



ANALISIS PENGARUH PENGGUNAAN TANDA BACA DALAM PENGGUNAAN BAHASA PEMROGRAMAN PYTHON

Jonathan Teguh Samuel Kaeng¹, Muhammad Fakhri Anshary Yusaf², Fidelia Hahas Asabela³, Afifah Naila Azzahra⁴, Muhammad Ibrahim Khoirudin⁵, Natalia Desy Angraeni⁶

Universitas Pembangunan Nasional 'Veteran' Jawa Timur, Indonesia

24081010140@student.upnjatim.ac.id¹, 24081010124@student.upnjatim.ac.id²,

24081010113@student.upnjatim.ac.id³, 24053010050@student.upnjatim.ac.id⁴,

24011010158@student.upnjatim.ac.id⁵, nataliadesy2412@gmail.com⁶

Abstrak:

Python merupakan salah satu jenis dari bahasa pemrograman yang sering digunakan oleh para programmer. Bahasa pemrograman Python juga dapat digunakan untuk banyak hal. Namun, sering kali dalam proses pengodeannya ditemukan kesalahan penulisan tanda baca. Sehingga hasil dari pengodean yang telah dilakukan menjadi error. Karya Tulis Ilmiah ini dibuat untuk mempelajari tentang, seberapa pentingnya tanda baca dalam pengodean menggunakan bahasa pemrograman Python. Dalam mempelajarinya kami menggunakan metode penelitian Studi Kasus Kualitatif, dengan disertai 10 contoh studikusus yang kami buat. Dimana, hasil dari penelitian ini menunjukkan bahwa seorang programmer salah dalam memilih atau menempatkan tanda baca, atau bahkan lupa untuk menambahkan tanda baca yang seharusnya, maka hasil dari kode yang telah dibuat akan menjadi sulit dipahami atau bahkan menjadi error. Selain itu, tanda baca juga tidak boleh asal dalam penggunaannya karena sudah ada tata cara dalam penulisan tanda baca saat menggunakan bahasa pemrograman Python. Sehingga kesimpulan dari penelitian ini adalah bahwa tanda baca sangatlah berpengaruh terhadap bahasa pemrograman Python. Setiap orang harus memperhatikan kapan waktu yang tepat dan tanda baca apa yang tepat untuk digunakan, Karena setiap bahasa pemrograman mempunyai ketentuannya masing-masing dalam penggunaan tanda baca. Para programmer juga harus memperhatikan penggunaan tanda baca agar tidak terjadi error saat kode yang telah dibuat dijalankan.

Kata kunci: Tanda baca, bahasa pemrograman Python, analisis penggunaan tanda baca, Pengaruh tanda baca, Python programming, Sintaksis Python.

Abstrack

Python is one type of programming language that is often used by programmers. The Python programming language can also be used for many things. However, in the coding process, punctuation errors are often found. So that the results of the coding that has been done become an error. This Scientific Paper was created to study how important punctuation is in coding using the Python programming language. In studying it, we use the Qualitative Case Study research method, accompanied by 10 examples of case studies that we made. Where, the results of this study show that a programmer is wrong in choosing or placing punctuation, or even forgets to add the punctuation that should be, then the results of the code that has been created will be difficult to understand or even become an error. In addition, punctuation should not be used carelessly because there are already procedures for writing punctuation when using the Python programming language. So the conclusion of this study is that punctuation is very influential on the Python programming language. Everyone must pay attention to when is the right time and what punctuation is right to use, because each programming language has its own provisions in the use of punctuation. Programmers must also pay attention to the use of punctuation so that errors do not occur when the code they have created is run.

Keywords: Punctuation, Python programming language, analysis of punctuation usage, Influence of punctuation, Python programming, Python syntax.

