

LAPORAN PENELITIAN



PEMBUATAN KODE PROGRAM DAN SIMULASI SKEMA MASAKAN ACD MENGGUNAKAN JUPYTER NOTEBOOK PADA PEMOGRAMAN PYTHON ANACONDA

HENDRI RANTAU., S.T., M.Eng

NIDN. 0503018402

**PROGRAM STUDI TEKNOLOGI KIMIA
POLITEKNIK LPP
YOGYAKARTA
2022**

LEMBAR PENGESAHAN

Judul Penelitian : Pembuatan kode program dan simulasi skema masakan ACD menggunakan Jupyter Notebook pada pemrograman Python Anaconda

Nama Peneliti : Hendri Rantau., S.T., M.Eng

NIDN : 0503018402

Jabatan Fungsional : Tenaga Pengajar

Program Studi : Teknologi Kimia

Nomor HP : 0813265757658

E-Mail : rnt@polteklpp.ac.id

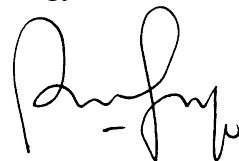
Sumber Pendanaan : Mandiri

Menyetujui
Kepala UPPM,



Lestari Hetalesi Saputri, S.T., M.Eng.
NIDN: 0525108401

Dosen Peneliti,
Yogyakarta, 25 Februari 2022



Hendri Rantau., S.T., M.Eng.
NIDN: 0503018402

Mengetahui,
Wakil Direktur I Bidang Akademik



Ratna Sri Harjanti, S.T., M.Eng.
NIP: 197802202005012002

RINGKASAN

Industri gula merupakan salah satu industri bahan pokok di Indonesia. Keberadaannya sangat diperlukan dikarenakan menghasilkan gula yang dibutuhkan oleh masyarakat pada umumnya. Dalam perkembangannya, industri gula dituntut untuk dapat beradaptasi dengan perkembangan teknologi saat ini agar dapat menghasilkan produk dengan efisiensi tinggi. Salah satu perkembangan teknologi yang saat ini adalah simulasi proses. Simulasi proses dibutuhkan untuk dapat memprediksi proses industri sehingga ketidaknormalan dalam proses dapat ditangani sebelum terjadi agar dapat mengurangi cacat produk.

Salah satu simulasi proses yang dapat dimanfaatkan dalam industri gula adalah simulasi skema masakan dengan menggunakan program Jupyter Notebook dalam pemrograman Python Anaconda.

Penelitian dilakukan selama bulan 3 Januari - 25 Februari 2022 di Politeknik LPP Yogyakarta. Penelitian dilakukan dalam membuat kode program dan melakukan simulasi dari kode program yang didapat.

Hasil penelitian menunjukkan bahwa simulasi skema masakan dapat dilakukan dengan menggunakan Jupyter Notebook pada pemrograman Python Anaconda

Kata Kunci : Simulasi, Jupyter, Python, Anaconda, Masakan

DAFTAR ISI

HALAMAN JUDUL.....	ii
LEMBAR PENGESAHAN.....	iii
RINGKASAN.....	iv
DAFTAR ISI.....	v
BAB I PENDAHULUAN.....	1
BAB II TINJAUAN PUSTAKA.....	2
BAB III METODE PENELITIAN.....	4
BAB IV HASIL DAN PEMBAHASAN.....	5
BAB V KESIMPULAN DAN SARAN.....	14
DAFTAR PUSTAKA.....	15

BAB I. PENDAHULUAN

A. Latar Belakang

Industri gula merupakan salah satu industri bahan pokok di Indonesia. Keberadaannya sangat diperlukan dikarenakan menghasilkan gula yang dibutuhkan oleh masyarakat pada umumnya.

Dalam perkembangannya, industri gula dituntut untuk dapat beradaptasi dengan perkembangan teknologi saat ini agar dapat menghasilkan produk dengan efisiensi tinggi.

Salah satu perkembangan teknologi yang saat ini adalah simulasi proses. Simulasi proses dibutuhkan untuk dapat memprediksi proses industri sehingga ketidaknormalan dalam proses dapat ditangani sebelum terjadi agar dapat mengurangi cacat produk.

