Buku Panduan

Penggunaan PITON

(Python IntegraTed wOrkspace and visualizatioN)



Tim Penyusun: Elvina Mewati Ayub Oscar Karnalim Maresha Caroline Wijanto

FAKULTAS TEKNOLOGI INFORMASI UNIVERSITAS KRISTEN MARANATHA

2018

KATA PENGANTAR

PITON (*Python IntegraTed wOrkspace and visualizatioN*) adalah aplikasi yang menyediakan lingkungan pembelajaran (*learning environment*) untuk pemrograman dasar dalam bahasa pemrograman Python yang dilengkapi fitur visualisasi dan fitur berbasis kebutuhan pengguna. Aplikasi ini disediakan untuk membantu pemula yang belajar pemrograman menggunakan bahasa Python, agar pemula dapat memahami jalannya eksekusi program dengan memakai fitur visualisasi. Fitur visualisasi dapat digunakan atau tidak digunakan sesuai kebutuhan pengguna.

Buku Panduan ini tersusun atas 3 bagian yaitu:

- Cara Instalasi Perangkat Yang Diperlukan Aplikasi PITON Untuk dapat menjalankan aplikasi ini, pada komputer yang digunakan harus sudah terinstalasi Sistem Operasi Windows 10, .NET Framework 3.5, serta kompailer Python versi 3.6.2.
- 2. Fitur Utama PITON

Dijelaskan fitur utama yang disediakan oleh PITON bagi pengguna.

3. Cara Penggunaan PITON

Dijelaskan cara penggunaan PITON dalam dua modus, yaitu cara penggunaan tanpa visualisasi dan cara penggunaan dengan visualisasi.

Harapan kami, buku panduan ini dapat membantu pengguna di dalam memanfaatkan aplikasi PITON untuk mempelajari pemrograman dengan bahasa Python.

Bandung, Mei 2018

Tim Penyusun

DAFTAR ISI

KATA PENGANTAR	2
DAFTAR ISI	
I. CARA INSTALASI PERANGKAT YANG DIPERLUKAN APLIKASI PITON	4
II. FITUR UTAMA PITON	7
III. CARA PENGGUNAAN PITON	10
III.1 Penggunaan PITON tanpa fitur visualisasi	10
III.2 Penggunaan PITON dengan fitur visualisasi	

I. CARA INSTALASI PERANGKAT YANG DIPERLUKAN APLIKASI PITON

Perangkat lunak yang diperlukan untuk menggunakan PITON, adalah:

1. Operating System Windows 10

2. Aktivasi .NET Framework 3.5

Berikut adalah langkah-langkah untuk mengaktifkan **.NET Framework** pada Windows 10, seperti pada Gambar 1:

- a. Menekan tombol Windows II pada *keyboard*, kemudian ketik "Windows Features",
 dan tekan Enter. *Dialog box* "Turn Windows features on or off" akan muncul.
- b. Pilih ".NET Framework 3.5 (includes .NET 2.0 dan 3.0) pada *checkbox*, tekan OK dan *reboot* komputer jika diminta.

🖾 Windows Features –		×
Turn Windows features on or off		0
To turn a feature on, select its check box. To turn a feature off, cle check box. A filled box means that only part of the feature is turn	ear its ied on	
Image:		^
Windows Communication Foundation HTTP Activa	tion	
🗌 퉬 Windows Communication Foundation Non-HTTP A	Activa	
Image:		
Active Directory Lightweight Directory Services		
🕀 🗹 퉬 Hyper-V		
🗹 퉬 Internet Explorer 10		
🗄 🗔 퉬 Internet Information Services		
Internet Information Services Hostable Web Core		
🗄 🗹 퉬 Media Features		
🗄 🗌 퉬 Microsoft Message Queue (MSMO) Server		~
<		>
ОК	Can	cel

Gambar 1 Aktivasi .NET Framework

3. Instalasi Python Version 3.6.4

Berikut adalah langkah-langkah untuk mengaktifkan **Python Version 3.6.4** pada Windows 10, yaitu dengan menginstal Python dari CD program sebagai berikut :

a. Salinlah *file* "python-3.6.4.exe" dari folder Python-364 dalam CD program seperti pada gambar 2.



- b. Double klik pada file "python-3.6.4.exe" untuk meng-install Python.
- c. Jendela untuk *setup* Python 3.6.4 akan muncul, berikan *checklist* pada "*Install launcher for all user (recommended*)" dan "*Add Python3.6 to PATH*" seperti pada gambar 3. Kemudian tekan "Install Now" untuk menginstalasi aplikasi Python. Jika komputer sudah pernah terinstalasi Python versi sebelumnya, maka "Install Now" akan berubah menjadi "Upgrade Now" seperti pada gambar 3.



Gambar 3 Instalasi Python

d. Instalasi Python akan berjalan seperti pada Gambar 4.



Gambar 4 Progres Instalasi

e. Setelah selesai melakukan instalasi dan berhasil, maka akan muncul jendela *pop up* seperti gambar 5.



