

Laporan Praktikum Kecerdasan Buatan

Semester Ganjil 2023/2024

Jurusan Teknik Elektro
Program Studi S1 Teknik Elektro
Fakultas Teknik
Universitas Tidar

Praktikum ke- : 1
Judul Praktikum : Pelacakan dalam Python

Nama : Jerico Christianto
NIM : 2220501082

“Laporan praktikum ini saya kerjakan dan selesaikan dengan sebaik-baiknya tanpa melakukan tindak kecurangan. Apabila di kemudian hari ditemukan adanya kecurangan pada laporan ini, maka saya bersedia menerima konsekuensinya.”

Tertanda,



Jerico Christianto

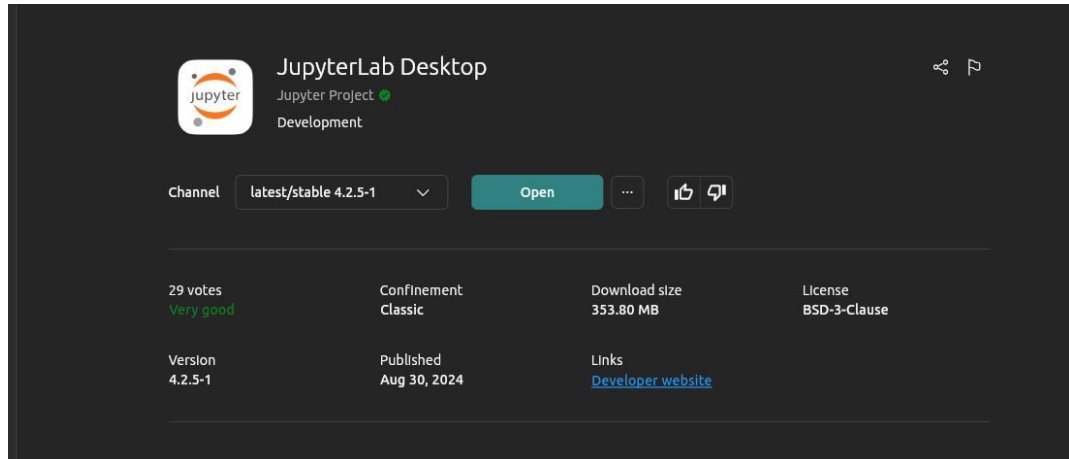
2220501082

A. Tujuan Praktikum

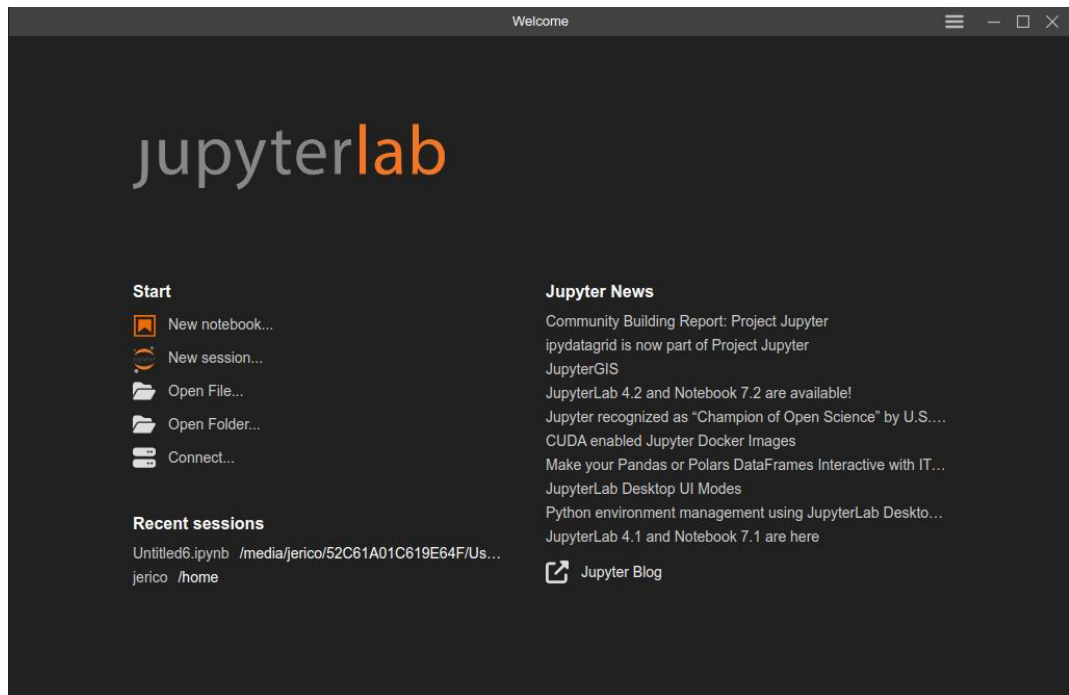
1. Memahami Konsep Dasar Kecerdasan Buatan
2. Memahami Konsep Dasar Pelacakan
3. Menerapkan Algoritma Pelacakan dalam Python

