

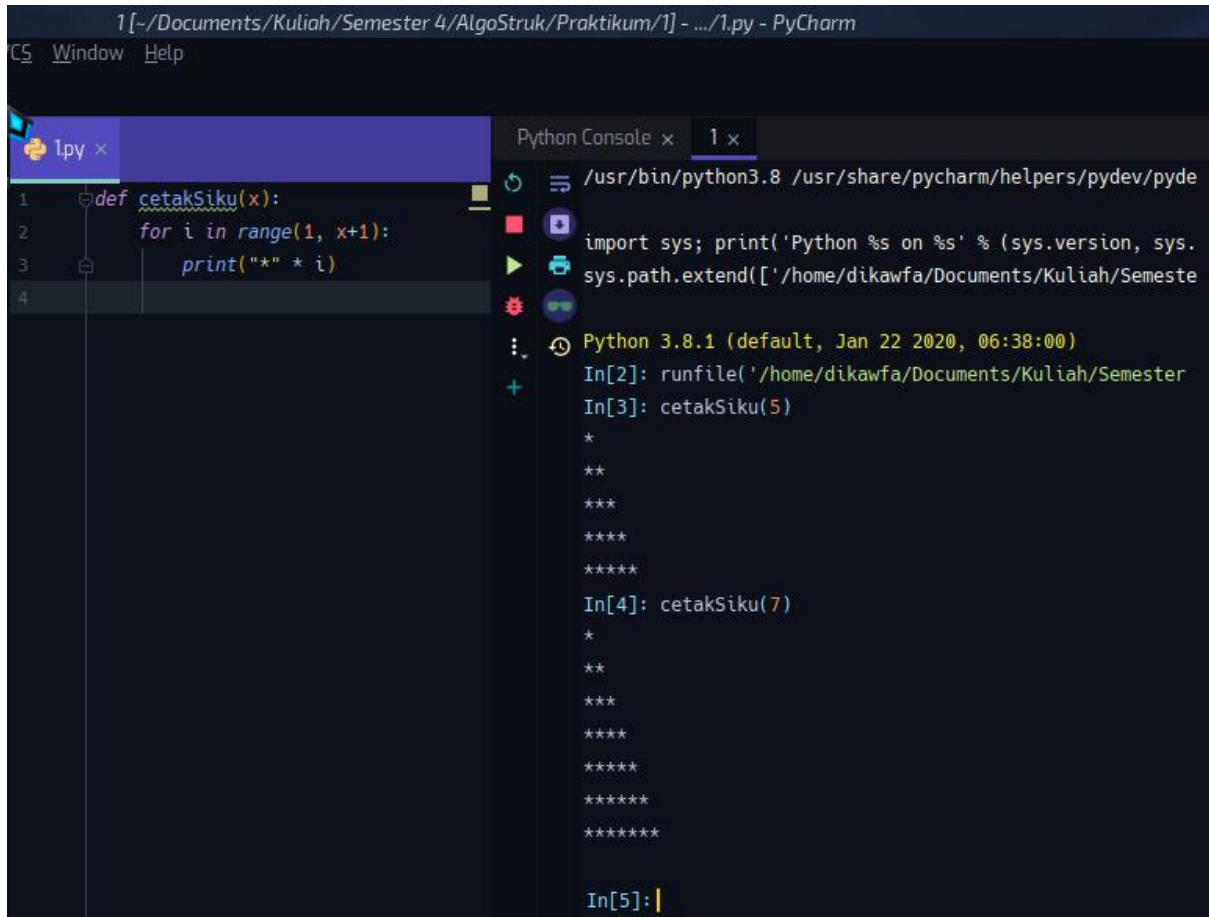
Laporan Praktikum Algoritma dan Struktur Data

NIM : L200180014

Nama : Andika Wirapala F. A.

Modul : 1

1.



The screenshot shows the PyCharm IDE with a file named `1.py`. The code defines a function `cetakSiku(x)` that prints a right-angled triangle of asterisks. The Python Console shows the execution of the function with `x=5` and `x=7`, resulting in two patterns of asterisks.