Salah satu simulasi proses yang dapat dimanfaatkan dalam industri gula adalah simulasi skema masakan dengan menggunakan program Jupyter Notebook dalam pemrograman Python Anaconda.

Program ini memiliki keunggulan dalam memproses dalam skema big data. Selain itu juga, program ini merupakan program freeware yang artinya dapat di unduh secara gratis melalui alamat situs www.python.org.

B. Tujuan Penelitian

1. Pembuatan kode program simulasi stasiun masakan menggunakan Python Notebook.
2. Mengetahui simulasi masakan skema ACD dengan program Jupyter notebook pada pemrograman Python Anaconda menggunakan kode program yang dibuat.

C. Manfaat Penelitian

Memberikan informasi mengenai skema masakan yang ideal sesuai hasil simulasi yang dapat memperbaiki kinerja stasiun masakan pada pabrik gula

D. Hipotesis

Kode program dan simulasi program dapat dilakukan dengan program Jupyter Notebook pada pemrograman Python Anaconda.

BAB II. TINJAUAN PUSTAKA

A. Python

Python adalah bahasa pemrograman berorientasi objek tingkat menengah yang mudah dipelajari dan digunakan namun cukup fleksibel untuk menangani berbagai tugas (Helmus & Collis, 2016). Sejak debut pertamanya pada tahun 1991 (Van Rossum & Drake Jr, 1995), sifat open-source-nya telah melambungkan popularitasnya, dan saat ini dianggap sebagai salah satu bahasa pemrograman terbaik untuk dipelajari (Saabith, Fareez, & Vinothraj, 2019).

Python merupakan program yang gratis untuk digunakan, kompatibel lintas platform (Mac, Windows, Linux, Ubuntu), dan memiliki persyaratan sistem yang rendah, memungkinkan siapa saja untuk mengembangkan dalam bahasa tersebut. Ini sudah memiliki komunitas pengguna yang besar, mulai dari orang biasa hingga pakar penelitian top, yang telah mengerjakan proyek menarik dalam ilmu data, pembelajaran mesin, kecerdasan buatan, pengembangan aplikasi dan game, penelitian ilmiah, dan bidang lainnya. Proyek-proyek ini mudah ditemukan karena komunitas sumber terbuka terus mengembangkan kemampuan bahasanya, dan dapat ditemukan hanya dengan mengetik "Python" ke mesin telusur mana pun.

Python menyediakan berbagai properti yang khusus khusus untuk disiplin ilmu psikologi, selain keunggulan yang tercantum di atas (Dalmaijer, 2016). Dari otomatisasi hingga analisis, dapat digunakan pada setiap tahap proses penelitian. Python sedang digunakan untuk membuat algoritme pembelajaran mesin canggih di bidang psikologi klinis dan ilmu saraf klinis untuk memahami awal, pemeliharaan, dan/atau remisi psikopatologi tergantung pada atribut pribadi dan keadaan sosiokultural (Coutanche & Hallion, 2019).

B. Anaconda

Anaconda adalah paket perangkat lunak gratis yang mencakup kotak peralatan yang dirancang khusus untuk penelitian dan sains. Saat menginstal Anaconda, memiliki akses ke berbagai lingkungan yang memungkinkan untuk mengembangkan dengan Python atau R.

Lingkungan ini, sering dikenal sebagai lingkungan pengembangan terintegrasi (IDE), adalah platform atau program yang membuat penulisan kode lebih mudah. Mereka menyediakan fungsi serupa untuk membuat teks sebagai pengolah teks seperti Microsoft Word, Google