B. Langkah Praktikum

1. Menginstall software jupyterlab, yang saya gunakan yaitu jupyterlab berbasis linux



2. Buka aplikasi JupyterLab



3. Setelah terbuka jalankan perintah “pip install kanren”

```
[1]: pip install kanren
Collecting kanren
  Using cached kanren-0.2.3.tar.gz (23 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting toolz (from kanren)
  Using cached toolz-0.12.1-py3-none-any.whl.metadata (5.1 kB)
Collecting multipledispatch (from kanren)
  Using cached multipledispatch-1.0.0-py3-none-any.whl.metadata (3.8 kB)
Collecting unification (from kanren)
  Using cached unification-0.2.2-py2.py3-none-any.whl.metadata (3.3 kB)
Using cached multipledispatch-1.0.0-py3-none-any.whl (12 kB)
Using cached toolz-0.12.1-py3-none-any.whl (56 kB)
Using cached unification-0.2.2-py2.py3-none-any.whl (10 kB)
Building wheels for collected packages: kanren
  Building wheel for kanren (setup.py): started
  Building wheel for kanren (setup.py): finished with status 'done'
  Created wheel for kanren: filename=kanren-0.2.3-py3-none-any.whl size=15890 sha256=8d9e30e9d835b7b241ff75de7ef65f3ff564519812456a031a7f4e2263b705
  Stored in directory: c:\users\asus\appdata\local\pip\cache\wheels\8c\09\6e\3d8304369a068f6bcad99d1e11b86a68ed3f795b24b2a3642
Successfully built kanren
Installing collected packages: multipledispatch, toolz, unification, kanren
Successfully installed kanren-0.2.3 multipledispatch-1.0.0 toolz-0.12.1 unification-0.2.2
Note: you may need to restart the kernel to use updated packages.
```

4. Tunggu prosesnya selesai lalu mulai ulang kernelnya

```
[3]: pip install kanren
```

Restart the kernel

5. Setelah selesai dimuat, periksa kembali apakah kanren sudah terpasang

```
[1]: pip install kanren
Requirement already satisfied: kanren in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (0.2.3)
Requirement already satisfied: toolz in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (from kanren) (0.12.1)
Requirement already satisfied: multipledispatch in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (from kanren) (1.0.0)
Requirement already satisfied: unification in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (from kanren) (0.2.2)
Note: you may need to restart the kernel to use updated packages.
```

6. Coba menulis modul logika dalam kanren

a.) Variabel Logika

```
[3]: x= var()
[4]: y= var()
```

b.) Constraint dan Goal

```
[6]: output= run(1, x, eq((x,y), (y,3)))
[7]: print (output)
```

c.) Ekspresi Logika

```
[1]: from collections.abc import Iterable
[2]: from kanren.core import var, eq, run
[3]: x= var()
[4]: y= var()
[6]: output= run(1, x, eq((x,y), (y,3)))
[7]: print (output)
```

d.) Fakta dan Relasi

```
[20]: from collections.abc import Iterable
      from kanren.facts import Relation, facts
      from kanren.core import var, eq, run
      anak= Relation()
      facts(anak, ("Adi", "Budi"), ("Cica", "Adi"))
      x=var()
      output = run(1,x,anak(x,"Budi"))
      print("\nNama anak Budi : ", output[0])
```