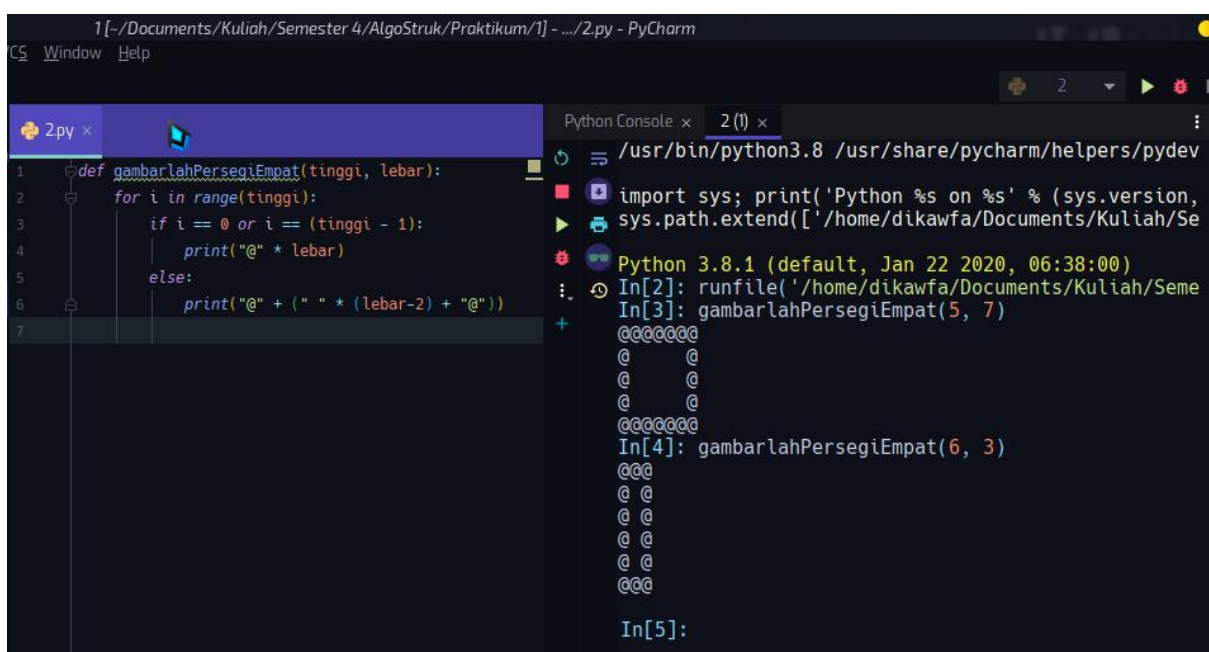
```
1 [-/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1] - .../1.py - PyCharm
CS Window Help

1.py x
1 def cetakSiku(x):
2     for i in range(1, x+1):
3         print("*" * i)
4

Python Console x 1 x
/usr/bin/python3.8 /usr/share/pycharm/helpers/pydev/pyde
import sys; print('Python %s on %s' % (sys.version, sys.
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semeste

Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Semester
In[3]: cetakSiku(5)
*
**
***
****
*****
In[4]: cetakSiku(7)
*
**
***
****
*****
*****
In[5]:
```

2.



The screenshot shows the PyCharm IDE with a file named `2.py`. The code defines a function `gambarlahPersegiEmpat(tinggi, lebar)` that prints a rectangle of '@' symbols. The Python Console shows the execution of the function with `(5, 7)` and `(6, 3)`, resulting in two patterns of '@' symbols.

```
1 [-/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1] - .../2.py - PyCharm
CS Window Help

2.py x
1 def gambarlahPersegiEmpat(tinggi, lebar):
2     for i in range(tinggi):
3         if i == 0 or i == (tinggi - 1):
4             print("@" * lebar)
5         else:
6             print("@" + (" " * (lebar-2)) + "@")
7

Python Console x 2 () x
/usr/bin/python3.8 /usr/share/pycharm/helpers/pydev
import sys; print('Python %s on %s' % (sys.version, sys.
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Se

Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Seme
In[3]: gambarlahPersegiEmpat(5, 7)
@@@@@@@
@       @
@       @
@       @
@@@@@@@
In[4]: gambarlahPersegiEmpat(6, 3)
@@@
@ @
@ @
@ @
@ @
@@@
In[5]:
```

3.