Gambar 5 Instalasi berhasil

II. FITUR UTAMA PITON

PITON - PYTHON INTEGRATED WORKS	PACE AND VISUALIZATION			
() C C C C C C C C C C C C C C C C C C	× 🗈 🖪 🛧 🏕 🗛		I	$- \triangleright \square C \leftarrow \rightarrow \mathbf{X}$
The arc and a second a secon	<pre>#Prome Linear Search Links For Characteristics (Comparison of the standard of the search of the</pre>);)))))) (lst, search)	1 1 1 1 1 st [0] 1 2 1 st [1] None None 9 earch 3 has11 None None 1 0 2 1 1 st [0] 1 2	⊙ Vrouuleari Viridee
	Output	Error	lst [1] None None	
SUBMIT 2 2 3	>>> 2 1 2 5 Nilai Tidak Ditemukan	* >>> F	n 2 search 3	

Gambar 6 Antarmuka PITON

Gambar 6 merupakan tampilan antarmuka aplikasi PITON, setiap bagian dari antarmuka akan dijelaskan sebagai berikut:

Gambar	Command	Fungsi	
4	New File	Membuat <i>file</i> baru	
	Get Folder	Membuka <i>folder</i> berisi <i>file</i> kode program yang berekstensi (*.py)	
	Get File	Membuka <i>file</i> kode program dengan ekstensi (*.py)	
	Save	Menyimpan <i>file</i> kode program	
h	Save As	Menyimpan <i>file</i> kode program	
Î	Delete	Menghapus <i>file</i> dari <i>Source Code</i> <i>Viewer</i>	
×	Cut	Memindahkan potongan kode program	

A. Command Toolbar merupakan tempat fitur-fitur untuk memanipulasi kode program.

Gambar	Command	Fungsi
	Сору	Menyalin atau menggandakan
	1.0	potongan kode program
	Paste	Menempelkan potongan kode
•	i uste	program
•	Undo	Membatalkan suatu perintah yang
	Chuo	sudah dilakukan sebelumnya
•	Redo	Mengulang kondisi yang telah
	Neuo	dibatalkan sebelumnya
		Mengatur kecepatan animasi
		visualisasi. Semakin ke kiri, semakin
	Speed Bar	bertambah kecepatan visualisasi dan
	Speea Bar	sebaliknya. Pengaturan speed bar
		hanya dapat dilakukan jika visualisasi
		berada pada kondisi tahap awal.
	Play	Menjalankan animasi untuk
		visualisasi
	Pause	Menunda visualisasi
C	Repet Mengembalikan kondisi visuali	
	neset	ketahap awal
4	Provious	Memundurkan kondisi visualisasi
	170710415	ketahap sebelumnya
د	Nort	Memajukan kondisi visualisasi
	Ινελι	ketahap setelahnya
×	Stop	Menghentikan visualisasi dan
	5.0p	eksekusi
Run	Run	Menjalankan eksekusi
X Stop	Stop	Menghentikan eksekusi
🗆 Visualisasi	Checkbox Visualisasi	Pilihan visualisasi

- **B.** *Source Code Viewer* merupakan bagian dari PITON untuk menampilkan *folder* dan *file* yang dibuka.
- **C. Panel** *Input* berfungsi untuk memasukkan entitas input baik yang satu ataupun banyak entitas input.
- **D.** Source Code Editor berfungsi untuk memanipulasi kode program.
- E. Panel Output berfungsi untuk menampilkan keluaran hasil eksekusi.
- **F. Panel** *Error* berfungsi untuk menampilkan pesan kesalahan jika kode program tidak berhasil dieksekusi.
- G. Panel Visualisasi berfungsi untuk menampilkan visualisasi variabel.

III. CARA PENGGUNAAN PITON

III.1 Penggunaan PITON tanpa fitur visualisasi

Tahapan penggunaan PITON tanpa fitur visualisasi, secara umum adalah sebagai berikut:

- 1. Salinlah folder PITON dari CD ke komputer.
- 2. Masuk ke folder PITON, kemudian *Double* klik pada file "PITON.exe" untuk menjalankan aplikasi PITON seperti pada gambar 7, sehingga muncul tampilan seperti gambar 8 di layar.





Gambar 8 Tampilan Awal PITON

3. Salinlah kode program dari file "LinearSearch" pada folder Panduan dalam CD program ke bagian *source code editor* untuk mencoba aplikasi PITON, sehingga ditampilkan kode program berikut ini pada editor PITON, seperti pada Gambar 9.

```
def linear_search (lst, search):
    for i in range (0, len(lst),1):
        if (lst [i] == search):
```

```
print ("Nilai ditemukan pada posisi " + str (i+ 1))
return 0
print ("Nilai tidak ditemukan.")
return - 1
def main ():
    n= int (input()
    lst = [None] * n
    for i in range (0 ,n, 1) :
        lst [ i ] = int (input())
        search = int (input())
        a = linear_search (lst, search)
if _____main _____ :
        main ( )
```



Gambar 9 Tampilan kode sumber program dalam PITON

 Menekan tombol *save/save as* pada *command toolbar* untuk menyimpan kode program. Simpanlah kode program dengan nama "LinearSearch" seperti pada Gambar 10.