Fakta dan Relasi

```
[23]: from collections.abc import Iterable
      from kanren.facts import Relation, facts
      from kanren.core import var, run, conde

      def get_sibling(x,y):
          temp = var()
          return conde((parent(temp, x), parent(temp, y)))

      if __name__ == '__main__':
          ortu = Relation()
          facts(ortu, ("Budi", "Adi"),
                ("Budi", "Ani"),
                ("Adi", "Cica"))

          x = var()
          output = run(1, x, ortu(x, "Adi"))
          print("\nNama Ayah Adi : ", output[0])

      # contoh definisi saudara (sibling) menggunakan relasi baru
      saudara = Relation()
      facts(saudara, ("Adi", "Ani"), ("Ani", "Adi"))
      brother = run(0, x, saudara(x, "Ani"))
      print("\nNama saudara laki-laki Ani : ", brother[0])
      sister = run(0, x, saudara(x, "Adi"))
      print("\nNama saudara Perempuan Adi : ", sister[0])

      # contoh definisi saudara (sibling) menggunakan relasi yang sudah ada (parent)
      saudara = run(0, x, get_sibling(x, "Adi"))
      saudara = [x for x in saudara if x != "Adi"]
      print("\nNama saudara Adi : ")
      for item in saudara:
          print(item)
```

C. Hasil Praktikum

1. Instalasi Kanren

```
[1]: pip install kanren

Collecting kanren
  Using cached kanren-0.2.3.tar.gz (23 kB)
  Preparing metadata (setup.py): started
  Preparing metadata (setup.py): finished with status 'done'
Collecting toolz (from kanren)
  Using cached toolz-0.12.1-py3-none-any.whl.metadata (5.1 kB)
Collecting multipledispatch (from kanren)
  Using cached multipledispatch-1.0.0-py3-none-any.whl.metadata (3.8 kB)
Collecting unification (from kanren)
  Using cached unification-0.2.2-py2.py3-none-any.whl.metadata (3.3 kB)
Using cached multipledispatch-1.0.0-py3-none-any.whl (12 kB)
Using cached toolz-0.12.1-py3-none-any.whl (56 kB)
Using cached unification-0.2.2-py2.py3-none-any.whl (10 kB)
Building wheels for collected packages: kanren
  Building wheel for kanren (setup.py): started
  Building wheel for kanren (setup.py): finished with status 'done'
  Created wheel for kanren: filename=kanren-0.2.3-py3-none-any.whl size=15890 sha256=8d8e30e9d835b7b241ff775de67ef85f3ff564519812456a031a774e2263b795
  Stored in directory: c:\users\asus\appdata\local\pip\cache\wheels\8c\09\6e\3d8304369a060f6bcad99d1e11b860a68ed3f795b24b2a3642
Successfully built kanren
Installing collected packages: multipledispatch, toolz, unification, kanren
Successfully installed kanren-0.2.3 multipledispatch-1.0.0 toolz-0.12.1 unification-0.2.2
Note: you may need to restart the kernel to use updated packages.
```

Instalasi Kanren Berhasil

```
[1]: pip install kanren

Requirement already satisfied: kanren in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (0.2.3)
Requirement already satisfied: toolz in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (from kanren) (0.12.1)
Requirement already satisfied: multipledispatch in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (from kanren) (1.0.0)
Requirement already satisfied: unification in /home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages (from kanren) (0.2.2)
Note: you may need to restart the kernel to use updated packages.
```

- Kanren adalah sebuah mesin inferensi yang menggunakan logika predikat orde pertama. Sederhananya, Kanren adalah sebuah program komputer yang sangat pintar dalam menarik kesimpulan dari informasi yang diberikan. Informasi ini kita berikan dalam bentuk fakta dan aturan.
- Inferensi adalah proses menarik kesimpulan baru dari fakta dan aturan yang sudah ada. Misalnya, jika kita tahu bahwa Budi adalah anak dari Ani, dan kita memiliki aturan bahwa orang tua dari seorang anak adalah orang tua, maka Kanren dapat menyimpulkan bahwa Ani adalah orang tua dari Budi.

2. Modul Logika dalam Kanren

a.) Variabel Logika

```
[3]: x= var()
[4]: y= var()
```

b.) Constraint dan Goal

```
[6]: output= run(1, x, eq((x,y), (y,3)))
[7]: print (output)
```

c.) Ekspresi Logika

```
[1]: from collections.abc import Iterable
[2]: from kanren.core import var, eq, run
[3]: x= var()
[4]: y= var()
[6]: output= run(1, x, eq((x,y), (y,3)))
[7]: print (output)
(3,)
```

d.) Fakta dan Relasi

```
[20]: from collections.abc import Iterable
      from kanren.facts import Relation, facts
      from kanren.core import var, eq, run
      anak= Relation()
      facts(anak, ("Adi", "Budi"), ("Cica", "Adi"))
      x=var()
      output = run(1,x,anak(x,"Budi"))
      print("\nNama anak Budi : ", output[0])

      Nama anak Budi :  Adi
```

- a. Fakta adalah pernyataan yang kita anggap benar. Misalnya:
 - i. anak(adi, budi). Ini artinya Adi adalah anak dari Budi.
 - ii. manusia(budi). Ini artinya Budi adalah manusia.
- b. Relasi adalah pernyataan yang menjelaskan hubungan antara fakta-fakta. Misalnya:
 - i. orangtua(X, Y) :- anak(Y, X). Aturan ini berarti jika Y adalah anak dari X, maka X adalah orang tua dari Y.

