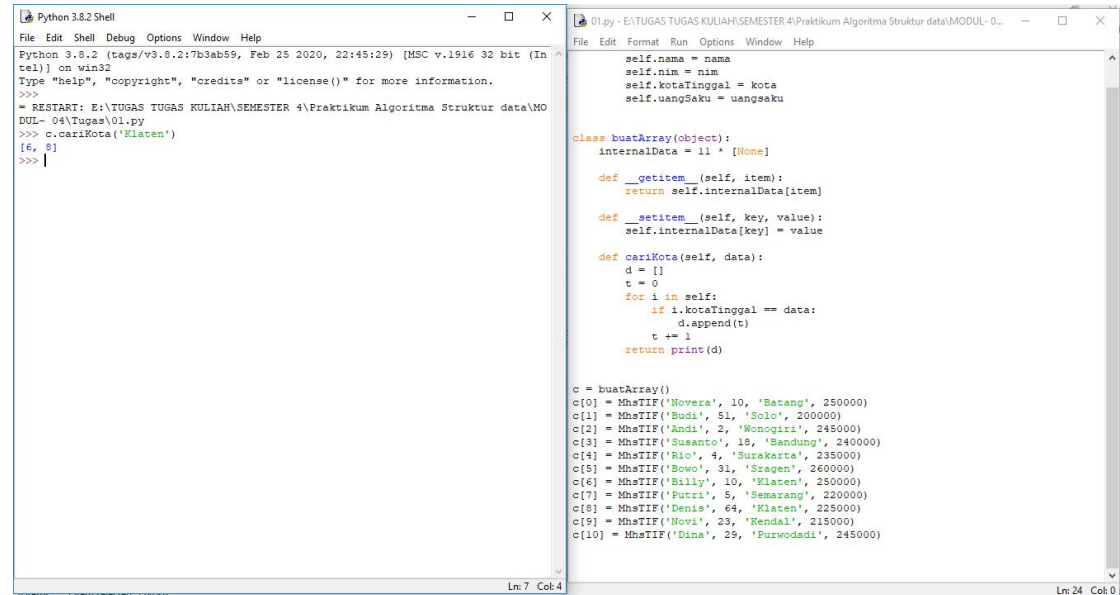


NAMA : NOVERA DYAH A.
NIM : L200180026
KELAS : B

TUGAS MODUL 4

1.



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL-04\Tugas\01.py
>>> c.cariKota('Klaten')
[6, 8]
>>> |

01.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL-04\Tugas\01.py
File Edit Format Run Options Window Help
self.nama = nama
self.nim = nim
self.kotaTinggal = kota
self.uangSaku = uangsaku

class buatArray(object):
    internalData = 11 * [None]

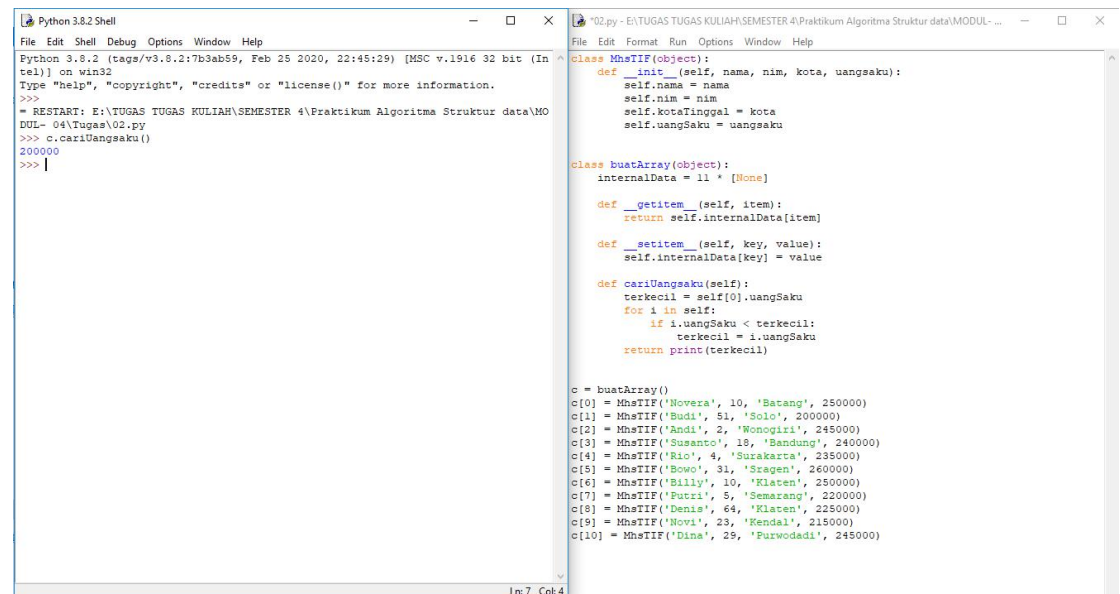
    def __getitem__(self, item):
        return self.internalData[item]

