Dokumen Pengembangan TRIAC
(TRIso *Analysis Code*)

LABORATORIUM KOMPUTASI

PUSAT TEKNOLOGI DAN KESELAMATAN REAKTOR NUKLIR

Disusun oleh:
Arya Adhyaksa Waskita

Supervisor:
Dr. Topan Setiadipura

31 Juli 2017

Daftar Isi

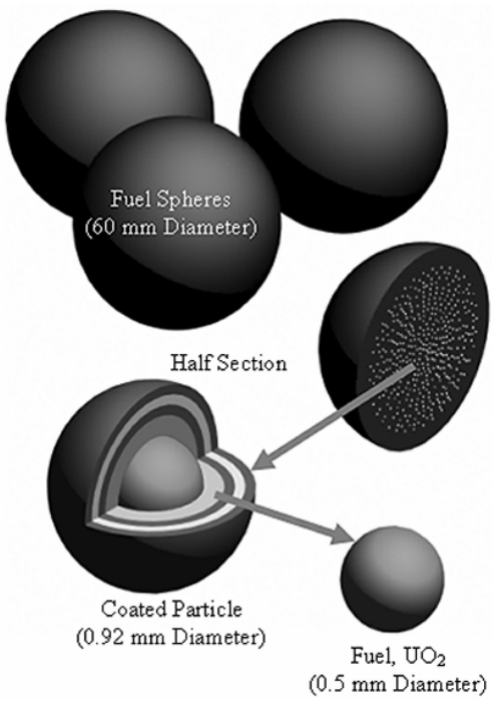
Daftar Gambar

Daftar Program

BAB 1

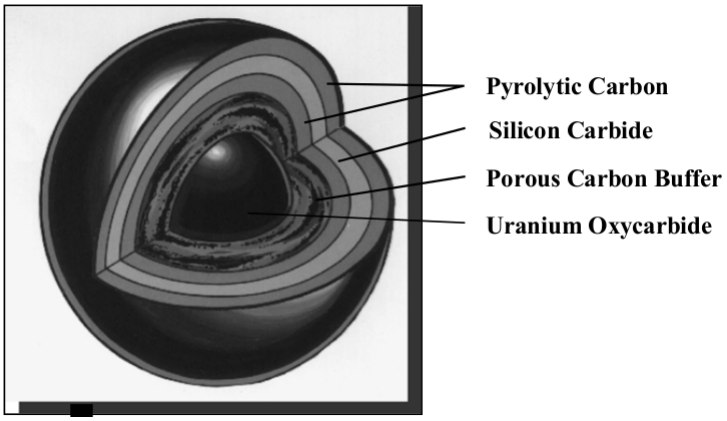
Pendahuluan

BATAN saat ini tengah berencana membangun reaktor riset baru berbasis HTGR (*High Temperature Gas-cooled Reactor*) [1] sebagai persiapan PLTN, yang akan dibangun di Indonesia di masa depan [2]. Salah satu yang perlu diperhatikan dalam pengembangan reaktor jenis ini adalah bahan bakarnya yang berjenis *pebble* yang bentuknya dapat diilustrasikan seperti pada Gambar 1.1. Bahan bakar harus dirancang sedemikian rupa sehingga rasio gagalnya bahan bakar selama operasi minimal.



Gambar 1.1: Ilustrasi bentuk bahan bakar *pebble* [1]

Bahan bakar berjenis *pebble* ini memiliki komponen utama yang dalam Gambar 1.1 disebut sebagai *coated particle*. Komposisi elemen pelapis (*coated*) dapat diilustrasikan dalam Gambar 1.2. Dalam upaya