	Save
3	Save As

Đ.	PITON - PYTHON INTEGRATED WORKSPACE AND VISUALIZATION	© i - ♂ ×
G-	🕼 🖹 🐂 🔭 🕨 🕞 Run 🗌 Visualisasi	
\odot	save As X	<u></u>
source	← → → ↑ □ → This PC → Desktop → SourceCodeTest ∨ δ Search SourceCodeTest ρ + str (±1))	Visual
: Code	Organize • New folder	isasi V
	Assets # ^ Name ^ Date modified Type	ariabe
	bin ≯ No items match your search.	
	j umenecum kim ke maidhe	
	photo1	
	photoledit	
	🐉 Dropbox	
	CheDrive	
	This PC	
) Inpu		
7	ine name Balandarization Save stype:	
	A Hide Folders	
	SUBMIT	
		.
		Þ
	Elvina (1472002)	

Gambar 10 Menyimpan file (SAVE)

5. Kemudian jalankan kode program dengan menekan tombol "Run" pada *command toolbar*.



6. PITON akan menampilkan pesan kesalahan karena kesalahan *syntax* pada kode program, seperti pada Gambar 11.

PITON - PYTHON INTEGRATED WORKSPA			
		> Run Visualisasi	
Contraction of the second seco	<pre>idef linear mearch(lst, mearch); for in range(0, hen(lst), 1); if (lst[1] == search); print("Nilai ditemukan.") return 0 print("Nilai ditemukan.") if of an in(input) if or in range(0, n, 1); if or in range(0, n, 1); if eli] = int(input()) search = int(input(</pre>		∢ Visualisasi Variabel
	4		▼
	Output	Error	
SUBMIT	۵. ۱	>>> line 10 lst = [None]*n	<u>_</u>
		SyntaxError: invalid syntax	
		4	, F ĭ

Gambar 11 Tampilan pesan kesalahan

 Perbaiki kode program dengan menambahkan ")" pada baris ke – 9, seperti pada Gambar 12.

```
def linear_search (lst, search):
    for i in range (0, len (lst), 1):
        if (lst [ i] == search):
            print ("Nilai ditemukan pada posisi " + str (i+1))
           return 0
    print ("Nilai tidak ditemukan." )
    return - 1
def main ():
    n= int (input())
                                #baris ke-9
    lst = [ None ] * n
    for i in range ( 0 , n, 1 ) :
        lst [ i ] = int ( input ( ) )
    search = int ( input ( ) )
    a = linear search ( lst, search )
if name == ' main ':
    main ()
```

٢				×
G		🗶 🖺 🔁 🔶		
Source Code	 Default UnearSearch.py 	<pre>l def linear search(lst, search): for in range(0, hen(lst), 1):</pre>	A	🕑 Visualisasi Variabel
 Input 	SUBMIT	4		
		Output 🕑 Error		
_		Elvina (1472002)		_

Gambar 12 Tampilan hasil perbaikan program

 Kemudian jalankan kembali kode program dengan menekan tombol "Run", sehingga muncul tampilan pada Gambar 13.

Run	
-----	--

PITON - PYTHON INTEGRATED WORKSP	ace and visualization	$ \mathbf{u} _{\mathbf{i}} = \mathbf{\sigma} \times \mathbf{u} _{\mathbf{i}}$
	🗴 🗈 🔁 🛧 🅕 🗙 Xstop 🗌 Visualisasi	
Import bertahap sesuai akur program input:	<pre>idef linear_mearch(lst, search): for 1 in temps(0, log(lat), 1): if of print("Nilai ditemukan pada posisi " + str(i+1)) return 0 print("Nilai ditemukan.") return -1 def main(; pupt()) ist = [Nome]*n lst [i=int(input()) search = int(input()) search = int(input()) if or is int(input()) if</pre>	Vrauitea Veritie
	Output 🕞 Error	
	>>>	A
SUBMIT		
	بر المحمد (1472002)	* }
= 🗆 📄 🌖 F 💬 [e ^ % 🦆 🤹 🤹 🛐 😰	a ← 🦟 ♥ (1)) 12:22 PM 12:22 PM 12:82 PM 12:82 PM

Gambar 13 Pilih Input bertahap

- Pilihlah terlebih dahulu cara *input* yang diinginkan untuk memasukkan *input*. Terdapat
 jenis cara *input* pada PITON, yaitu Input Bertahap, Input dari File Text, dan Input
 Sekaligus. Setiap cara akan dijelaskan secara rinci.
 - a. Input Bertahap



Gambar 14 Tampilan input bertahap

Memasukkan *input* untuk kode program secara bertahap. PITON akan meminta *input* kembali jika dibutuhkan setelah *input* yang dimasukkan sebelumnya berhasil diproses seperti pada Gambar 14 dan Gambar 15.