    def __setitem__(self, key, value):
        self.internalData[key] = value

    def cariKota(self, data):
        d = []
        t = 0
        for i in self:
            if i.kotaTinggal == data:
                d.append(t)
                t += 1
        return print(d)

c = buatArray()
c[0] = MhsTIF('Novera', 10, 'Batang', 250000)
c[1] = MhsTIF('Budi', 51, 'Solo', 200000)
c[2] = MhsTIF('Andi', 2, 'Wonogiri', 245000)
c[3] = MhsTIF('Susanto', 18, 'Bandung', 240000)
c[4] = MhsTIF('Rio', 4, 'Surakarta', 235000)
c[5] = MhsTIF('Bowo', 31, 'Sragen', 260000)
c[6] = MhsTIF('Billy', 10, 'Klaten', 250000)
c[7] = MhsTIF('Putri', 5, 'Semarang', 220000)
c[8] = MhsTIF('Dennis', 64, 'Klaten', 225000)
c[9] = MhsTIF('Novi', 23, 'Kendal', 215000)
c[10] = MhsTIF('Dina', 29, 'Purwodadi', 245000)
```

2.



```
Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL-04\Tugas\02.py
>>> c.cariUangSaku()
200000
>>> |

02.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL-04\Tugas\02.py
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

class buatArray(object):
    internalData = 11 * [None]

    def __getitem__(self, item):
        return self.internalData[item]

    def __setitem__(self, key, value):
        self.internalData[key] = value

    def cariUangSaku(self):
        terkecil = self[0].uangSaku
        for i in self:
            if i.uangSaku < terkecil:
                terkecil = i.uangSaku
        return print(terkecil)

c = buatArray()
c[0] = MhsTIF('Novera', 10, 'Batang', 250000)
c[1] = MhsTIF('Budi', 51, 'Solo', 200000)
c[2] = MhsTIF('Andi', 2, 'Wonogiri', 245000)
c[3] = MhsTIF('Susanto', 18, 'Bandung', 240000)
c[4] = MhsTIF('Rio', 4, 'Surakarta', 235000)
c[5] = MhsTIF('Bowo', 31, 'Sragen', 260000)
c[6] = MhsTIF('Billy', 10, 'Klaten', 250000)
c[7] = MhsTIF('Putri', 5, 'Semarang', 220000)
c[8] = MhsTIF('Dennis', 64, 'Klaten', 225000)
c[9] = MhsTIF('Novi', 23, 'Kendal', 215000)
c[10] = MhsTIF('Dina', 29, 'Purwodadi', 245000)
```

3.

```
03.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL-0... Python 3.8.2 Shell
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

class buatArray(object):
    internalData = 11 * [None]

    def __getitem__(self, item):
        return self.internalData[item]

    def __setitem__(self, key, value):
        self.internalData[key] = value

    def siapaTerkecil(self):
        terkecil = self[0].uangSaku
        d = []
        for i in self:
            if i.uangSaku <= terkecil:
                terkecil = i.uangSaku
        for i in self:
            if terkecil == i.uangSaku:
                d.append((i.nama, i.nim, i.kotaTinggal, i.uangSaku))
        return d

c = buatArray()
c[0] = MhsTIF('Novera', 10, 'Batang', 250000)
c[1] = MhsTIF('Budi', 51, 'Solo', 200000)
c[2] = MhsTIF('Andi', 2, 'Wonogiri', 245000)
c[3] = MhsTIF('Susanto', 18, 'Bandung', 240000)
c[4] = MhsTIF('Rio', 4, 'Surakarta', 235000)
c[5] = MhsTIF('Bowo', 31, 'Sragen', 260000)
c[6] = MhsTIF('Billy', 10, 'Klaten', 250000)
c[7] = MhsTIF('Putri', 5, 'Semarang', 220000)
c[8] = MhsTIF('Denis', 64, 'Klaten', 225000)
c[9] = MhsTIF('Novi', 23, 'Kendal', 215000)
c[10] = MhsTIF('Dina', 29, 'Purwodadi', 245000)

Ln: 11 Col: 0 Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MO
DUL- 04\Tugas\03.py
>>> c.siapaTerkecil()
[('Budi', 51, 'Solo', 200000)]
>>>
```

4.