Corresponding: Jonathan Teguh Samuel Kaeng
E-mail: 24081010140@student.upnjatim.ac.id



PENDAHULUAN

Dalam era digital saat ini, bahasa pemrograman menjadi salah satu elemen kunci dalam berbagai bidang, mulai dari pengembangan perangkat lunak hingga analisis data. Salah satu bahasa pemrograman yang semakin populer adalah Python, yang dikenal dengan sintaksnya yang sederhana dan mudah dipahami (Fadilah Aulia dan Yahfizham, 2024). Namun, di balik kemudahan ini, penggunaan tanda baca yang tidak tepat dapat menyebabkan kesalahan fatal dalam pemrograman. Hal ini menimbulkan isu global mengenai pentingnya pemahaman mendalam terhadap sintaks dan tanda baca dalam bahasa pemrograman.

Secara spesifik, Python memiliki aturan tanda baca yang ketat, seperti penggunaan titik dua (:), tanda kurung, dan indentasi. Kesalahan dalam penerapan aturan ini tidak hanya dapat menyebabkan program gagal berjalan, tetapi juga menyulitkan debugging. Sebuah studi menunjukkan bahwa sebagian besar kesalahan pemrograman terjadi akibat kurangnya pemahaman terhadap penggunaan tanda baca (Sabban, 2024). Oleh karena itu, analisis mendalam mengenai pengaruh tanda baca dalam penggunaan Python menjadi sangat penting.

Perkembangan teknologi global telah mendorong adopsi Python dalam berbagai sektor, termasuk pendidikan, industri, dan penelitian ilmiah. Sebagai contoh, Python digunakan untuk analisis data besar, pengembangan kecerdasan buatan, dan pengelolaan basis data. Dalam konteks global, pentingnya efisiensi dan akurasi dalam pemrograman menjadi sorotan utama (Nazar, 2024). Kesalahan sintaks, terutama dalam penggunaan tanda baca, dapat memengaruhi produktivitas dan kualitas aplikasi yang dikembangkan.

Dalam konteks lokal, pengajaran Python di perguruan tinggi dan pelatihan industri menunjukkan bahwa banyak pemula kesulitan memahami aturan tanda baca dalam bahasa ini (Mohammad, 2018). Salah satu masalah utama adalah minimnya sumber daya pembelajaran yang secara eksplisit membahas penggunaan tanda baca. Penelitian menunjukkan bahwa 60% kesalahan pemrograman Python di lingkungan pendidikan terjadi akibat kesalahan pada sintaks dasar (Muhammad et al, 2020).

Urgensi penelitian ini didorong oleh meningkatnya adopsi Python sebagai bahasa pemrograman utama. Python sering digunakan dalam pengembangan aplikasi berbasis algoritma kompleks, seperti Natural Language Processing (NLP) (Nova, 2024). Namun, kurangnya perhatian terhadap detail seperti tanda baca dapat menjadi hambatan dalam mencapai tujuan tersebut. Sebagai contoh, implementasi NLP sering kali gagal akibat kesalahan sintaks kecil yang sulit dideteksi (Ledjap et al., 2024).

Studi sebelumnya telah mengeksplorasi berbagai aspek pembelajaran Python, termasuk pendekatan logika algoritma (Muhammad et al, 2020) dan penerapan Python dalam analisis data besar (Ua, 2023). Namun, hanya sedikit penelitian yang secara spesifik membahas pengaruh penggunaan tanda baca. Buku "Langkah Awal Menguasai Bahasa Pemrograman Python" oleh

Sutriawan (2024) memberikan panduan dasar, tetapi tidak cukup mendalam dalam membahas peran tanda baca dalam sintaks Python. Penelitian ini bertujuan untuk mengisi kesenjangan tersebut.

Penelitian ini menawarkan pendekatan baru dengan fokus pada analisis empiris penggunaan tanda baca dalam Python. Berbeda dengan studi sebelumnya, penelitian ini tidak hanya membahas kesalahan umum, tetapi juga dampaknya terhadap efisiensi dan keandalan program. Selain itu, penelitian ini akan mengeksplorasi bagaimana penggunaan alat seperti Google Colab dapat membantu pemrogram memahami pentingnya tanda baca dalam pengkodean (Nazar, 2024).