Docs, dan Pages, tetapi lebih dari itu. Banyak kemampuan penting yang tersedia di IDE

untuk menulis, mengedit, dan men-debug kode, memvisualisasikan dan memeriksa data, menyimpan variabel, menyajikan hasil, dan berkolaborasi dalam proyek. Bahasa pemrograman tidak berbeda dari presentasi dan kekhasan IDE. Akibatnya, mengganti IDE Anaconda tidak menghasilkan perubahan signifikan pada kode Python. Memahami sintaks Python adalah bagian tersulit dari kurva belajar akan dapat dengan mudah mentransfer keterampilan pengkodean dari satu IDE ke IDE lainnya setelah menguasainya. Satu IDE tidak selalu lebih unggul dari yang lain; masing-masing memiliki kelebihan dan kekurangannya sendiri. Saat mengunduh toolkit akan mendapatkan akses ke perpustakaan besar fungsi bawaan yang dibuat oleh komunitas Python di masa lalu. Rutinitas ini digabungkan ke dalam pustaka, yang dapat diunduh dengan mudah menggunakan Anaconda. Ada berbagai cara untuk menginstal dan menggunakan Python di komputer, tetapi Anaconda adalah antarmuka pengguna grafis (GUI) langsung yang didukung dengan baik yang dilengkapi dengan pustaka paling signifikan dan lingkungan pengembangan terintegrasi (IDE) yang sudah diinstal sebelumnya. Anaconda juga membuat pemeliharaan semua perpustakaan ini tetap up to date menjadi jauh lebih mudah. Akibatnya, daripada menginstal Python satu per satu dengan berbagai IDE, pustaka, dan fungsi, Anaconda dapat menangani semuanya untuk sekaligus.

C. JUPYTER NOTEBOOK - JUPYTER LAB

Jupyter Notebook adalah program pengembangan interaktif berbasis web yang berjalan di browser default. Setiap blok kode dapat dilakukan secara independen, membuatnya sangat fleksibel dan mudah dimainkan. Ini memungkinkan penggunaan berbagai jenis teks dalam Notebook yang sama. Akibatnya, keluaran kode, visualisasi, persamaan, dan teks biasa semuanya dapat diakses di satu lokasi. Ini memudahkan untuk membuat dan mendistribusikan dokumen, serta menunjukkan kode dan temuan dengan cara yang logis dan menarik. Ini sangat ideal untuk kolaborasi karena berbasis web dan memudahkan untuk berbagi Notebook dengan orang lain. JupyterLab adalah plugin Notebook Jupyter dengan lebih banyak fungsi. Notebook Jupyter digabungkan dengan aplikasi tambahan di JupyterLab, seperti Terminal baris perintah, Konsol kode, dan Editor Teks.

BAB III. METODE PENELITIAN

A. Tempat dan Waktu Penelitian

Penelitian dilakukan di Politeknik LPP Yogyakarta selama bulan Januari - 25 Februari 2022

B. Bahan dan Alat

Laptop HP Probook 4420s dengan sistem operasi Windows 7 SP 2 dengan mengunduh program Python Anaconda3 - 2019.10 - Windows - X86_64

C. Metode Penelitian

1. Rancangan Percobaan

Penelitian ini dilakukan dengan melakukan simulasi pada rancangan pabrik gula berkapasitas 7500 TCD

2. Tata Laksana

a. Persiapan

Meng-*install*- program Python Anaconda 3 - 2019.10 pada laptop.

b. Pelaksanaan penelitian

Membuat kode program simulasi dan melakukan simulasi dengan kode program yang dibuat.

D. Variabel Pengamatan

1. Kode programan simulasi masakan
2. Nilai operasional proses stasiun masakan

E. Analisis data

Menggunakan program simulasi yang dilakukan menggunakan Jupyter Notebook

BAB IV. HASIL DAN PEMBAHASAN

A. Kode pemrograman

```
import pandas as pd
import numpy as np
from IPython.display import display, HTML
pd.set_option('display.notebook_repr_html', True)
def _repr_latex_(self):
    return "\centering{%s}" % self.to_latex()
    #return "\raggedright{%s}" % self.to_latex()
pd.DataFrame._repr_latex_ = _repr_latex_
# monkey patch pandas DataFrame
def cobenze(a,b,c,d,e): #Fungsi cobenze, a = HK atau brix tertinggi, b
# c = HK atau brix terendah, d = tonbrix atau t
    if e == 1 :
        x = ((b-c)/(a-c))*d
    elif e == 2 :
        x = ((a-c)/(b-c))*d
    elif e == 3 :
        x = ((a-b)/(a-c))*d
    elif e == 4 :
        x = ((a-c)/(a-b))*d
    else :
        print('Fungsi tidak ada !')
    return x
#Data Analisa bahan
ton_tebu = 7500
NM_percent_tebu = 105
brix_NM = 10.5
pol_blotong = 3
HK_ntap = 72
retensi = 90
```