Kode [-/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1/Kode] - .../3.py - PyCharm

Run Tools VCS Window Help

3.py x

```

1  vokal = 'aiueo'
2  def jumlahHurufVokal(kata):
3      kata = kata.lower()
4      jmlV = 0
5      pjgKata = len(kata)
6      for huruf in kata:
7          if huruf in vokal:
8              jmlV += 1
9      print('(' + str(pjgKata) + ',' + str(jmlV) + ')')
10
11 def jumlahHurufKonsonan(kata):
12     kata = kata.lower()
13     jmlK = 0
14     pjgKata = len(kata)
15     for huruf in kata:
16         if huruf not in vokal:
17             jmlK += 1
18     print('(' + str(pjgKata) + ',' + str(jmlK) + ')')
19

```

3 x

```

"/home/dikawfa/Documents/Kuliah/Semester 4/A
import sys; print('Python %s on %s' % (sys.v
sys.path.extend(['/home/dikawfa/Documents/Ku
Python Console
>>> runfile('/home/dikawfa/Documents/Kuliah/
>>> jumlahHurufVokal("Surakarta")
(9,4)
>>> jumlahHurufKonsonan("Surakarta")
(9,5)
>>> jumlahHurufVokal("AndiWFA")
(7,3)
>>> jumlahHurufKonsonan("AndiWFA")
(7,4)
>>> jumlahHurufVokal("aaaAAA")
(6,6)
>>> jumlahHurufKonsonan("aaaAAA")
(6,0)
>>>

```

4.

1 [-/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1] - .../4.py - PyCharm

Window Help

4.py x

```

1  def rerata(b):
2      jumlah = 0
3      for i in b:
4          jumlah += i
5      rata = jumlah/len(b)
6      return rata
7

```

3 x 4 (1) x

```

/usr/bin/python3.8 /usr/share/pycharm/helper
import sys; print('Python %s on %s' % (sys.v
sys.path.extend(['/home/dikawfa/Documents/Ku
Python 3.8.1 (default, Jan 22 2020, 06:38:00
In[2]: runfile('/home/dikawfa/Documents/Kuli
In[3]: rerata([1, 2, 3, 4, 5])
Out[3]: 3.0
In[4]: g = [80, 75, 85, 90]
In[5]: rerata(g)
Out[5]: 82.5
In[6]:

```

5.

The screenshot shows the PyCharm IDE with a Python script in the editor and its console output. The script defines a function `apakahPrima(n)` that checks if a number `n` is prime. It uses a list `primaKecil` for small primes and a loop for larger numbers. The console shows the execution of the script, with the function being called for various inputs.

```

1 from math import sqrt as sq
2
3 def apakahPrima(n):
4     n = int(n)
5     assert n >= 0
6     primaKecil = [2, 3, 5, 7, 11]
7     bukanPrKecil = [0, 1, 4, 6, 8, 9, 10]
8     if n in primaKecil:
9         return True
10    elif n in bukanPrKecil:
11        return False
12    else:
13        for i in range(2, int(sq(n) + 1)):
14            if (n % i) == 0:
15                return False
16        else:
17            return True
18

```

Python Console:

```

/usr/bin/python3.8 /usr/share/pycharm/helper
import sys; print('Python %s on %s' % (sys.v
sys.path.extend(['/home/dikawfa/Documents/Ku
Python 3.8.1 (default, Jan 22 2020, 06:38:00
In[2]: runfile('/home/dikawfa/Documents/Kuli
In[3]: apakahPrima(5)
Out[3]: True
In[4]: apakahPrima(29)
Out[4]: True
In[5]: apakahPrima(70)
Out[5]: False
In[6]: apakahPrima(997)
Out[6]: True
In[7]: apakahPrima(999)
Out[7]: False
In[8]:

```

6.