Penelitian ini bertujuan untuk Menganalisis pengaruh penggunaan tanda baca dalam keberhasilan eksekusi program Python. Mengidentifikasi jenis kesalahan tanda baca yang paling umum dilakukan oleh pemrogram. Mengembangkan rekomendasi pembelajaran untuk meningkatkan pemahaman tentang tanda baca dalam Python.

METODE PENELITIAN

Penelitian ini termasuk ke dalam kategori penelitian kualitatif dengan pendekatan studi kasus. Penelitian kualitatif bertujuan untuk memahami fenomena secara mendalam dalam konteks alami (Creswell, 2014). Studi kasus individu digunakan untuk mengeksplorasi pengaruh penggunaan tanda baca dalam bahasa pemrograman Python. Dengan metode ini, peneliti dapat mengeksplorasi detail dari setiap kasus yang diamati, memberikan pemahaman yang mendalam mengenai permasalahan yang dikaji.

Pendekatan yang digunakan adalah studi kasus. Pendekatan ini dipilih karena memungkinkan analisis rinci dari setiap kasus individu yang mencakup berbagai permasalahan dan solusi dalam penulisan kode Python. Penelitian dilakukan menggunakan 10 studi kasus yang dibuat secara khusus, yang mencerminkan berbagai situasi kesalahan tanda baca pada pengodean.

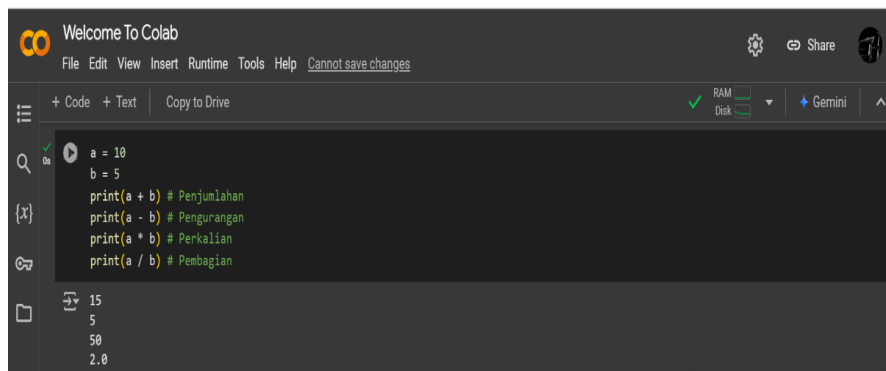
Analisis data dilakukan secara deskriptif, dengan menganalisis hasil dari setiap studi kasus berdasarkan kesalahan tanda baca yang ditemukan dan dampaknya terhadap eksekusi kode. Data dianalisis menggunakan teknik analisis tematik untuk mengidentifikasi pola atau tema utama yang muncul dari kasus-kasus tersebut (Braun & Clarke, 2006). Hasil analisis kemudian disajikan dalam bentuk narasi dan visualisasi, seperti tabel dan grafik untuk memberikan gambaran yang jelas tentang pengaruh tanda baca dalam pemrograman.

Data dikumpulkan melalui simulasi langsung dari pengodean menggunakan Python, yang dilakukan di platform Google Colab. Setiap kasus melibatkan pembuatan kode dengan variasi kesalahan tanda baca, kemudian mengamati dan mencatat dampaknya terhadap hasil eksekusi. Teknik pengumpulan data ini memungkinkan peneliti memperoleh data empiris secara langsung dari studi kasus.

HASIL DAN PEMBAHASAN

Programming merupakan fokus utama dalam informatika. Ketelitian dalam menulis tanda baca pada bahasa pemrograman harus terus ditingkatkan karena bahasa pemrograman tidak hanya kemampuan dalam logika saja, tetapi juga ketelitian dalam melihat kode yang ditulis.

Gambar 1.