🕑 Input	Input bertahap sesuai alur program Input :
	SUBMIT

Gambar 15 Tampilan Input

Tahapan penggunaan Input bertahap, adalah :

i. Masukkan angka **3** kemudian tekan tombol "Submit" seperti pada Gambar 16.

PITON - PYTHON INTEGRATED WORKSP	ACE AND VISUALIZATION		$\mathbb{Q} \mid i \mid = -\mathfrak{S} \mid \times$	
	× 🗈 🗈 🔸 🏓	Stop Visualisasi		
 Image: Search.py ♦ LinearSearch.py 	<pre>2</pre>			
ाput bertahap sesuai alur program	-			
E Input:				
4	4			
	Output			
	***	>>>	A	
SUBMIT				
	v	4	• • •	

Gambar 16 Tampilan Input bertahap (1)

ii. Masukkan kembali angka 1 kemudian tekan tombol "Submit" seperti pada Gambar 17.

Ô	PITON - PYTHON INTEGRATED WORKSP	CE AND VISUALIZATION	$ \mathbf{u} = \mathbf{i}$	ε×
G		× 🗈 🖻 🛧 🎓	Stop Visualisasi	
Source Code	 Default LinearSearch.py 	<pre>idef linear_search(lat, search): for in range(0, hen(lat), l): if (lat[] = search : print("Bial ditemukan.pda posisi " + str(i+1)) return 0 print("Li tida ditemukan.") return 0 if n int(input()) if or int(input()) ist = [Wood]*n 0, n, l): for int(j=1; int(input()) is a intear_search(lat, search) if name_ rear(hits, search) if name_ rea</pre>		🕑 Visualisasi Variabel 1
ک ۳	Input bertahap sesuai alur program			
put	Input:			-
		Output	Error	
		>>> ^	>>>	-
	SUBMIT	3		
	3			
			4	-
		Elvine (1472002)		

Gambar 17 Tampilan Input bertahap (2)

iii. Masukkan kembali angka 2 kemudian tekan tombol "Submit" seperti pada Gambar 18.

٢	piton - python integrated workspace and visualization $\square \mid i \mid$ = $\square \mid z \mid$ ×				
G		× 🗈 🗈 🛧 🅕	Stop Visualisasi		
Source Code	 Default LinearSearCupy 	<pre>rdhpy 2 for i in range(0, len(ist), 1): 3</pre>		A	🕑 Vīsualisasi Variabel
<u>د</u>	Input bertahap sesuai alur program				
put	Input : 2				
			\bigcirc	•	
		Output	() Error		
	SURMIT	***	>>>		
	3	1			
	1				
				Ŧ	
			4	Þ	

Gambar 18 Tampilan Input bertahap (3)

iv. Masukkan kembali angka 3 kemudian tekan tombol "Submit" seperti pada Gambar 19.

Ô	PITON - PYTHON INTEGRATED WORKSPA	CE AND VISUALIZATION	Q i -	ē ×
G		× 🗈 🗈 🔸 🏕 🗡	Stop Visualisasi	
🕞 source Code	 Default UnearSearch.py 	<pre>2 for 1 in range(0, len(lsr), l): 3</pre>		(🕑 Visualisasi Variabel ৰ
<u>ج</u>	Input bertahap sesuai alur program			
put	Input:			.
		Output	C Error	•
		>>>	>>>	-
	SUBMIT	3		
	3	1 2		
	2			
				-
			4	

Gambar 19 Tampilan Input bertahap (4)

v. Masukkan kembali angka **2** kemudian tekan tombol "Submit" seperti pada Gambar 20.

Ê	PITON - PYTHON INTEGRATED WORKSPACE AND VISUALIZATION 🕮 i i = 👼 🗙				
G		🗙 🗈 🖪 🛧 🅕 🗙 Xstop 🗌 Visualisasi			
🕞 source Code	Enfault EinearSearch.py	<pre>1 def linear_mearch(lst, search): for in range(0, hen(lst), l): if (lst[i] == search): print("Wilsi ditemukan.pda posisi " + str(i+1)) return 0 print("Wilsi ditemukan.") return 0 print("Wilsi ditak ditemukan.") return 0 print("Nilsi ditemukan.") return 0 print("Nilsi ditak ditemukan.") return 0 print("Nilsi ditemukan.") return 0 return 0</pre>	⊙ Vinalesi Webe		
() 1	Input bertahap sesuai alur program				
put	Input:				
	4	4			
		Output Error			
		>>> ^ >>>	A		
	SUBMIT	3			
	3				
	2 3	3			
		· .	, *		
		Elvina (1472002)			

Gambar 20 Tampilan Input bertahap (5)

vi. Aplikasi PITON akan mengeluarkan *output* hasil kode program pada kolom ekspansi output seperti pada Gambar 21.

