



YAYASAN SASMITA JAYA

UNIVERSITAS PAMULANG

Jl. Surya Kencana No.1 Pamulang Barat, Pamulang, Tangerang Selatan – Banten, Kode Pos: 15417.

Telp./Fax. (021) 7412566, website: <http://unpam.ac.id/>

Nama Mahasiswa	: Sundapa	Mata Kuliah	: Kecerdasan Buatan
Nomer induk Mahasiswa	: 191011400208	Nama Dosen	:
Semester	: 6 (enam)	Nilai	:
Jurusan	: Teknik Informatika	Kode Kelas	: 06TPLM002

LEMBAR JAWABAN

```
def down(x, xmin, xmax):
    return (xmax - x) / (xmax - xmin)

def up(x, xmin, xmax):
    return (x - xmin) / (xmax - xmin)

class Permintaan():
    minimum = 2100
    maximum = 3500

    def turun(self, x):
        if x >= self.maximum:
            return 0
        elif x <= self.minimum:
            return 1
        else:
            return down(x, self.minimum, self.maximum)

    def naik(self, x):
        if x >= self.maximum:
            return 1
        elif x <= self.minimum:
            return 0
        else:
            return up(x, self.minimum, self.maximum)

class Persediaan():
    minimum = 100
    maximum = 250

    def sedikit(self, x):
        if x >= self.maximum:
            return 0
        elif x <= self.minimum:
            return 1
        else:
            return down(x, self.minimum, self.maximum)
```

```

def banyak(self, x):
    if x >= self.maximum:
        return 1
    elif x <= self.minimum:
        return 0
    else:
        return up(x, self.minimum, self.maximum)

class Produksi():
    minimum = 1000
    maximum = 5000
    permintaan = 0
    persediaan = 0

    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

    def _bertambah(self, a):
        return a*(self.maximum - self.minimum) + self.minimum

    def _inferensi(self, pmt=Permintaan(), psd=Persediaan()):
        result = []
        # [R1] JIKA Permintaan TURUN, dan Persediaan BANYAK, MAKA
        # Produksi Barang BERKURANG.
        a1 = min(pmt.turun(self.permintaan), psd.banyak(self.persediaan))
        z1 = self._berkurang(a1)
        result.append((a1, z1))
        # [R2] JIKA Permintaan TURUN, dan Persediaan SEDIKIT, MAKA
        # Produksi Barang BERKURANG.
        a2 = min(pmt.turun(self.permintaan), psd.sedikit(self.persediaan))
        z2 = self._berkurang(a2)
        result.append((a2, z2))
        # [R3] JIKA Permintaan NAIK, dan Persediaan BANYAK, MAKA
        # Produksi Barang BERTAMBAH.
        a3 = min(pmt.naik(self.permintaan), psd.banyak(self.persediaan))
        z3 = self._bertambah(a3)
        result.append((a3, z3))
        # [R4] JIKA Permintaan NAIK, dan Persediaan SEDIKIT, MAKA
        # Produksi Barang BERTAMBAH.
        a4 = min(pmt.naik(self.permintaan), psd.sedikit(self.persediaan))
        z4 = self._bertambah(a4)
        result.append((a4, z4))
        return result

    def defuzifikasi(self, data_inferensi=[]):
        #  $(\alpha_1 * z_1 + \alpha_2 * z_2 + \alpha_3 * z_3 + \alpha_4 * z_4) / (\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4)$ 
        data_inferensi = data_inferensi if data_inferensi else self._inferensi()
        res_a_z = 0
        res_a = 0
        for data in data_inferensi:
            # data[0] = a

```

```

        # data[1] = z
        res_a_z += data[0] * data[1]
        res_a += data[0]
    return res_a_z/res_a

```

```

def turun(x, xmin, xmax):
    return (xmax - x) / (xmax - xmin)

def naik(x, xmin, xmax):
    return (x - xmin) / (xmax - xmin)

class PermintaanBaru():
    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

    def _bertambah(self, a):
        return a*(self.maximum - self.minimum) + self.minimum

    def turun(self, x):
        if x >= self.median:
            return 0
        elif x <= self.minimum:
            return 1
        else:
            return turun(x, self.minimum, self.median)

    def naik(self, x):
        if x >= self.maximum:
            return 1
        elif x <= self.median:
            return 0
        else:
            return naik(x, self.median, self.maximum)

    def tetap(self, x):
        if x >= self.maximum or x <= self.minimum:
            return 0
        elif self.minimum < x < self.median:
            return naik(x, self.minimum, self.median)
        elif self.median < x < self.maximum:
            return turun(x, self.median, self.maximum)
        else:
            return 1

# class permintaan lama
class PermintaanLama():
    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

