5],Al2024,0);
     Point('4',[1.5 0.5],Al2024,0);
     Point('5',[1.5 -0.5],Al2024,0);
     Point('6',[0.5 -0.5],Al2024,0);
     Point('7',[-0.5 -0.5],Al2024,0);
     Point('8',[-1.5 -0.5],Al2024,0)];
w = [Wall('1_2',p(1),p(2),Al2024,1);
     Wall('2_3',p(2),p(3),Al2024,1);
     Wall('3_4',p(3),p(4),Al2024,1);
     Wall('4_5',p(4),p(5),Al2024,1);
     Wall('5_6',p(5),p(6),Al2024,1);
     Wall('6_7',p(6),p(7),Al2024,1);
     Wall('7_8',p(7),p(8),Al2024,1);
     Wall('8_1',p(8),p(1),Al2024,1);
     Wall('2_7',p(2),p(7),Al2024,1);
     Wall('3_6',p(3),p(6),Al2024,1)];
loop = [Loop([1 2 7 8],p,w);
        Loop([2 3 6 7],p,w);
        Loop([3 4 5 6],p,w)];
```



Contoh pengisian lebih *advance*. Dibuat banyak variabel supaya tidak pusing hitung angka. Jumlah point juga lebih banyak karena ada banyak *stringer*. Pakai fungsi NACA0008 untuk ngebentuk airfoil. Bisa dilihat di DaftarInputDavid1.m.

```
t = 2;
tmidl = 2.2;
tmidu = 2.1;
tLE = 1.6;
A_st = t*40-2*t^2;
A_sparf = 3*30*4-2*4^2;
p = [Point('s1',c*NACA0008(5,1),Al2024,A_st);%1
Point('s2',c*NACA0008(10,1),Al2024,A_st);%2
Point('1',c*NACA0008(15,1),Al2024,A_sparf);%3
Point('s3',c*NACA0008(21,1),Al2024,A_st);%4
Point('s4',c*NACA0008(27,1),Al2024,A_st);%5
Point('s5',c*NACA0008(33,1),Al2024,A_st);%6
Point('s6',c*NACA0008(39,1),Al2024,A_st);%7
Point('2',c*NACA0008(45,1),Al2024,A_sparf);%8
Point('s7',c*NACA0008(51,1),Al2024,A_st);%9
Point('s8',c*NACA0008(57,1),Al2024,A_st);%10
Point('s9',c*NACA0008(63,1),Al2024,A_st);%11
Point('s10',c*NACA0008(69,1),Al2024,A_st);%12
Point('3',c*NACA0008(75,1),Al2024,A_sparf);%13
Point('4',c*NACA0008(75,-1),Al2024,A_sparf);%14
Point('s11',c*NACA0008(69,-1),Al2024,A_st);%15
Point('s12',c*NACA0008(63,-1),Al2024,A_st);%16
Point('s13',c*NACA0008(57,-1),Al2024,A_st);%17
Point('s14',c*NACA0008(51,-1),Al2024,A_st);%18
Point('5',c*NACA0008(45,-1),Al2024,A_sparf);%19
Point('s15',c*NACA0008(39,-1),Al2024,A_st);%20
Point('s16',c*NACA0008(33,-1),Al2024,A_st);%21
Point('s17',c*NACA0008(27,-1),Al2024,A_st);%22
Point('s18',c*NACA0008(21,-1),Al2024,A_st);%23
Point('6',c*NACA0008(15,-1),Al2024,A_sparf);%24
Point('s19',c*NACA0008(10,-1),Al2024,A_st);%25
Point('s20',c*NACA0008(5,-1),Al2024,A_st);%26
Point('0',c*NACA0008(0,1),Al2024,0)];%27
w = [AutoWall(p,Al2024,t);
Wall('6_1',findP(p,'6'),findP(p,'1'),Al2024,2);
Wall('2_5',findP(p,'2'),findP(p,'5'),Al2024,2)];
for i = 3:7
    w(i).t = tmidu; w(i).area = recalculateArea(w(i));
end
for i = 19:23
    w(i).t = tmidl; w(i).area = recalculateArea(w(i));
end
for i = [1:2 24:27]
    w(i).t = tLE; w(i).area = recalculateArea(w(i));
end
loop = [Loop([3 4 5 6 7 8 19 20 21 22 23 24],p,w);
Loop([8 9 10 11 12 13 14 15 16 17 18 19],p,w);
Loop([1 2 3 24 25 26 27],p,w)];
```



Penentuan Gaya Internal

Gaya yang dialami profil struktur ditentukan di Main.m.

```
clear all
%input chord airfoil
c = 1.316;
%kond Lift dalam
V = 16811; %dalam N
d = c/2; %off set dari titik x=0 dalam m
T = -2895; %dalam Nm
M = 7279;
%kond Lift luar
% V = -16942; %dalam N
% d = c/2; %off set dari titik x=0 dalam m
% T = 4153; %dalam Nm
% M = -7339;

%input data ribs
L_rib = 0.724; %jarak antar ribs dalam m
```

Ada 5 variabel utama yang wajib diisi:

V adalah gaya internal *shear* dalam N. Positif ke arah atas.

d adalah posisi gaya internal *shear* dari titik x=0 dalam m.

T adalah torsi yang dialami dalam Nm. Positif CCW.

M adalah momen yang dialami ke dalam Nm. Positif ke arah atas (compresi di atas, tension di bawah).

L_rib adalah jarak antar rib, digunakan untuk perhitungan *buckling shear* di skin.