mso_percent_NM = 0.6
bagacillo_rasio = 0.5
hw_percent_blotong = 110
mso_percent_blotong = 20
so_percent_mud = 7.5
brix_ntap = 8
brix_NKS = 65
HK_NKS = 72
HK_GKP = 99.9
HK_gula_C = 93
HK_gula_D = 88
HK_gula_D2 = 91
brix_msk_A = 94
brix_msk_C = 96
brix_msk_D = 98
brix_mol_A = 76
brix_mol_C = 76
brix_gula_C = 99
brix_klare_D = 76
brix_tetes = 88
kk_a = 51
kk_c = 50
kk_d = 50
kk_d2 = 85
iu_NKS = 7000
iu_gula_C = 3000
iu_gula_D2 = 4500

#Variable awal

tonbrix_ntap = 0
tonbrix_gula_C = 0
tonbrix_gula_D2 = 0

tonbrix_NKS_msk_C = 0

tonbrix_NKS_msk_D = 0

tonbrix_klare_D = 0

HK_klare_D = 0

#Loop RUN

Iterasi run=0

while(run<500):

#Nira Mentah

ton_NM = NM_percent_tebu*ton_tebu/100

tonbrix_NM = ton_NM*brix_NM/100

tonbrix_NM_mix = tonbrix_NM + tonbrix_ntap

ton_padatan_blotong = ton_NM*mso_percent_NM/100

ton_bagacillo = ton_padatan_blotong*bagacillo_rasio/(retensi/100)

ton_total_padatan_blotong = ton_bagacillo+ton_padatan_blotong

ton_blotong = ton_total_padatan_blotong/(mso_percent_blotong/100)

brix_blotong = (HK_ntap/pol_blotong)/100

tonbrix_blotong = ton_blotong*brix_blotong/100

ton_padatan_mud = ton_padatan_blotong/retensi*100

ton_total_padatan_mud = ton_padatan_mud+ton_bagacillo

ton_mud = ton_total_padatan_mud/so_percent_mud*100

ton_hw = ton_blotong*hw_percent_blotong/100

ton_ntap = ton_mud+ton_hw-ton_blotong

tonbrix_ntap = ton_ntap*brix_ntap/100

#Nira Kental Sulfitir

tonbrix_NKS = tonbrix_NM_mix-tonbrix_blotong-tonbrix_ntap

ton_NKS = tonbrix_NKS/brix_NKS*100

#Masakan A

tonbrix_NKS_msk_A = tonbrix_NKS-tonbrix_NKS_msk_C-tonbrix_NKS_msk_D

```

tonbrix_msk_A = tonbrix_NKS_msk_A+tonbrix_gula_C+tonbrix_gula_D2
tonbrix_GKP = tonbrix_msk_A*kk_a/100
tonbrix_mol_A = tonbrix_msk_A-tonbrix_GKP
ton_msk_A =
tonbrix_msk_A/brix_msk_A*100 ton_mol_A
= tonbrix_mol_A/brix_mol_A*100
HK_msk_A = ((tonbrix_NKS_msk_A*HK_NKS)+(tonbrix_gula_C*HK_gula_C)
+ (tonbrix_gula_D2*HK_gula_D2)) /tonbrix_msk_A
HK_mol_A = ((tonbrix_msk_A*HK_msk_A) -
(tonbrix_GKP*HK_GKP))/tonbrix_mol_A
iu_msk_A = ((tonbrix_NKS_msk_A*iu_NKS)+(tonbrix_gula_C*iu_gula_C)+
(tonbrix_gula_D2*iu_gula_D2))/tonbrix_msk_A
iu_GKP = iu_msk_A/21
on_GKP = tonbrix_GKP/99.995*100