```

```

def _bertambah(self, a):
    return a*(self.maximum - self.minimum) + self.minimum

def turun(self, x):
    if x >= self.maximum:
        return 0
    elif x<= self.minimum:
        return 1
    else:
        return turun(x, self.minimum, self.maximum)

def naik(self, x):
    if x >= self.maximum:
        return 1
    elif x<= self.minimum:
        return 0
    else:
        return naik(x, self.minimum, self.maximum)

def tetap(self, x):
    if x >= self.maximum or x<= self.minimum:
        return 0
    elif self.minimum < x < self.maximum:
        return 1
    else:
        return 0

# create fuzzy with 3 variables and 10 rules in class Persediaan() and define the rules
class PersediaanBaru():
    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

    def _bertambah(self, a):
        return a*(self.maximum - self.minimum) + self.minimum

    def sedikit(self, x):
        if x >= self.median:
            return 0
        elif x<= self.minimum:
            return 1
        else:
            return turun(x, self.minimum, self.median)

    def banyak(self, x):
        if x >= self.maximum:
            return 1
        elif x<= self.median:
            return 0
        else:
            return naik(x, self.median, self.maximum)

```

```

def tetap(self, x):
    if x >= self.maximum or x <= self.minimum:
        return 0
    elif self.minimum < x < self.median:
        return naik(x, self.minimum, self.median)
    elif self.median < x < self.maximum:
        return turun(x, self.median, self.maximum)
    else:
        return 1

# class persediaan lama
class PersediaanLama():
    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

    def _bertambah(self, a):
        return a*(self.maximum - self.minimum) + self.minimum

    def sedikit(self, x):
        if x >= self.maximum:
            return 0
        elif x <= self.minimum:
            return 1
        else:
            return turun(x, self.minimum, self.maximum)

    def banyak(self, x):
        if x >= self.maximum:
            return 1
        elif x <= self.minimum:
            return 0
        else:
            return naik(x, self.minimum, self.maximum)

    def tetap(self, x):
        if x >= self.maximum or x <= self.minimum:
            return 0
        elif self.minimum < x < self.maximum:
            return naik(x, self.minimum, self.maximum)
        else:
            return 1

class ProduksiBaru():
    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

    def _bertambah(self, a):
        return a*(self.maximum - self.minimum) + self.minimum

    def interensi(self, x):
        return self.tetap(x) * self.turun(x) * self.naik(x)

```

```

def turun(self, x):
    if x >= self.median:
        return 0
    elif x <= self.minimum:
        return 1
    else:
        return turun(x, self.minimum, self.median)

def naik(self, x):
    if x >= self.maximum:
        return 1
    elif x <= self.median:
        return 0
    else:
        return naik(x, self.median, self.maximum)

def tetap(self, x):
    if x >= self.maximum or x <= self.minimum:
        return 0
    elif self.minimum < x < self.median:
        return naik(x, self.minimum, self.median)
    elif self.median < x < self.maximum:
        return turun(x, self.median, self.maximum)
    else:
        return 1

# new class produk lama
class ProdukLama():
    def _berkurang(self, a):
        return self.maximum - a*(self.maximum - self.minimum)

    def _bertambah(self, a):
        return a*(self.maximum - self.minimum) + self.minimum

    def interensi(self, x):
        return self.tetap(x) * self.turun(x) * self.naik(x)

    def turun(self, x):
        if x >= self.median:
            return 0
        elif x <= self.minimum:
            return 1
        else:
            return turun(x, self.minimum, self.median)

    def naik(self, x):
        if x >= self.maximum:
            return 1
        elif x <= self.median:
            return 0
        else:
            return naik(x, self.median, self.maximum)

```

```
def tetap(self, x):
    if x >= self.maximum or x<= self.minimum:
        return 0
    elif self.minimum < x < self.median:
        return naik(x, self.minimum, self.median)
    elif self.median < x < self.maximum:
        return turun(x, self.median, self.maximum)
    else:
        return 1

# def interferensi self pmt psd pdt

def interferensi(self, pmt, psd, pdt):
    return self.tetap(pmt) * self.turun(pmt) * self.naik(pmt) * self.tetap(psd) *
self.turun(psd) * self.naik(psd) * self.tetap(pdt) * self.turun(pdt) * self.naik(pdt)
```