Kode Program untuk pelacakan silsilah keluarga dalam Python :

```
[23]: from collections.abc import Iterable
from kanren.facts import Relation, facts
from kanren.core import var, run, conde
def get_sibling(x,y):
    temp = var()
    return conde((parent(temp, x), parent(temp, y)))
if __name__ == '__main__':
    ortu = Relation()
    facts(ortu, ("Budi", "Adi"),
          ("Budi", "Ani"),
          ("Adi", "Cica"))
x = var()
output = run(1, x, ortu(x, "Adi"))
print("\nNama Ayah Adi : ", output[0])

# contoh definisi saudara (sibling) menggunakan relasi baru
saudara = Relation()
facts(saudara, ("Adi", "Ani"), ("Ani", "Adi"))
brother = run(0, x, saudara(x, "Ani"))
print("\nNama saudara laki-laki Ani : ", brother[0])
sister = run(0, x, saudara(x, "Adi"))
print("\nNama saudara Perempuan Adi : ", sister[0])

# contoh definisi saudara (sibling) menggunakan relasi yang sudah ada (parent)
saudara = run(0, x, get_sibling(x, "Adi"))
saudara = [x for x in saudara if x != "Adi"]
print("\nNama saudara Adi : ")
for item in saudara:
    print(item)
```

Output yang dihasilkan

```
print("\nNama saudara Perempuan Adi : ", sister[0])

# contoh definisi saudara (sibling) menggunakan relasi yang sudah ada (parent)
saudara = run(0, x, get_sibling(x, "Adi"))
saudara = [x for x in saudara if x != "Adi"]
print("\nNama saudara Adi : ")
for item in saudara:
    print(item)

Nama Ayah Adi : Budi

Nama saudara laki-laki Ani : Adi

Nama saudara Perempuan Adi : Ani

-----
NameError                                Traceback (most recent call last)
Cell In[23], line 25
    22 print("\nNama saudara Perempuan Adi : ", sister[0])
    24 # contoh definisi saudara (sibling) menggunakan relasi yang sudah ada (parent)
--> 25 saudara = run(0,x, get_sibling(x, "Adi"))
    26 saudara = [x for x in saudara if x != "Adi"]
    27 print("\nNama saudara Adi : ")

Cell In[23], line 6, in get_sibling(x, y)
     4 def get_sibling(x,y):
     5     temp = var()
--> 6     return conde((parent(temp, x), parent(temp, y)))

NameError: name 'parent' is not defined

[ ]:
```

Terdapat kesalahan pada kode program dimana **"name 'parent' is not defined"**, ini dikarenakan nama **'parent'** tidak ditemukan melainkan yang ada adalah **'ortu'** seperti di dalam modul

```
ortu = Relation()
facts(ortu, ("Budi", "Adi"),
```


Perbaiki Kode Program :

```

1 from collections.abc import Iterable
2 from kanren.facts import Relation, facts
3 from kanren.core import var, run, conde
4
5 def get_sibling(x,y):
6     temp = var()
7     return conde((parent(temp, x), parent(temp, y)))
8 if __name__ == '__main__':
9     parent = Relation()
10    facts(parent, ("Budi", "Adi"),
11               ("Budi", "Ani"),
12               ("Adi", "Cica"))
13    x = var()
14    output = run(1, x, parent(x, "Adi"))
15    print("\nNama Ayah Adi : ", output[0])
16
17 # contoh definisi saudara (sibling) menggunakan relasi baru
18 saudara = Relation()
19 facts(saudara, ("Adi", "Ani"), ("Ani", "Adi"))
20 brother = run(0, x, saudara(x, "Ani"))
21 print("\nNama saudara laki-laki Ani : ", brother[0])
22 sister = run(0, x, saudara(x, "Adi"))
23 print("\nNama saudara perempuan Adi : ", sister[0])
24
25 # contoh definisi saudara (sibling) menggunakan relasi yang sudah ada (parent)
26 saudara = run(0, x, get_sibling(x, "Adi"))
27 saudara = [x for x in saudara if x != "Adi"]
28 print("\nNama saudara Adi : ")
29 for item in saudara:
30     print(item)