The screenshot shows the Google Colab interface. The code cell contains the following Python code:

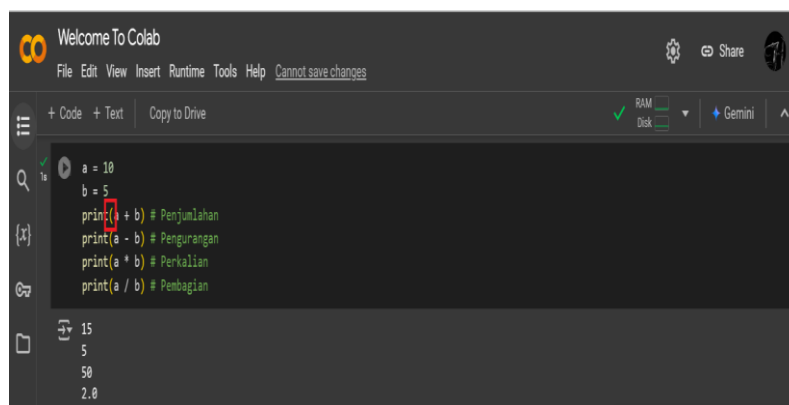
```
a = 10
b = 5
print(a + b) # Penjumlahan
print(a - b) # Pengurangan
print(a * b) # Perkalian
print(a / b) # Pembagian
```

The output cell shows the results of the calculations:

```
15
5
50
2.0
```

Dalam Gambar 1, menunjukkan bahwa kode yang ditulis menghasilkan jawaban yang sesuai dan tidak terjadi kesalahan. Hal itu disebabkan karena penggunaan tanda baca seperti tanda kurung () harus diperhatikan dalam bahasa pemrograman.

Gambar 2.



The screenshot shows the Google Colab interface. The code cell contains the following Python code:

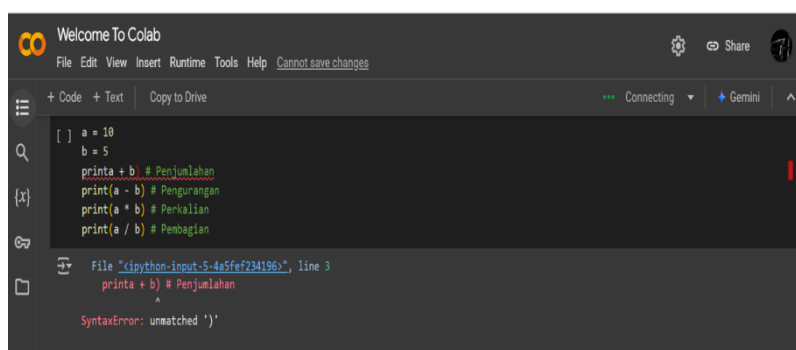
```
a = 10
b = 5
print(a + b) # Penjumlahan
print(a - b) # Pengurangan
print(a * b) # Perkalian
print(a / b) # Pembagian
```

The output cell shows the results of the calculations:

```
15
5
50
2.0
```

A red box highlights the opening parenthesis in the first `print(a + b)` statement.

Gambar 3.



The screenshot shows the Google Colab interface. The code cell contains the following Python code:

```
[ ] a = 10
b = 5
printa + b) # Penjumlahan
print(a - b) # Pengurangan
print(a * b) # Perkalian
print(a / b) # Pembagian
```

The output cell shows the error message:

```
File "c:\python-input-5-4a5fef234196", line 3
printa + b) # Penjumlahan
          ^
SyntaxError: unmatched ')'
```

Jika tanda kurung pada kotak merah dihilangkan seperti pada Gambar 2, maka akan menghasilkan kode *error* atau kesalahan pada kode seperti pada Gambar 3. Tanda baca seperti tanda kurung () harus diperhatikan dengan baik agar kode dapat menghasilkan jawaban yang akurat tanpa kesalahan.