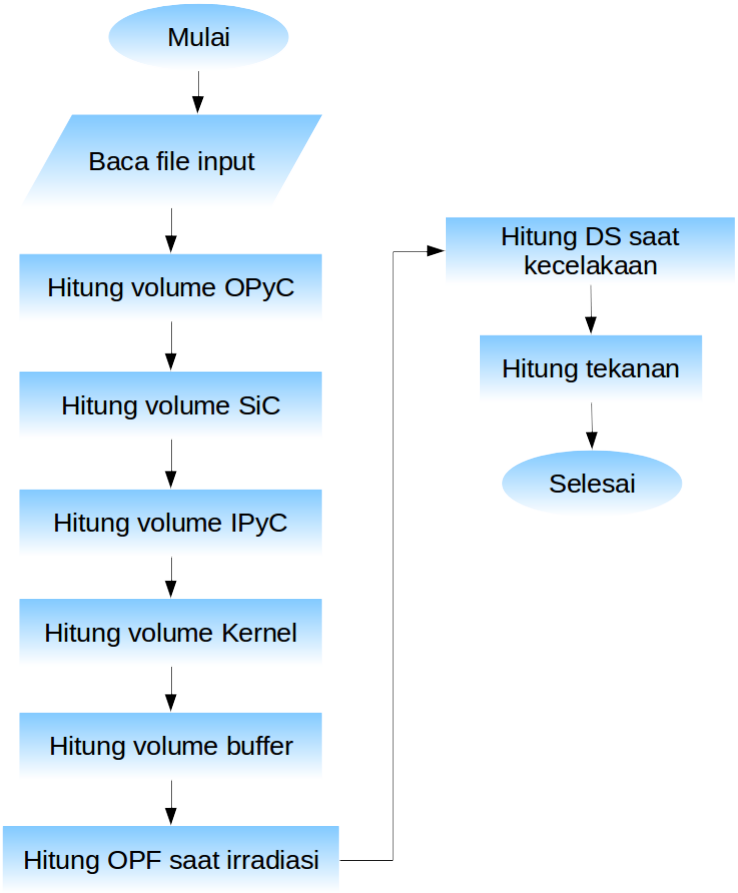
Gambar 1.2: Komposisi elemen pelapis partikel [1]

BAB 2

Alur Perhitungan

2.1 Pendahuluan

Secara umum, perhitungan TRIAC mengikuti diagram alir seperti pada Gambar 2.1 berikut. Sementara kode sumbernya disajikan dalam Listing 2.1.