The screenshot shows the PyCharm IDE with a Python script in the editor and its console output. The script defines a function `apakahPrima(n)` that checks if a number `n` is prime. It uses a list `primaKecil` for small primes and a loop for larger numbers. The console shows the execution of the script, with the function being called for various inputs.

```

1 from math import sqrt as sq
2
3 def apakahPrima(n):
4     n = int(n)
5     assert n >= 0
6     primaKecil = [2, 3, 5, 7, 11]
7     bukanPrKecil = [0, 1, 4, 6, 8, 9, 10]
8     if n in primaKecil:
9         return True
10    elif n in bukanPrKecil:
11        return False
12    else:
13        for i in range(2, int(sq(n) + 1)):
14            if (n % i) == 0:
15                return False
16        else:
17            return True
18
19 for i in range(2, 1001):
20     if apakahPrima(i):
21         print(i)
22

```

Python Console:

```

809
811
821
823
827
829
839
853
857
859
863
877
881
883
887
907
911
919
929
937
941
947
953
967
971
977
983
991
997
In[3]:

```

7.

The image shows a PyCharm IDE window with a file named `7.py`. The code defines a function `faktorPrima(x)` that finds the prime factors of `x`. It uses a list `a` to store prime factors and `b` to store the remaining value. The function iterates from 2 to `x+1` to find prime factors, then uses a while loop to divide the remaining value by the found prime factors until it is 1. The function prints the list of prime factors.

```

1 def faktorPrima(x):
2     a = []
3     b = []
4     bil = x
5     for i in range(2, x+1):
6         prima = True
7         for u in range(2, i):
8             if i % u == 0:
9                 prima = False
10        if prima:
11            a.append(i)
12    idx = 0
13    while bil > 1:
14        try:
15            if (bil % a[idx]) == 0:
16                hasil = bil / a[idx]
17                bil = hasil
18                b.append(a[idx])
19            else:
20                idx = idx + 1
21        except IndexError:
22            break
23    print(b)
24

```

The right pane shows the execution results for the function `faktorPrima` with various inputs:

```

/usr/bin/python3.8 /usr/share/pycharm/helper
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1'])
Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1')
In[3]: faktorPrima(13)
Out[3]: [13]
In[4]: faktorPrima(12)
Out[4]: [2, 2, 3]
In[5]: faktorPrima(10)
Out[5]: [2, 5]
In[6]: faktorPrima(120)
Out[6]: [2, 2, 2, 3, 5]
In[7]: faktorPrima(100)
Out[7]: [2, 2, 5, 5]
In[8]: faktorPrima(197)
Out[8]: [197]
In[9]: faktorPrima(29)
Out[9]: [29]
In[10]:

```

8.

The image shows a PyCharm IDE window with a file named `8.py`. The code defines a function `apakahTerkandung(a, b)` that checks if string `a` is a subset of string `b`. It returns `a in b`.

```

1 def apakahTerkandung(a, b):
2     return a in b
3

```

The right pane shows the execution results for the function `apakahTerkandung` with various inputs:

```

/usr/bin/python3.8 /usr/share/pycharm/helpers/pydev/pydevconsole.py
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1'])
Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1')
In[3]: h = "do"
In[4]: k = "Indonesia tanah air beta"
In[5]: apakahTerkandung(h, k)
Out[5]: True
In[6]: apakahTerkandung("alfa", k)
Out[6]: False
In[7]: apakahTerkandung("alfa", "alfatoksine")
Out[7]: True
In[8]: apakahTerkandung(k, "do re mi")
Out[8]: False
In[9]: apakahTerkandung(h, "do re mi")
Out[9]: True
In[10]:

```


9.

The screenshot shows the PyCharm IDE with a file named `9.py`. The code is a loop that iterates from 1 to 101, checking if the number is divisible by 3 and 5. If both, it prints "Python UMS". If only 3, it prints "Python". If only 5, it prints "UMS".

```

1 for i in range(1, 101):
2     if (i % 3) == 0 and (i % 5) == 0:
3         i = "Python UMS"
4     elif (i % 3) == 0:
5         i = "Python"
6     elif (i % 5) == 0:
7         i = "UMS"
8     print(i)
9

```

The output window shows the following results:

```

Python
73
74
Python UMS
76
77
Python
79
UMS
Python
82
83
Python
UMS
86
Python
88
89
Python UMS
91
92
Python
94
UMS
Python
97
98
Python
UMS

```

The prompt `In[3]:` is visible at the bottom of the output window.