```
04.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL-0... Python 3.8.2 Shell
File Edit Format Run Options Window Help
class MhsTIF(object):
    def __init__(self, nama, nim, kota, uangsaku):
        self.nama = nama
        self.nim = nim
        self.kotaTinggal = kota
        self.uangSaku = uangsaku

class buatArray(object):
    internalData = 11 * [None]

    def __getitem__(self, item):
        return self.internalData[item]

    def __setitem__(self, key, value):
        self.internalData[key] = value

    def kurang250ribu(self):
        d = []
        for i in self:
            if i.uangSaku < 250000:
                d.append((i.nama, i.nim, i.kotaTinggal, i.uangSaku))
        return d

c = buatArray()
c[0] = MhsTIF('Novera', 10, 'Batang', 250000)
c[1] = MhsTIF('Budi', 51, 'Solo', 200000)
c[2] = MhsTIF('Andi', 2, 'Wonogiri', 245000)
c[3] = MhsTIF('Susanto', 18, 'Bandung', 240000)
c[4] = MhsTIF('Rio', 4, 'Surakarta', 235000)
c[5] = MhsTIF('Bowo', 31, 'Sragen', 260000)
c[6] = MhsTIF('Billy', 10, 'Klaten', 250000)
c[7] = MhsTIF('Putri', 5, 'Semarang', 220000)
c[8] = MhsTIF('Denis', 64, 'Klaten', 225000)
c[9] = MhsTIF('Novi', 23, 'Kendal', 215000)
c[10] = MhsTIF('Dina', 29, 'Purwodadi', 245000)

Ln: 18 Col: 19 Python 3.8.2 Shell
File Edit Shell Debug Options Window Help
Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MO
DUL- 04\Tugas\04.py
>>> c.kurang250ribu()
[('Budi', 51, 'Solo', 200000), ('Andi', 2, 'Wonogiri', 245000), ('Susanto', 18, 'Bandung', 240000), ('Rio', 4, 'Surakarta', 235000), ('Putri', 5, 'Semarang', 220000), ('Denis', 64, 'Klaten', 225000), ('Novi', 23, 'Kendal', 215000), ('Dina', 29, 'Purwodadi', 245000)]
>>>
```

5.

*05.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- ...
File Edit Format Run Options Window Help

```

class node(object):
    def __init__(self, data, next = None):
        self.data = data
        self.next = next

    def cariLinkedList(self, dicari):
        curNode = self
        while curNode is not None:
            if curNode.next is not None:
                if curNode.data != dicari:
                    curNode = curNode.next
            else:
                print ("Data ", dicari, "ada dalam linked list")
                break
            elif curNode.next is None:
                print ("Data ", dicari, "tidak ada dalam linked list")
                break

```

Ln: 19 Col: 0

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- 04\Tugas\05.py
>>> a = node(12)
>>> menu = a
>>> a.next = node(34)
>>> a = a.next
>>> a.next = node(10)
>>> a = a.next
>>> a.next = node(45)
>>> menu.cariLinkedList(10)
Data 10 ada dalam linked list
>>> menu.cariLinkedList(110)
Data 110 tidak ada dalam linked list
>>>

```

Ln: 16 Col: 0

6.