Gambar 2.1: Diagram alir perhitungan TRAIC

Listing 2.1: triac.py

```
import math,sys

def readdata(namafile):
        f=open(namafile,"r")
        statusGeometry="[m]"
        statusCharacteristics="SiC Tensile Strength [Pa]"
        statusIrradiation="INPUT: Irradiation Temp. Hystory"
        statusAccident="INPUT: Accident Temp. Hystory"
        statusAll=0
        dimensi=[]
        characteristics=[]
        irradiation=[]
        accident=[]
        i=0
        x=0
        for baris in f.readlines():
                i=i+1
                element=baris.split('\t')
                if statusAll==0:
                        if element[0]==statusGeometry:
                                for j in range(1,6):
                                        y=float(element[j])
                                        dimensi.append(y)
                                statusAll=1

                elif statusAll==1:
                        if element[0]==statusCharacteristics:
                                x=i+1
                        elif i==x:
                                try:
                                        for j in range(0,5):
                                                y=float(element[j])
                                                characteristics.append(y)
                                                statusAll=2
                                except:
                                        x=i+1

                elif statusAll==2:
                        if element[0]==statusIrradiation:
                                x=i+1
                        elif element[0]==statusAccident:
                                statusAll=3
                        else:
                                temp=[]
                                try:
                                        y=int(element[0])
                                        temp.append(y)
                                        y=float(element[1])
                                        temp.append(y)
                                        y=int(element[2])
                                        temp.append(y)
                                        irradiation.append(temp)
                                except:
                                        x=i+1

                elif statusAll==3:
                        temp=[]
                        try:
                                y=int(element[0])
                                temp.append(y)
                                y=float(element[1])
                                temp.append(y)
                                y=int(element[2])
                                temp.append(y)
                                accident.append(temp)
                        except:
                                x=i+1

        return dimensi, characteristics, irradiation, accident
```

```
70
71
72  def OPF(irradiation):
73          x=len(irradiation)
74          print("Irradiation length:",x)
75          y=17*24*3600
76          z=0.0
77          for i in range(x):
78                  j=irradiation[i]
79                  g=2*(8.32e-11)*(math.exp(-163000/(8.3143*j[2])))*(1020-j[1])*24*3600*y
80                  z=z+g
81
82          """print("OPF:",z)"""
83
84          tb=1020*24*3600
85          Tb=0.85e4/((2*math.log10(tb))-(math.log10(z))-10.08)
86          """print("Tb:",Tb)"""
87          logDS=-2.3-(8116/Tb)
88          ds=math.pow(10,logDS)
89          """print("DS:",ds)"""
90          taui=ds*tb
91          """print("taui:",taui)"""
92          return z,Tb,ds,taui
93
94  def FD(tauA,tauI):
95          summasi=tauA+tauI
96          looping=0.0
97          loopingPlus=0.0
98          loopingI=0.0
99          for n in range (1,2000):
100                 pangkat=math.pow(n,2)*math.pow(math.pi,2)*tauA
101                 A=math.exp(-(pangkat))
102                 B=math.pow(n,4)*math.pow(math.pi,4)
103                 looping=looping+((1-(A)/(B)))
104
105                 pangkatI=math.pow(n,2)*math.pow(math.pi,2)*tauI
106                 AI=math.exp(-(pangkatI))
107                 loopingI=loopingI*((1-(AI)/(B)))
108                 if tauA==0:
109                         """print(n,pangkatI,AI,B,AI/B,loopingI)"""
110                 loopingPlus=loopingPlus+(1-(math.exp(-(math.pow(n,2)*math.pow(math.pi,2)*summasi)))/(
111         if tauA==0:
112                 FTau=1-((6/tauI)*loopingI)
113                 print('tauI=',tauI)
114                 print('loopingI=',loopingI)
115                 print('ftauI=',FTau)
116                 Fd=FTau
117         else:
118                 FTau=1-((6/tauA)*looping)
119                 FTauPlus=1-((6/summasi)*loopingPlus)
120                 Fd=((summasi*FTauPlus)-(tauA*FTau))/tauI
121         return Fd
122
123 def OPFAccident(Tb,T):
124         tb=1020*24*3600
125         logOPF=-10.08-(8500/Tb)+(2*math.log10(tb))-(0.404*((10000/T)-(10000/(Tb+75))))
126         return math.pow(10,logOPF)
127
128 def Pressure(Tb,dsAccident,Vk,Vf,accident):
129         Ff=0.31
130         R=8.3143
131         Vm=2.43796e-5
132         Fb=0.08
133         p=[]
134         for i in range(len(accident)):
135                 x=accident[i]
136                 y=dsAccident[i]
137                 p.append((y[2]*Ff*OPFAccident(Tb,x[2])*Fb)/((Vf/Vk)*R*x[2]/Vm))
138         return p
139
```

```
def DS(accident,tauI):
        x=len(accident)
        ds=[]
        for i in range(x):
                j=accident[i]
                logDS=-2.3-(8116/j[2])
                k=[]
                l=math.pow(10,logDS)
                k.append(l)
                tauA=l*j[1]*24*3600
                k.append(tauA)
                Fd=FD(tauA,tauI)
                k.append(Fd)
                ds.append(k)
        return ds

if __name__=="__main__":
        if len(sys.argv)==2:
                f=sys.argv[1]
        else:
                f="example.pan.in"

        x=readdata(f)
        dimensi=x[0]
        characteristics=x[1]
        irradiation=x[2]
        accident=x[3]
        """print("Dimensi:",dimensi)
        print("Karakteristik:",characteristics)
        print("Irradiasi:",irradiation)
        print("Accident:",accident)"""
        """Volume Outer Pyrolitic Carbon"""
        VolRef1=(4/3)*math.pi*dimensi[0]*dimensi[0]*dimensi[0]
        VolRef2=(4/3)*math.pi*dimensi[1]*dimensi[1]*dimensi[1]
        VolOPyC=VolRef1-VolRef2

        """Volume SiC"""
        VolRef3=(4/3)*math.pi*dimensi[2]*dimensi[2]*dimensi[2]
        VolSiC=VolRef2-VolRef3

        """Volume IPyC"""
        VolRef4=(4/3)*math.pi*dimensi[3]*dimensi[3]*dimensi[3]
        VolIPyC=VolRef3-VolRef4

        """Volume Buffer & Volume Kernel"""
        VolKernel=(4/3)*math.pi*dimensi[4]*dimensi[4]*dimensi[4]
        VolBuff=VolRef4-VolKernel
        print("Volume OPyC:",VolOPyC)
        print("Volume SiC:",VolSiC)
        print("Volume IPyC:",VolIPyC)
        print("Volume Buffer:",VolBuff)
        print("Volume Kernel:",VolKernel)
        TB=OPF(irradiation)
        z=TB[0]
        Tb=TB[1]
        ds=TB[2]
        taui=TB[3]
        print("OPF=",z,"Tb=",Tb,"DS=",ds,"TauI=",taui)
        dsAccident=DS(accident,TB[3])
        for i in range(len(dsAccident)):
                x=dsAccident[i]
                print(x[0],x[1],x[2])

        p=Pressure(Tb,dsAccident,VolKernel,VolBuff/2,accident)
        print(p)
```

2.2 Membaca *file input*

Sub rutin ini ditujukan untuk membaca file input dengan format seperti terdapat pada Lampiran 1. Sub rutin ini menggunakan skema yang kaku karena identifikasi nilai-nilai yang akan dibaca ditentukan oleh suatu teks tertentu. Setelah teks yang menjadi penanda, nilai-nilai yang dibutuhkan dibaca. Tetapi, nilai tersebut dapat langsung berada dalam satu baris bersama dengan teks penanda, atau berada pada baris yang berbeda. Sub rutin ini terdapat pada baris ke-3 s/d baris ke-69 dalam Listing 2.1

Terdapat empat jenis data yang perlu dibaca dari *file input* dalam Lampiran 1, masing-masing adalah sebagai berikut.