UNIVERSITAS PAMULANG
KARTU UJIAN AKHIR SEMESTER GENAP 2021/2022
NOMOR UJIAN : 017008207281

FAKULTAS / PRODI : TEKNIK / TEKNIK INFORMATIKA S1

NAMA MAHASISWA: SUNDAPA

NIM : 191011400208

SHIFT : REGULER B

No	Hari/ Tanggal	Waktu	Ruang	Kelas	Mata Kuliah	Paraf
1				06TPLM002	KOMPUTER GRAFIK 1	1
2				06TPLM002	PEMROGRAMAN WEB 2	2
3				06TPLM002	REKAYASA PERANGKAT LUNAK	3
4				06TPLM002	SISTEM INFORMASI MANAJEMEN	4
5				06TPLM002	KECERDASAN BUATAN	5
6				06TPLM002	TEKNIK KOMPILASI	6
7				06TPLM002	KERJA PRAKTEK	7
8				06TPLM002	MOBILE PROGRAMMING	8

Peraturan dan Tata Tertib Peserta Ujian

1. Peserta ujian harus berpakaian rapi, sopan dan memakai jaket Almamater
2. Peserta ujian sudah berada di ruangan sepuluh menit sebelum ujian dimulai
3. Peserta ujian yang terlambat diperkenankan mengikuti ujian setelah mendapat ijin, tanpa perpanjangan waktu
4. Peserta ujian hanya diperkenankan membawa alat-alat yang ditentukan oleh panitia ujian
5. Peserta ujian dilarang membantu teman, mencontoh dari teman dan tindakan-tindakan lainnya yang mengganggu peserta ujian lain
6. Peserta ujian yang melanggar tata tertib ujian dikenakan sanksi akademik



Tangerang Selatan, 4 Juli 2022
Ketua Panitia Ujian

UBAID AL FARUQ, S.Pd., M. Pd
NIDN. 0418028702



UNIVERSITAS PAMULANG
DATA PEMBAYARAN SEMESTER GENAP 2021/2022

FAKULTAS / PRODI : TEKNIK / TEKNIK INFORMATIKA S1

NAMA MAHASISWA : SUNDAPA

NIM : 191011400208

SHIFT : REGULER B

DATA PEMBAYARAN TAGIHAN UANG KULIAH

NO	NOMOR TAGIHAN	NO URUT	PEMBAYARAN	JML BAYAR	STATUS BAYAR	TGL BAYAR	CHANNEL	TEMPAT BAYAR
1	2120115652102201	1	REGISTRASI	400000	LUNAS	2022-02-11 00:00:00.000000	NEWMOBILE	Bank BNI
2	2120115652102301	2	ANGSURAN KE-2	200000	LUNAS	2022-04-11 16:57:58.311000	IBANKING	Bank MANDIRI
3	2120115652102401	3	ANGSURAN KE-3	200000	LUNAS	2022-04-11 16:58:50.957000	IBANKING	Bank MANDIRI
4	2120115652100501	4	UTS	250000	LUNAS	2022-04-11 16:59:33.770000	IBANKING	Bank MANDIRI
5	2120115652102501	5	ANGSURAN KE-4	200000	LUNAS	2022-06-27 00:00:00.000000	NEWMOBILE	Bank BNI
6	2120115652102601	6	ANGSURAN KE-5	200000	LUNAS	2022-06-27 00:00:00.000000	NEWMOBILE	Bank BNI
7	2120115652102701	7	ANGSURAN KE-6	200000	LUNAS	2022-06-27 00:00:00.000000	NEWMOBILE	Bank BNI
8	2120115652100401	8	PRAKTEK	100000	LUNAS	2022-06-27 00:00:00.000000	NEWMOBILE	Bank BNI
9	2120115652100601	9	UAS	250000	LUNAS	2022-06-27 00:00:00.000000	NEWMOBILE	Bank BNI

DATA PEMBAYARAN TAGIHAN LAINNYA

NO	NOMOR TAGIHAN	NO URUT	PEMBAYARAN	JML BAYAR	STATUS BAYAR	TGL BAYAR	CHANNEL	TEMPAT BAYAR
----	---------------	---------	------------	-----------	--------------	-----------	---------	--------------