```

#Masakan C

```

tonbrix_NKS_msk_C = 0
porsi_mol_A_ke_msk_C = 0
tonbrix_mol_A_msk_C = tonbrix_mol_A*porsi_mol_A_ke_msk_C
tonbrix_msk_C = tonbrix_mol_A_msk_C +
tonbrix_NKS_msk_C
tonbrix_gula_C = tonbrix_msk_C*kk_c/100
tonbrix_mol_C = tonbrix_msk_C-tonbrix_gula_C

```

if

```

tonbrix_msk_C == 0:
HK_msk_C=0
HK_mol_C=0
ton_msk_C=0
ton_mol_C=0
ton_gula_C=0
HK_gula_C = 0

```

iu_gula_C = 0

brix_gula_C=0

brix_msk_C=0

brix_mol_C=0

else:

ton_msk_C =

tonbrix_msk_C/brix_msk_C*100 ton_mol_C

= tonbrix_mol_C/brix_mol_C*100

HK_msk_C = ((tonbrix_mol_A_msk_C*HK_mol_A) +

(tonbrix_NKS_msk_C*H/))tonbrix_msk_C

HK_mol_C = ((tonbrix_msk_C*HK_msk_C)-(tonbrix_gula_C*HK_gula_C))
/tonbrix_mol_C

ton_gula_C = tonbrix_gula_C/99.5*100

#Masakan D

tonbrix_NKS_msk_D = 60

tonbrix_mol_A_msk_D = tonbrix_mol_A-tonbrix_mol_A_msk_C

tonbrix_msk_D = tonbrix_mol_C + tonbrix_mol_A_msk_D +
tonbrix_NKS_msk_D + tonbrix_klare_D

tonbrix_gula_D = tonbrix_msk_D*kk_d/100

tonbrix_tetes = tonbrix_msk_D-tonbrix_gula_D

ton_msk_D =

tonbrix_msk_D/brix_msk_D*100

ton_tetes = tonbrix_tetes/brix_tetes*100

HK_msk_D =

((tonbrix_mol_C*HK_mol_C)+(tonbrix_mol_A_msk_D*HK_mol_A) +
(tonbrix_NKS_msk_D*HK_NKS) +

(tonbrix_klare_D*HK_klare_D/tonbrix_msk_D

HK_tetes = ((tonbrix_msk_D*HK_msk_D) -

(tonbrix_gula_D*HK_gula_D))/tonbrix_tetes

ton_gula_D = tonbrix_gula_D/90*100

tonbrix_gula_D2 = tonbrix_gula_D*kk_d2/100

tonbrix_klare_D = tonbrix_gula_D-tonbrix_gula_D2

```

ton_gula_D2 = tonbrix_gula_D2/99.5*100
ton_klare_D = tonbrix_klare_D/brix_klare_D*100
HK_klare_D = ((tonbrix_gula_D*HK_gula_D) -
(tonbrix_gula_D2*HK_gula_D/tonbrix_klare_D

```

run+=1

else:

```

print("") print('>>',run,'Iterasi')
rend =
ton_GKP/ton_tebu*100
print("")
print('Kap. Giling =','{:,.1f}'.format(ton_tebu))
print('Ton GKP      =','{:,.1f}'.format(ton_GKP))
print('GKP % Tebu   =','{:,.2f}'.format(rend))
print("")
print('%brix NM =','{:,.2f}'.format(brix_NM))
print('HK NKS      =','{:,.2f}'.format(HK_NKS))
print('% Retensi =','{:,.2f}'.format(retensi))
print('Nira Tapis % NM      =','{:,.2f}'.format(ton_ntap/ton_NM*100))
print('Blotong % Tebu      =','{:,.1f}'.format(ton_blotong/ton_tebu*1 data2
= [['Nira Kental',ton_NKS,tonbrix_NKS,brix_NKS,HK_NKS,iu_NKS]
    ['Nira Kental ke Msk A',"tonbrix_NKS_msk_A,brix_NKS,HK_N
    ['Magma Gula C ke Msk A',"tonbrix_gula_C","HK_gula_C,iu
    ['Leburan Gula D2 ke Msk A',"tonbrix_gula_D2","HK_gula_ ['<<
    Masakan A >>',"ton_msk_A,tonbrix_msk_A,brix_msk_A,HK_ms
    ['GKP',ton_GKP,tonbrix_GKP,99.995,HK_GKP,iu_GKP],
    ['Mol A',ton_mol_A,tonbrix_mol_A,brix_mol_A,HK_mol_A,"], ['Mol
    A ke Msk C',"tonbrix_mol_A_msk_C,brix_mol_A,HK_mol ['Nira
    Kental ke Msk C',"tonbrix_NKS_msk_C,brix_NKS,HK_N
    ['<< Masakan
    C >>',"ton_msk_C,tonbrix_msk_C,brix_msk_C,HK_ms['Gula
    C',ton_gula_C,tonbrix_gula_C,brix_gula_C,HK_gula_C,i ['Mol
    C',ton_mol_C,tonbrix_mol_C,brix_mol_C,HK_mol_C,"],