*06.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- ...
File Edit Format Run Options Window Help

```

def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1

    while low <= high:
        mid = (high + low) // 2
        if kumpulan[mid] == target:
            return "target berada di index " + str(mid)
            break
        elif target < kumpulan[mid]:
            high = mid - 1
        else:
            low = mid + 1
    return False

```

Ln: 18 Col: 0

Python 3.8.2 Shell
File Edit Shell Debug Options Window Help

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- 04\Tugas\06.py
>>> daftar = [10, 25, 30, 45, 75, 80, 100]
>>> binSe(daftar, 50)
False
>>> binSe(daftar, 95)
False
>>> binSe(daftar, 75)
'target berada di index 4'
>>> binSe(daftar, 25)
'target berada di index 1'
>>>

```

Ln: 14 Col: 4

7.

07.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- ...
Python 3.8.2 Shell

```

def binSe(kumpulan, target):
    low = 0
    high = len(kumpulan) - 1
    listku = []

    while low <= high:
        if kumpulan[low] == target:
            listku.append(low)
            low += 1
        else:
            low += 1
    return listku

```

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> s = [2, 6, 5, 6, 4, 6, 7, 8, 6, 2, 4, 10, 4, 14, 15]
>>> binSe(s, 6)
[1, 3, 5, 8]
>>> binSe(s, 2)
[0, 9]
>>>

```

8.

08.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- 0...
Python 3.8.2 Shell

```

print("""
Soal :
Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk
menemukan angka yang TEPAT. untuk angka 1-1000 dibutuhkan
maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya""")

print("""
Jawab :
Ada dua kemungkinan pola yang bisa digunakan.
Misalkan, angka yang akan ditebak adalah 70.

-POLA PERTAMA-
a = nilai tebakan pertama // 2
tebakan selanjutnya = nilai tebakan "lebih dari" + a

*jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
tetap nilai lebih dari sebelumnya

a = a // 2

SIMULASI
tebakan ke-1 : 50 (mengambil nilai tengah) Jawaban = "Lebih dari Itu"
tebakan ke-2 : 75 (dari 50 + 25) Jawaban = "Kurang dari Itu"
tebakan ke-3 : 62 (dari 50 + 12) Jawaban = "Lebih dari Itu"
tebakan ke-4 : 68 (dari 62 + 6) Jawaban = "Lebih dari Itu"
tebakan ke-5 : 71 (dari 68 + 3) Jawaban = "Kurang dari Itu"
tebakan ke-6 : 69 (dari 68 + 1) Jawaban = "Lebih dari Itu"
tebakan ke-7 : antara 71 dan 69 hanya ada 1 angka = 70

-POLA KEDUA-
menggunakan barisan geometri Sn = 2^n
barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
Misal angka yang akan ditebak adalah 68
Tebakan ke-1 : 64 dijawab lebih dari itu
Tebakan ke-2 : 96 (dari 64 + 32) dijawab "Kurang dari itu"
Tebakan ke-3 : 80 (dari 64 + 16) dijawab "Kurang dari itu"
Tebakan ke-4 : 72 (dari 64 + 8) dijawab "Kurang dari itu"
Tebakan ke-5 : 68 (dari 64 + 4) dijawab "Lebih dari itu"
Tebakan ke-6 : 70 (dari 68 + 2) dijawab "TEPAT"
""")

```

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MO
DUL- 04\Tugas\08.py
>>>

```

```

Soal :
Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk
menemukan angka yang TEPAT. untuk angka 1-1000 dibutuhkan
maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya

Jawab :
Ada dua kemungkinan pola yang bisa digunakan.
Misalkan, angka yang akan ditebak adalah 70.

-POLA PERTAMA-
a = nilai tebakan pertama // 2
tebakan selanjutnya = nilai tebakan "lebih dari" + a

*jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
tetap nilai lebih dari sebelumnya

a = a // 2

SIMULASI
tebakan ke-1 : 50 (mengambil nilai tengah) Jawaban = "Lebih dari Itu"
tebakan ke-2 : 75 (dari 50 + 25) Jawaban = "Kurang dari Itu"
tebakan ke-3 : 62 (dari 50 + 12) Jawaban = "Lebih dari Itu"
tebakan ke-4 : 68 (dari 62 + 6) Jawaban = "Lebih dari Itu"
tebakan ke-5 : 71 (dari 68 + 3) Jawaban = "Kurang dari Itu"
tebakan ke-6 : 69 (dari 68 + 1) Jawaban = "Lebih dari Itu"
tebakan ke-7 : antara 71 dan 69 hanya ada 1 angka = 70

-POLA KEDUA-
menggunakan barisan geometri Sn = 2^n
barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
Misal angka yang akan ditebak adalah 68

```

```

08.py - E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MODUL- 0...
Python 3.8.2 Shell

```

```

Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk
menemukan angka yang TEPAT. untuk angka 1-1000 dibutuhkan
maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya""")

print("""
Jawab :
Ada dua kemungkinan pola yang bisa digunakan.
Misalkan, angka yang akan ditebak adalah 70.