10.

The screenshot shows the PyCharm IDE with a file named `10.py`. The code defines a function `selesaikanABC(a, b, c)` that calculates the roots of a quadratic equation $ax^2 + bx + c = 0$. It uses the discriminant $D = b^2 - 4ac$ to determine if there are real roots. If $D < 0$, it prints a message. Otherwise, it calculates the roots x_1 and x_2 and returns them.

```

1 from math import sqrt as akar
2 def selesaikanABC(a, b, c):
3     a = float(a)
4     b = float(b)
5     c = float(c)
6     D = b**2 - 4*a*c
7     if D < 0:
8         print("Determinan negatif. Persamaan tidak mempunyai akar real.")
9     else:
10        x1 = (-b + akar(D))/(2*a)
11        x2 = (-b - akar(D))/(2*a)
12        hasil = (x1, x2)
13        return hasil
14

```

The output window shows the following results:

```

/usr/bin/python3.8 /usr/share/pycharm/helpers/pydev/pydevc
import sys; print('Python %s on %s' % (sys.version, sys.pl
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semester
Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Semester 4/
In[3]: selesaikanABC(1, 2, 3)
Determinan negatif. Persamaan tidak mempunyai akar real.
In[4]: selesaikanABC(3, 4, 5)
Determinan negatif. Persamaan tidak mempunyai akar real.
In[5]: selesaikanABC(1, -5, 6)
Out[5]: (3.0, 2.0)
In[6]: selesaikanABC(2, -4, 9)
Determinan negatif. Persamaan tidak mempunyai akar real.
In[7]: selesaikanABC(1, -5, 6)
Out[7]: (3.0, 2.0)
In[8]:

```

```

1 def cekKabisat(n):
2     if n % 4 == 0:
3         if n % 100 == 0 and n % 400 == 0:
4             return True
5         elif n % 100 == 0 and n % 400 != 0:
6             return False
7         return True
8     return False
9
10
11 def apakahKabisat(n):
12     if cekKabisat(n):
13         print(str(n) + " adalah Tahun kabisat")
14     else:
15         print(str(n) + " bukan Tahun kabisat")
16
17
18 # Cek langsung beberapa contoh tahun
19 apakahKabisat(1896)
20 apakahKabisat(1897)
21 apakahKabisat(1900)
22 apakahKabisat(2000)
23 apakahKabisat(2004)
24 apakahKabisat(2100)
25

```

```

/usr/bin/python3.8 /usr/share/pycharm/he
import sys; print('Python %s on %s' % (s
sys.path.extend(['/home/dikawfa/Document
Python 3.8.1 (default, Jan 22 2020, 06:3
In[2]: runfile('/home/dikawfa/Documents/
1896 adalah Tahun kabisat
1897 bukan Tahun kabisat
1900 bukan Tahun kabisat
2000 adalah Tahun kabisat
2004 adalah Tahun kabisat
2100 bukan Tahun kabisat
In[3]: apakahKabisat(2020)
2020 adalah Tahun kabisat
In[4]: apakahKabisat(2024)
2024 adalah Tahun kabisat
In[5]: apakahKabisat(2400)
2400 adalah Tahun kabisat
In[6]: apakahKabisat(2600)
2600 bukan Tahun kabisat
In[7]:

```

11.

```

1 import random
2
3 r = random.randint(1, 100)
4 a = ""Coba Tebak sebuah angka bulat
5 antara 1 sampai 100 yang saya simpan.""
6
7 print(a)
8
9 b = "Masukkan tebakan ke-"
10 f = ":%> "
11 c = 1
12 d = str(c)
13
14 for i in range(1, 100):
15     e = (b+d+f)
16     a = int(input(e))
17     c += 1
18     d = str(c)
19     if a < r:
20         print("Itu terlalu kecil. Coba lagi.")
21     elif a > r:
22         print("Itu terlalu besar. Coba lagi.")
23     elif a == r:
24         print("Ya. Anda benar")
25         break
26

```

```

/usr/bin/python3.8 /usr/share/pycharm/helpers/
import sys; print('Python %s on %s' % (sys.ver
sys.path.extend(['/home/dikawfa/Documents/Kuli
Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuli
Coba Tebak sebuah angka bulat
antara 1 sampai 100 yang saya simpan.
Masukkan tebakan ke-1:> ? 50
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-2:> ? 75
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-3:> ? 90
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-4:> ? 99
Itu terlalu besar. Coba lagi.
Masukkan tebakan ke-5:> ? 96
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-6:> ? 97
Itu terlalu kecil. Coba lagi.
Masukkan tebakan ke-7:> ? 98
Ya. Anda benar
In[3]:

```

12.