PITON - PYTHON INTEGRATED WORKSP	ACE AND VISUALIZATION		\mathbb{Q} $ \mathbf{i} = -\mathbf{\sigma} \times \mathbf{x}$
	× 🗈 🗈 🛧 🏞	Run Visualisasi	
Perfault Default UnearSearch.py Input bertahap secual alur program Input:	<pre>idef linear search(lst, search): for in range(0, len(ist), l): if (lst(i) == search): print("Nilsi distemukan.peda return 0 print("Nilsi distemukan.") def in isk distemukan.") def in range(0, n, l): lst = [Nong)*n if or in range(0, n, l): lst = [Nong)*n if or in range(0, n, l): lst = int(input(1)) seatch = int(input(1)) if main() main()</pre>	posisi " + str(1+1))	Viraulusei Vorabel
	Output	(Error	
SUBART 3 1 2 3 2	>>> 3 1 2 3 2 Nilai ditemukan pada posisi 2		, . , .
	Ely Ely	ina (1472002)	

Gambar 21 Tampilan Output

b. *Input* dari *File Text* (*.txt)

Memasukkan *input* untuk kode program melalui *file text* yang sudah dirancang seperti pada gambar 22. *File text* yang sama dapat digunakan berulang kali. Kelebihan "*Enter*" saat membuat *file text input*, akan membuat *input* untuk kode program dianggap tidak *valid*.

ø	piton - python integrated workspace and visualization $\square \mid \mathbf{i} \mid = = = \square \times$				
G		🗴 🗈 🗈 🔶 Uisualisasi			
⊙ source Code ⊙ Inpu	Pefault UnearSearch.py Input dari file test (".tst)	<pre>lef linear every(list, search):); for in range(0, lenk(lat), l): if (lat(1) == cearch): return 0 return 0 return -1 return -1 for in range(0, n, l): lat = [None]'n lat (l]='n(input()) lat = linear every(list, set a = linear every(list, set lat (l) = main(); lat = (l) = main(); lat = (l) = main(); lat = linear every(list, set lat = lint(list, set)(list</pre>	⊙ Viranikasi Varishel ¶		
			-		
			• • • •		
	SUBMIT	>>>			
	JOHNT				
		This (1473000)	•		

Gambar 22 Tampilan Input dari file text

Tahapan penggunaan input dari file text, antara lain:

 Membuat *file text* (*.txt) dengan bantuan Notepad, berikan nama *file* tersebut dengan "InputLinearSearch". File ini tersedia juga dalam folder Panduan pada CD program dan dapat disalin ke komputer. Isi file text dapat dilihat pada gambar 23.

InputLinearSearch.txt - Notepad					
File	Edit	Format	View	Help	
3					
1					
2					
3					
2					
	0	1 02 1	·		



ii. Tekan tombol memilih "…" untuk memilih *file text* InputLinearSearch seperti pada gambar 24.

Memilih File :	
	SURMI
	SUBINIT

Gambar 24 Tampilan Pilih File text

iii. Tekan open untuk memasukkan file InputLinearSearch seperti pada gambar 25.



Gambar 25 Tampilan Open File text

iv. Kemudian tekan tombol "submit" untuk melanjutkan kode program seperti pada gambar 26. Aplikasi PITON akan mengeluarkan *output* hasil kode program pada kolom ekspansi *output* seperti pada gambar 27.

PITON - PYTHON INTEGRATED WORKSP	ACE AND VISUALIZATION	© i -	б×
	× 🗈 🖪 🛧 🏲	Stop Visualisasi	
Import deri file test (*.tst) Memilih File : C/USers/Elvina/Desktop//Sour	<pre>idef linear_search(lat, search): for in range(0, len(lat), l): if (lat[i] == search): return 0 return 0</pre>		♥ Visualizari Viriabel
	Output	Еггог	
SUBART	>>>	»»	A
	~	4	<u> </u>

Gambar 26 Tampilan hasil Open File text



Gambar 27 Tampilan hasil output

c. Input sekaligus



Gambar 28 Tampilan Input sekaligus

Memasukkan *input* untuk kode program dengan memasukkan seluruh *input* yang dibutuhkan kode program seperti pada gambar 28. *Input* dimasukkan dalam *text box* seperti pada gambar 29. Kelebihan "*Enter*" saat memasukkan *input* untuk kode program, membuat input untuk kode program dianggap tidak *valid*.

🕑 Input	Input sekaligus sebelum progr Input :	am dijalankan
	A Source Teel	SUBMIT

Gambar 29 Tampilan Input sekaligus

Tahapan penggunaan input sekaligus, sebagai berikut:

i. Masukkan *input* seperti pada gambar 30.



Gambar 30 Tampilan setelah mengetik input sekaligus

ii. Kemudian tekan tombol "submit" untuk melanjutkan kode program. Aplikasi
 PITON akan mengeluarkan *output* hasil kode program pada kolom ekspansi
 output seperti pada gambar 31.