-POLA PERTAMA-
a = nilai tebakan pertama // 2
tebakan selanjutnya = nilai tebakan "lebih dari" + a

*jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
tetap nilai lebih dari sebelumnya

a = a // 2

SIMULASI
tebakan ke-1 : 50 (mengambil nilai tengah) Jawaban = "Lebih dari Itu"
tebakan ke-2 : 75 (dari 50 + 25) Jawaban = "Kurang dari Itu"
tebakan ke-3 : 62 (dari 50 + 12) Jawaban = "Lebih dari Itu"
tebakan ke-4 : 68 (dari 62 + 6) Jawaban = "Lebih dari Itu"
tebakan ke-5 : 71 (dari 68 + 3) Jawaban = "Kurang dari Itu"
tebakan ke-6 : 69 (dari 68 + 1) Jawaban = "Lebih dari Itu"
tebakan ke-7 : antara 71 dan 69 hanya ada 1 angka = 70

-POLA KEDUA-
menggunakan barisan geometri Sn = 2^n
barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
Misal angka yang akan ditebak adalah 68
Tebakan ke-1 : 64 dijawab lebih dari itu
Tebakan ke-2 : 96 (dari 64 + 32) dijawab "Kurang dari itu"
Tebakan ke-3 : 80 (dari 64 + 16) dijawab "Kurang dari itu"
Tebakan ke-4 : 72 (dari 64 + 8) dijawab "Kurang dari itu"
Tebakan ke-5 : 68 (dari 64 + 4) dijawab "Lebih dari itu"
Tebakan ke-6 : 70 (dari 68 + 2) dijawab "TEPAT"
""")

```

```

Python 3.8.2 (tags/v3.8.2:7b3ab59, Feb 25 2020, 22:45:29) [MSC v.1916 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> = RESTART: E:\TUGAS TUGAS KULIAH\SEMESTER 4\Praktikum Algoritma Struktur data\MO
DUL- 04\Tugas\08.py
>>>

```

```

Soal :
Pada permainan tebak angka, 1-100 dibutuhkan maksimal 7 kali tebakan untuk
menemukan angka yang TEPAT. untuk angka 1-1000 dibutuhkan
maksimal 10 kali tebakan. Mengapa demikian? Bagaimana polanya

Jawab :
Ada dua kemungkinan pola yang bisa digunakan.
Misalkan, angka yang akan ditebak adalah 70.

-POLA PERTAMA-
a = nilai tebakan pertama // 2
tebakan selanjutnya = nilai tebakan "lebih dari" + a

*jika hasil tebakan selanjutnya "kurang dari", maka nilai yang dipakai
tetap nilai lebih dari sebelumnya

a = a // 2

SIMULASI
tebakan ke-1 : 50 (mengambil nilai tengah) Jawaban = "Lebih dari Itu"
tebakan ke-2 : 75 (dari 50 + 25) Jawaban = "Kurang dari Itu"
tebakan ke-3 : 62 (dari 50 + 12) Jawaban = "Lebih dari Itu"
tebakan ke-4 : 68 (dari 62 + 6) Jawaban = "Lebih dari Itu"
tebakan ke-5 : 71 (dari 68 + 3) Jawaban = "Kurang dari Itu"
tebakan ke-6 : 69 (dari 68 + 1) Jawaban = "Lebih dari Itu"
tebakan ke-7 : antara 71 dan 69 hanya ada 1 angka = 70

-POLA KEDUA-
menggunakan barisan geometri Sn = 2^n
barisan yang terjadi adalah : 2, 4, 8, 16, 32, 64
Misal angka yang akan ditebak adalah 68
Tebakan ke-1 : 64 dijawab lebih dari itu
Tebakan ke-2 : 96 (dari 64 + 32) dijawab "Kurang dari itu"
Tebakan ke-3 : 80 (dari 64 + 16) dijawab "Kurang dari itu"
Tebakan ke-4 : 72 (dari 64 + 8) dijawab "Kurang dari itu"
Tebakan ke-5 : 68 (dari 64 + 4) dijawab "Lebih dari itu"
Tebakan ke-6 : 70 (dari 68 + 2) dijawab "TEPAT"

```