1. Data tentang geometri *pebble*. Data ini diidentifikasi menggunakan teks yang didefinisikan oleh variabel `statusGeometry` (baris ke-5 pada Listing 2.1). Di dalam data geometri, terdapat empat data berbeda, masing-masing secara berurutan adalah panjang jejari *pebble* terluar, OPyC (*Outer Pyrolitic Carbon*), SiC (*Silicon Carbide*), IPyC (*Inner Pyrolitic Carbon*), *buffer* dan kernel.

2. Data tentang kekuatan SiC. Data ini diidentifikasi menggunakan teks yang didefinisikan oleh varibel `statusCharacteristics` (baris ke-6 pada Listing 2.1). Ada empat nilai yang perlu dibaca terkait kekuatan SiC, masing-masing adalah SiC *Tensile Strength* [Pa], *Weibull Modulus Burnup* [FIMA], *Fission Yield of stable fission gasses* [Ff], *Fast Neutron Fluence* dan rasio berat Th terhadap U-235 pada kernel.

3. Data tentang sejarah irradiasi. Data ini diidentifikasi menggunakan teks yang didefinisikan oleh variabel `statusIrradiation` (baris ke-) Listing 2.1. Data ini merupakan data temperatur bahan bakar *pebble* pada selang waktu tertentu. Sebagai contoh, data yang disajikan pada Lampiran 1 diambil pada selang waktu 17 hari.

.

Daftar Referensi

[1] J. Wang, "An integrated performance model for high temperature gas cooled reactor coated particle fuel," Ph.D. dissertation, Massachusetts Institute of Technology, 2004.

[2] "Reaktor daya eksperimental (rde)," http://www.batan.go.id/index.php/id/reaktor-daya-eksperimental-rde, diakses: 17-07-2017.

LAMPIRAN

Lampiran 1

TRIAC-BATAN
TRISO Analysis Code of BATAN
"Developed by Computational Laboratory, Center for Nuclear Reactor Technology and Safety, BATAN"

Case Title: (describe your problem case here)

TRISO Geometry:
Outer radius CFP SiC IPyC buffer kernel center

[m] 4.60E-04 4.20E-04 3.85E-04 3.45E-04 2.50E-04 0
Properties and Operation Parameters:

SiC Tensile Strength [Pa] Weibull Modulus Burnup [FIMA] "Fission Yield of stable fission gasses, Ff" Fast Neutron Fluence Weight ratio of th to U-235 in kernel

8.34E+08 8.02 0.09 0.31 2.4
Properties and Operation Parameters related with thermal decomposition:

Alpha Beta
0.0001 4
-1 1401.6 0.1 10
INPUT: Irradiation Temp. Hystory

1	0	593
2	17	833
3	34	1023
4	51	1093
5	68	1123
6	85	833
7	102	1023
8	119	1093
9	136	1123
10	153	833
11	170	1023
12	187	1093
13	204	1123
14	221	833
15	238	1023
16	255	1093
17	272	1123
18	289	833
19	306	1023
20	323	1093
21	340	1123
22	357	833
23	374	1023
24	391	1093
25	408	1123
26	425	833
27	442	1023
28	459	1093
29	476	1123
30	493	833
31	510	1023
32	527	1093
33	544	1123
34	561	833
35	578	1023
36	595	1093
37	612	1123

```
38    629   833
39    646   1023
40    663   1093
41    680   1123
42    697   833
43    714   1023
44    731   1093
45    748   1123
46    765   833
47    782   1023
48    799   1093
49    816   1123
50    833   833
51    850   1023
52    867   1093
53    884   1123
54    901   833
55    918   1023
56    935   1093
57    952   1123
58    969   833
59    986   1023
60    1003  1093
61    1020  1123

      0
      -1    180   1
INPUT: Accident Temp. Hystory

1     0     1033
2     0.0271      1033
3     0.2208      1068
4     1     1160
5     10    1571
6     20    1728
7     30    1752
8     35    1749
9     60    1690
10    90    1605
11    120   1526
12    180   1395

      0
```