Gambar 31 Tampilan hasil output

III.2 Penggunaan PITON dengan fitur visualisasi

Penggunaan PITON dengan fitur visualisasi akan menampilkan panel visualisasi di sebelah kanan source code editor. Pada panel visualisasi akan ditunjukkan variabel yang dipakai oleh program dan juga perubahan nilai selama program berjalan. Fitur visualisasi juga akan menunjukkan posisi instruksi/perintah dalam program yang sedang dijalankan. Tahapan penggunaan fitur visualisasi adalah :

1. Berikan *checklist* pada *checkbox* visualisasi, kemudian jalankan kode program dengan menekan tombol "Run" seperti pada gambar 32.

	Visuali	sasi	Checkbox Visualisasi
	Run		Run



Gambar 32 Tampilan dengan fitur Visualisasi

- Pilihlah terlebih dahulu cara *input* yang diinginkan untuk memasukkan *input*. Terdapat
 3 jenis cara *input* pada PITON, yaitu:
 - a. Input Bertahap

Memasukkan input untuk kode program secara bertahap seperti pada gambar 33. PITON akan meminta input kembali jika dibutuhkan setelah input yang dimasukkan sebelumnya berhasil diproses seperti pada gambar 34.

PITON - PYTHON INTEGRATED WORKS	ACE AND VISUALIZATION		$\mathbb{Q} \mid i \mid = - \mathbb{P} \mid \times$
	× 🗈 🖪 🔸 🏞		$00 \ C \ \leftarrow \ \rightarrow \mathbf{X}$
	<pre>idef linear search(lst, search):</pre>	str(i+1)) Pilih Tipe Input Pilih Tipe Input Input dari file text Input stealgus	O Vinuiteai Vunibel
Input:	1		
	I = 1	▼	
	Output Ot Error		
SUBMIT	>>> ^ >>>	^	
		• •	

Gambar 33 Visualisasi dengan Input bertahap

🕑 Input	Input bertahap sesuai alur program Input :
	SUBMIT

Gambar 34 Tampilan input bertahap

Tahapan penggunaan *input* bertahap, antara lain:

i. Masukkan angka **3** kemudian tekan tombol "Submit" seperti pada Gambar 35.

😰 PITON - P'	YTHON INTEGRATED WORKSP	ACE AND VISUALIZATION			© i - ♂ ×
4 0 D	B h i	X 🗈 🖪 🛧 🏞			$-\bullet \rightarrow \square \ C \leftarrow \rightarrow \mathbf{X}$
S source Code S source Code S source Code	ault inearSearch.py p sessai alur program	<pre>1 def lies our ramp(0, search); for i n ramp(0, sea(int), for i n ramp(0, sea(int), if (int(i) == earch i; return 0 print("Nilsi tids ditemular return -1 return -1 print("Nilsi tids ditemular return -1 return -1 return -1 return -1 if or i n range(0, n, 1); ist = [Nong)*n if or i n range(0, n, 1); ist = ist(imp(1)) s = h = ist(imp(1), search); ist = main(); main()</pre>	l): kan pada posisi " + str(i+1)) n.")		Vrussien Vrussier Vrussier
3				v	
		Output	Error		
	SUBMIT	>>>	* »»>	▲	
			· ()	*	

Gambar 35 Tampilan input bertahap (1)

ii. Masukkan angka 1 kemudian tekan tombol "Submit" seperti pada Gambar 36.

PITON - PYTHON INTEGRATED WORKS	ACE AND VISUALIZATION	© i = ♂ ×
	× 🗈 🖻 🛧 🏞	
Imput bertahap seusai akur program Imput bertahap seusai akur program Imput bertahap seusai akur program	<pre>idef linear search1st, search1: for i in range(0. hen(lat), 1): if (lat(1) == search 1: return 0 print("Niki disemukan.") detrain(i): n= in(input()) lat = [Nong]*n lat = [Nong]*n lat = [Nong]*n lat [1]=in(input()) lat = [Nong]*n lat = [Nong]*n lat (i=name_ent(), n, 1): lat = [Nong]*n lat (i=name_ent(), n, 2): lat = [Nong]*n lat (i=name_ent(), n, 2): lat = [Nong]*n lat (i=name_ent(), n, 2): lat (i=name_ent(), n, search) lat (i=name_ent(), n, 2): main() lat (i=name_ent(), n, 2): lat (i=name_ent(), n, 2):</pre>	1 C Youlder Worker
	Output Error	
SUBMIT	>>> *>> *>> *>	
3		

Gambar 36 Tampilan input bertahap (2)

iii. Masukkan angka **2** kemudian tekan tombol "Submit" seperti pada Gambar 37.



Gambar 37 Tampilan input bertahap (3)

iv. Masukkan angka 3 kemudian tekan tombol "Submit" seperti pada Gambar 38.



Gambar 38 Tampilan input bertahap (4)

v. Masukkan angka **2** kemudian tekan tombol "Submit" seperti pada Gambar 39.



Gambar 39 Tampilan input bertahap (5)

vi. Aplikasi PITON akan mengeluarkan *output* hasil kode program pada kolom ekspansi output dan menjalankan visualisasi pada panel visualisasi variabel seperti pada gambar 40.