```

```

['Nira Kental ke Msk D',"tonbrix_NKS_msk_D,brix_NKS,HK_N
['Mol A ke Msk D',"tonbrix_mol_A_msk_D,brix_mol_A,HK_mol
['Klare D ke Msk D',ton_klare_D,tonbrix_klare_D,brix_klare ['<<
Masakan D >>',ton_msk_D,tonbrix_msk_D,brix_msk_D,HK_ms
['Gula D',ton_gula_D,tonbrix_gula_D,90,HK_gula_D,"], ['Gula
D2',ton_gula_D2,tonbrix_gula_D2,99.5,HK_gula_D2,iu_g ['Klare
D',ton_klare_D,tonbrix_klare_D,brix_klare_D,HK_klar
['Tetes',ton_tetes,tonbrix_tetes,brix_tetes,HK_tetes,"]]
df2 = pd.DataFrame(data2,columns=['Nama Bahan','Ton','Tonbrix','%br
pd.options.display.float_format = '{:,.2f}'.format
f2.style.set_properties(**{'text-align': 'left'})
df2.style.set_table_styles([ dict(selector='th', props=[('text-align', 'center')])
display(df2)

msk_A_tebu=ton_msk_A/ton_tebu*100
msk_C_tebu=ton_msk_C/ton_tebu*100
msk_D_tebu=ton_msk_D/ton_tebu*100
tetes_tebu = ton_tetes/ton_tebu*100
print('Msk A % Tebu =', '{:,.1f}'.format(msk_A_tebu))
print('Msk C % Tebu =', '{:,.1f}'.format(msk_C_tebu))
print('Msk D % Tebu =', '{:,.1f}'.format(msk_D_tebu))
print('Tetes % Tebu =', '{:,.1f}'.format(tetes_tebu))
%run ./simpan_pict_acd_r1.ipynb

```

B. HASIL SIMULASI

	Nama Bahan	Ton	Tonbrix	%brix	HK	ICUMSA
0	Nira Kental	1,270.76	825.99	65	72.00	7000
1	Nira Kental ke Msk A		765.99	65	72.00	7000
2	Magma Gula C ke Msk A		0.00		0.00	0
3	Leburan Gula D2 ke Msk A		258.13		91.00	4500
4	<< Masakan A >>	1,089.50	1,024.13	94	76.79	6,369.87

5	GKP	522.33	522.31	100.00	99.90	303.33
6	Mol A	660.29	501.82	76	52.73	
7	Mol A ke Msk C		0.00	76	52.73	
8	Nira Kental ke Msk C		0.00	65	72.00	7000
9	<< Masakan C >>	0	0.00	0	0.00	
10	Gula C	0	0.00	0	0.00	0
11	Mol C	0	0.00	0	0.00	
12	Nira Kental ke Msk D		60.00	65	72.00	7000
13	Mol A ke Msk D		501.82	76	52.73	
14	Klare D ke Msk D	59.94	45.55	76	71.00	
15	<< Masakan D >>	619.77	607.38	98	56.01	
16	Gula D	337.43	303.69	90	88.00	
17	Gula D2	259.43	258.13	99.50	91.00	4500
18	Klare D	59.94	45.55	76	71.00	
19	Tetes	345.10	303.69	88	24.02	