```

Output yang Dihasilkan :

```

Nama Ayah Adi : Budi

Nama saudara laki-laki Ani : Adi

Nama saudara perempuan Adi : Ani

Nama saudara Adi :
Ani

```

D. Kendala Praktikum

1. Terdapat beberapa error saat menjalankan kode program pada jupyterlab tetapi dapat diatasi

```

from kanren.core import var, eq, run

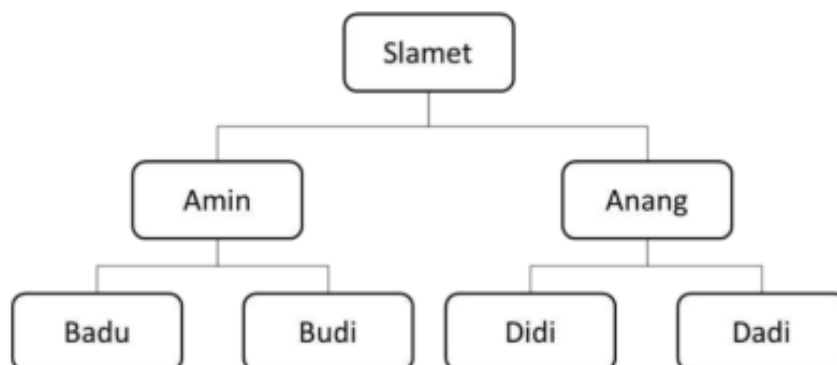
-----
ImportError                                Traceback (most recent call last)
Cell In[3], line 1
----> 1 from kanren.core import var, eq, run

ImportError: cannot import name 'var' from 'kanren.core' (/home/jerico/.config/jupyterlab-desktop/jlab_server/lib/python3.12/site-packages/kanren/core.py)

```

E. Studi Kasus

Membuat Silsilah Keluarga seperti berikut :



Kode Program :

```

1 from collections.abc import Iterable
2 from kanren.facts import Relation, facts
3 from kanren.core import var, run, conde
4
5 def get_sibling(x,y):
6     temp = var()
7     return conde((parent(temp, x), parent(temp, y)))
8 if __name__ == '__main__':
9     ortu = Relation()
10    facts(ortu, ("Slamet", "Amin"),
11           ("Slamet", "Anang"),
12           ("Amin", "Badu"),
13           ("Amin", "Budi"),
14           ("Anang", "Didi"),
15           ("Anang", "Dadi"))
16
17    x = var()
18    output = run(1, x, ortu(x, "Amin"))
19    print("\nNama Ayah Amin : ", output[0])
20    output = run(1, x, ortu(x, "Anang"))
21    print("\nNama Ayah Anang : ", output[0])
22    output = run(1, x, ortu(x, "Badu"))
23    print("\nNama Ayah Badu : ", output[0])
24    output = run(1, x, ortu(x, "Budi"))
25    print("\nNama Ayah Budi : ", output[0])
26    output = run(1, x, ortu(x, "Didi"))
27    print("\nNama Ayah Didi : ", output[0])
28    output = run(1, x, ortu(x, "Dadi"))
29    print("\nNama Ayah Dadi : ", output[0])
30
31    saudara = Relation()
32    facts(saudara, ("Amin", "Anang"),
33              ("Badu", "Budi"),
34              ("Didi", "Dadi"))
35
36    saudara = run(0, x, get_sibling(x, "Amin"))
37    saudara = [x for x in saudara if x != "Amin"]
38    print("\nNama saudara Amin : ")
39    saudara = run(0, x, get_sibling(x, "Budi"))
40    saudara = [x for x in saudara if x != "Budi"]
41    print("\nNama saudara Budi : ")
42    saudara = run(0, x, get_sibling(x, "Didi"))
43    saudara = [x for x in saudara if x != "Didi"]
44    print("\nNama saudara Didi : ")
45    for item in saudara:
46        print(item)
47

```

Output yang Dihasilkan :

```

Nama Ayah Amin : Slamet
Nama Ayah Anang : Slamet
Nama Ayah Badu : Amin
Nama Ayah Budi : Amin
Nama Ayah Didi : Anang
Nama Ayah Dadi : Anang
Nama saudara Amin : Anang
Nama saudara Budi : Badu
Nama saudara Didi : Dadi

```