Gambar 40 Tampilan output (visualisasi)

b. *Input* dari *File Text* (*.txt)

PITON - PYTHON INTEGRATED WORKSP	ACE AND VISUALIZATION	$\mathbb{O} \mathbf{i} = \mathbf{a} \times$
	× • • •	$ \triangleright 00 \ C \leftarrow \rightarrow \mathbf{X}$
Population Input deil file text (".tot) Memilih File: m	<pre>ider linear_search(ist, search): for in range(0, ben(ist), 1): if (lat(1) == search): return 0 print("Niai tidak ditemukan.") isturn 0 print("Niai tidak ditemukan.") isturn 0 ist in range(0, cp. 1): for in range(0, cp. 1): input betabap @ Input define text Input setaligus input setaligus input setaligus</pre>	Viruuliovi Vorabel
	Output Error	
SUBMIT	>>> ^ >>> ^ A	

Gambar 41 Visualisasi dengan Input dari file text

Memasukkan *input* untuk kode program melalui *file text* yang sudah dirancang seperti pada gambar 41. Untuk itu akan digunakan *file text* "InputLinearSearch" yang sama seperti pada III.1, seperti ditampilkan pada Gambar 42.

🥘 InputLinearSearch.txt - Notepad				
File	Edit	Format	View	Help
3				
1				
2				
3				
2				



Tahapan penggunaan input dari file text, antara lain:

Setelah file "InputLinearSearch" disiapkan, tekan tombol memilih "…" seperti pada gambar 43 untuk memilih *file text* InputLinearSearch seperti pada gambar 44.



Gambar 43 Memilih file text



Gambar 44 Tampilan Open file text

ii. Tekan *open* untuk memasukkan *file* InputLinearSearch, sehingga ditampilkan seperti pada gambar 45.

D PITON - PYTHON INTEGRATED WORKSP	ACE AND VISUALIZATION			© i - ♂ ×
	× D 🖪 🛧 🏞		I	$ 00 C \leftarrow \rightarrow \mathbf{X} $
Imput dari file text (*.txt) Memilih File ; C(Msers/Ebrina/Desktop)/Sour	<pre>ca co co</pre>			O Vranificai Variabel
	Output	Error		
SUBMIT	***	>>> A		
	v	4 ×		

Gambar 45 Tampilan hasil memilih file text

 iii. Kemudian tekan tombol "submit" untuk melanjutkan kode program. Aplikasi PITON akan mengeluarkan *output* hasil kode program pada kolom ekspansi *output* dan menjalankan visualisasi pada kolom ekspansi visualisasi variabel seperti pada gambar 46.

PITON - PYTHON INTEGRATED WORKSPA	CE AND VISUALIZATION	$\mathbb{O} \mathbf{i} = -\sigma \times$
	X 🗈 🖻 🛧 🏓	$ \triangleright 00 C \leftarrow \rightarrow \mathbf{X}$
Opport dan file text (*.td) Memilih File ; CKUsen/Ebrina/Desktop/Sour	<pre>1 def limer searchlit, searchl: 1 def limer searchlit, searchl: 1 f list(] == searchlit; 1 f list(] == searchlit; 1 f return 0 2 print("Nilaitiditemukan.") 2 return -1 3 def m int(input()) 10 list = [Wong]*n 11 for ini range(0, n, 1): 12 list[1=int(input()) 13 search = int(input()) 14 search = int(input(); 15 search = int(input(); 16 m amag ==="main_": 16 main() 17 main() 18 search = int(input(); 19 search = int(input(); 10 search</pre>	A Viole None None None None None None None Non
	Output Strror	
SUBMIT 3 1 2 3 2	>>> 3 1 2 3 2 Hilai ditemukan pada posisi 2 •	

Gambar 46 Tampilan hasil output (visualisasi)

c. Input sekaligus



Gambar 47 Visualisasi dengan Input sekaligus

Memasukkan *input* untuk kode program dengan memasukkan seluruh *input* yang dibutuhkan kode program seperti pada gambar 47. Kelebihan "*Enter*" saat memasukkan *input* untuk kode program, membuat input untuk kode program dianggap tidak *valid*. Input sekaligus dimasukkan melalui bagian seperti pada gambar 48.

🕑 Input	Input sekaligus sebelum progr Input :	ram dijalankan
	4	
		SUBMIT

Gambar 48 Tampilan input sekaligus

Tahapan penggunaan *input* sekaligus, antara lain:

i. Masukkan *input* sebagai berikut seperti pada gambar 49:



Gambar 49 Tampilan setelah input sekaligus diketik

ii. Kemudian tekan tombol "submit" untuk melanjutkan kode program. Aplikasi
 PITON akan mengeluarkan *output* hasil kode program pada kolom ekspansi
 output dan menjalankan visualisasi pada kolom ekspansi visualisasi variabel
 seperti pada gambar 50.