>> 500 Iterasi

Kap. Giling = 7,500.0

Ton GKP = 522.3

GKP % Tebu = 6.96

%brix NM = 10.50

HK NKS = 72.00

% Retensi = 90.00

Nira Tapis % NM = 13.80

Blotong % Tebu = 4.9

Msk A % Tebu = 14.5

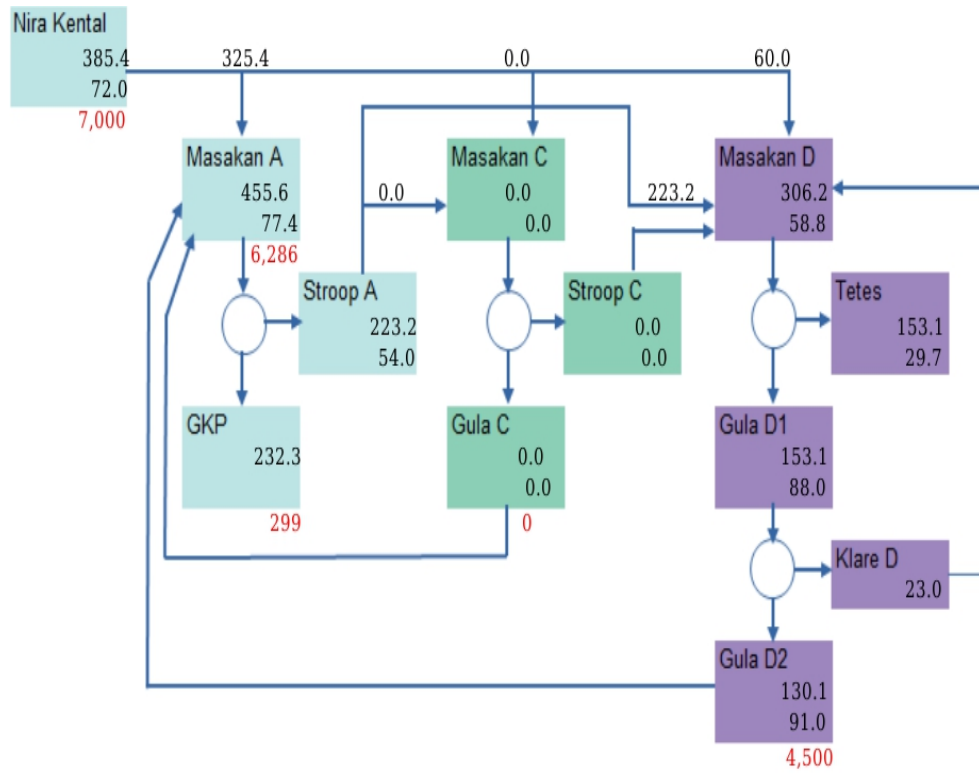
Msk C % Tebu = 0.0

Msk D % Tebu = 8.3

Tetes % Tebu = 4.6

C. TAMPILAN GRAFIS

In [3]: Image1(filename = "acd_hasil1_r1.png", width=900, height=600)



BAB V. KESIMPULAN DAN SARAN

A. Kesimpulan

1. Kode program di atas dapat digunakan untuk melakukan simulasi pemrograman stasiun masakan pabrik gula dengan sistem ACD

B. Saran

Perlu dilakukan penelitian lanjutan untuk melakukan simulasi dengan berbagai variasi data

DAFTAR PUSTAKA

- Helmus, J. J., & Collis, S. M. (2016). The python arm radar toolkit (py-art), a library for working with weather radar data in the python programming language. *Journal of Open Research Software*, 4.
- Saabith, A. S., Fareez, M., & Vinothraj, T. (2019). Python current trend applications-an overview. *International Journal of Advance Engineering and Research Development*, 6(10).
- Van Rossum, G., & Drake Jr, F. (1995). Python tutorial (vol.620). *CWI Report CS-R9526, Amsterdam, Netherlands, mseke. karlin. mff. cuni. cz/~halas/IT/tutorial. pdf*.
- Knorr, F. G., Marxen, M., & Petzold, J. (2019). Pyparadigm-a python library to build screens in a declarative way. *Frontiers in neuroinformatics*, 13, 59.