13.

The screenshot shows a PyCharm IDE window titled "1 [-/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1] - .../13.py - PyCharm". The editor displays a Python function `katakan(bil)` that converts a number into its Indonesian word representation. The function uses a list of words for digits 1-11 and recursive calls for larger numbers. The console on the right shows the execution of the script, including a system version check and several test cases for the `katakan` function.

```

1 def katakan(bil):
2     angka = ["", "Satu ", "Dua ", "Tiga ", "Empat ", "Lima ", "Enam ",
3             "Tujuh ", "Delapan ", "Sembilan ", "Sepuluh ", "Sebelas "]
4     hasil = ""
5     n = int(bil)
6     if 0 <= n <= 11:
7         hasil = angka[n]
8     elif n < 20:
9         hasil = katakan(n - 10) + "Belas"
10    elif n < 100:
11        hasil = katakan(n / 10) + "Puluh " + katakan(n % 10)
12    elif n < 200:
13        hasil = "Seratus " + katakan(n - 100)
14    elif n < 1000:
15        hasil = katakan(n / 100) + "Ratus " + katakan(n % 100)
16    elif n < 2000:
17        hasil = "Seribu " + katakan(n - 1000)
18    elif n < 1000000:
19        hasil = katakan(n / 1000) + "Ribuh " + katakan(n % 1000)
20    elif n < 100000000:
21        hasil = katakan(n / 1000000) + "Juta " + katakan(n % 1000000)
22    elif n > 1000000000:
23        hasil = "Maaf, program tidak membaca angka lebih dari Satu Milyar"
24    return hasil
25

```

```

/usr/bin/python3.8 /usr/share/pycharm/helpers/pydev/pydevconsole.py
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/'])
Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/13.py')
In[3]: katakan(1945)
Out[3]: 'Seribu Sembilan Ratus Empat Puluh Lima '
In[4]: katakan(1350000)
Out[4]: 'Satu Juta Tiga Ratus Lima Puluh Ribu '
In[5]: katakan(13)
Out[5]: 'Tiga Belas'
In[6]: katakan(114)
Out[6]: 'Seratus Empat Belas'
In[7]: katakan(175250)
Out[7]: 'Seratus Tujuh Puluh Lima Ribu Dua Ratus Lima Puluh '
In[8]:

```

14.

The screenshot shows a PyCharm IDE window titled "1 [-/Documents/Kuliah/Semester 4/AlgoStruk/Praktikum/1] - .../14.py - PyCharm". The editor displays a Python function `formatRupiah(n)` that formats a number as a string with commas as thousands separators. The console on the right shows the execution of the script, including a system version check and several test cases for the `formatRupiah` function.

```

1 def formatRupiah(n):
2     y = str(n)
3     if len(y) <= 3:
4         return 'Rp ' + y
5     else:
6         p = y[-3:]
7         q = y[:-3]
8         return formatRupiah(q) + '.' + p
9

```

```

/usr/bin/python3.8 /usr/share/pycharm/helpers/pydev/pydevconsole.py
import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/'])
Python 3.8.1 (default, Jan 22 2020, 06:38:00)
In[2]: runfile('/home/dikawfa/Documents/Kuliah/Semester 4/AlgoStruk/14.py')
In[3]: formatRupiah(1350000)
Out[3]: 'Rp 1.350.000'
In[4]: formatRupiah(1000)
Out[4]: 'Rp 1.000'
In[5]: formatRupiah(1500)
Out[5]: 'Rp 1.500'
In[6]: formatRupiah(1500000)
Out[6]: 'Rp 1.500.000'
In[7]:

```