PITON - PYTHON INTEGRATED WORKSP	ACE AND VISUALIZATION		$ \mathbf{u} _{\mathbf{u}} = \mathbf{e} \mathbf{x} $
	X 🗈 🖪 🛧 🏕		$ \triangleright \Box C \leftarrow \rightarrow \mathbf{X}$
O source Com O novot sekaligus sebelum program dijalankan Input :	<pre>idef liner search(ist, search):</pre>	: pada posisi " + str(i+1)))	1 0 1st 1 2 1st 1 2 1st 1 3 search 2
	Output	C Error	
General Control of Con	>>> 3 1 2 3 2 Nilai ditemukan pada posisi 2	>>> (×

Gambar 50 Tampilan output (visualisasi)

3. Pengaturan Visualisasi kode program

Jika *checkbox* visualisasi diaktifkan sebelum menjalankan kode program, maka PITON akan menampilkan fitur tambahan pada *command toolbar* untuk pengaturan visualisasi seperti pada gambar 51. Fitur tambahan dapat digunakan untuk mengatur animasi pada saat visualisasi.

Gambar 51 Fitur tambahan untuk Visualisasi pada Command toolbar

Simbol	Nama Simbol	Aksi
	Pause	Menunda visualisasi
÷	Previous	Memundurkan kondisi visualisasi ketahap sebelumnya
\rightarrow	Next	Memajukan kondisi visualisasi ketahap setelahnya
C	Reset	Mengembalikan kondisi visualisasi ketahap awal
	Speed Bar	Mengatur kecepatan animasi visualisasi. Semakin ke kiri, semakin bertambah kecepatan visualisasi dan sebaliknya. Pengaturan <i>speed bar</i> hanya dapat dilakukan jika visualisasi berada pada kondisi tahap awal.
	Play	Menjalankan animasi untuk visualisasi
×	Stop	Menghentikan visualisasi dan eksekusi

Cobalah fitur visualisasi dengan memilih simbol sesuai urutan berikut:

Beberapa fitur tambahan yang dapat digunakan, antara lain:

1. Menghapus file dari aplikasi PITON

Menghapus file dapat dilakukan dengan menggunakan fitur "delete"



a. Pilih *file* yang akan dihapus dari aplikasi PITON seperti pada Gambar 52.



Gambar 52 Memilih file

b. Tekan tombol "delete", aplikasi akan memunculkan jendela *pop up* berupa pertanyaan seperti pada gambar 53. Tekan tombol "OK" untuk menghapus *file*. *File* tidak akan terhapus dari memori komputer, melainkan *file* hanya terhapus dari aplikasi PITON.



Gambar 53 Konfirmasi untuk menghapus

c. File akan hilang dari tampilan aplikasi PITON seperti pada gambar 54.



Gambar 54 Tampilan setelah file dihapus

2. Membuka kembali kode program (*.py)

Membuka kode program dapat dilakukan dengan menggunakan fitur Get File.

a. Tekan tombol "Get File"



b. Pilih *file* dengan ekstensi *.py yang akan dieksekusi maupun dimanipulasi.
 Pada kasus ini pilihlah *file* "LinearSearch.py" seperti pada gambar 55.

PITON - PYTHON INTEGRATED WORKSP		© i - ♂ ×
G 🗊 🖻 🖻 🖿 İ	🗙 🖹 🗈 🍝	
Image: Open	> SourceCodeTest v b Search SourceCodeTest p	≪ Visualisat
Ciganize ~ New folder Organize ~ New folder Properties # Din # Din # Din # Din # Din # Din # Protoclast Din # Din # D	Python File (".py)	a Vinide
	Output	* •
	LITTIN (171 LIVE)	

Gambar 55 Tampilan memilih File untuk dibuka

c. Tekan tombol "Open", maka *file* LinearSearch akan muncul pada ekspansi *Source Code* seperti gambar 56.



Gambar 56 Tampilan setelah Open file

3. Membuka *folder* berisi beberapa kode program (*.py)

Membuka *folder* berisi beberapa kode program dapat dilakukan dengan menggunakan fitur *get folder*.

a. Tekan tombol "get folder"



b. Pilih *folder* yang akan ditampilkan isi *file*-nya. Pilihlah *folder* yang berisi
 LinearSearch.py seperti pada gambar 57. Tekan "Select Folder" untuk
 menampilkan semua *file* dengan ekstensi (*.py) pada *folder* tersebut.



Gambar 57 Tampilan memilih folder

c. *File* akan ditampilkan sebagai *file* dari direktori *folder* yang dipilih seperti pada gambar 58.



Gambar 58 Tampilan setelah membuka folder

d. Jika *file* yang terdapat pada *folder* sudah ada pada aplikasi PITON, maka akan muncul jendela *pop up* yang menyatakan bahwa *file* tersebut sudah ada seperti pada gambar 59.



Gambar 59 Konfirmasi file sudah ada

e. *Folder* yang dipilih tetap akan ditampilkan, namun *file* yang sudah ada tidak akan ditampilkan lagi seperti pada gambar 60.



Gambar 60 Tampilan Semula