Studi Kasus 2



```
Welcome To Colab.ipynb
File Edit View Insert Runtime Tools Help

+ Code + Text

▶ oktal = "755"
  desimal = int(oktal, 8)
  biner = bin(desimal)[2:]

  hak_akses = {
    '0': '---',
    '1': '--X',
    '2': '-W-',
    '3': '-WX',
    '4': 'r--',
    '5': 'r-X',
    '6': 'rW-',
    '7': 'r-WX'
  }

  akses = " ".join([hak_akses[digit] for digit in oktal])

  print(f'Oktal {oktal} = Biner {biner}')
  print(f'Hak Akses: {akses}')

Oktal 755 = Biner 111101101
Hak Akses: r-WX r-X r-X
```

Gambar 4.



```
Welcome To Colab.ipynb
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

▶ oktal = "755"
  desimal = int(oktal, 8)
  biner = bin(desimal)[2:]

  hak_akses = {
    '0': '---',
    '1': '--X',
    '2': '-W-',
    '3': '-WX',
    '4': 'r--',
    '5': 'r-X',
    '6': 'rW-',
    '7': 'r-WX'
  }

  akses = " ".join([hak_akses[digit] for digit in oktal])

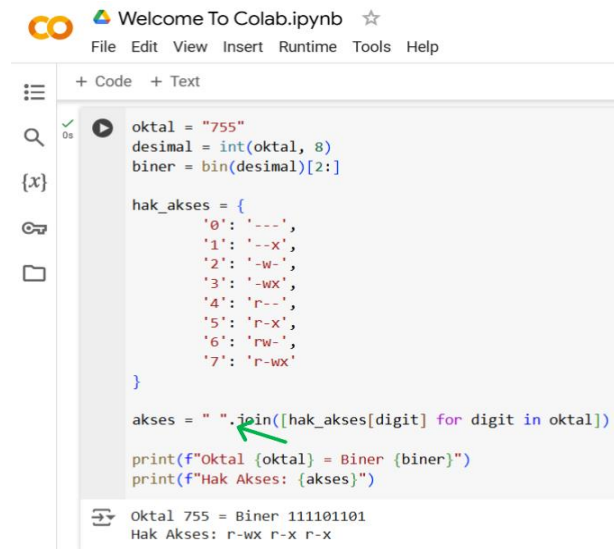
  print(f'Oktal {oktal} = Biner {biner}')
  print(f'Hak Akses: {akses}')

File "<ipython-input-17-cb3b56264953>", line 5
    hak_akses =
    ^
SyntaxError: invalid syntax
```

Gambar 5.

Tanda panah berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak keasalahan pada kode. Pada gambar 4 merupakan contoh dari penerapan tanda baca berupa kurung kurawal yang benar, sehingga kode dapat dijalankan tanpa *error*. Sedangkan pada Gambar 5, merupakan contoh apabila pemrogram lupa untuk menuliskan tanda baca kurung kurawal. Sehingga terjadi *error* pada saat kode dijalankan.

Studi Kasus 3



```
oktal = "755"
desimal = int(oktal, 8)
biner = bin(desimal)[2:]

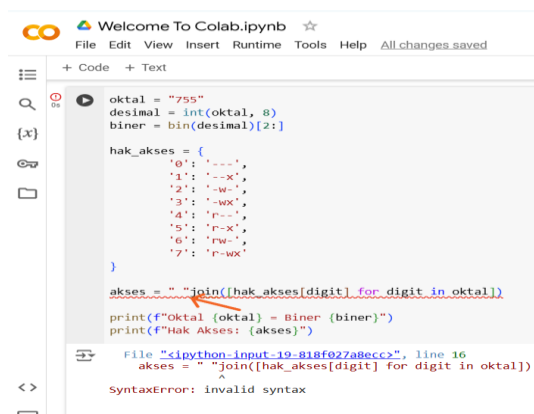
hak_akses = {
    '0': '---',
    '1': '--x',
    '2': '-w-',
    '3': '-wx',
    '4': 'r--',
    '5': 'r-x',
    '6': 'rw-',
    '7': 'rwx'
}

akses = " ".join([hak_akses[digit] for digit in oktal])

print(f'Oktal {oktal} = Biner {biner}")
print(f'Hak Akses: {akses}")
```

Oktal 755 = Biner 11101101
Hak Akses: r-wx r-x r-x

Gambar 6.



```
oktal = "755"
desimal = int(oktal, 8)
biner = bin(desimal)[2:]

hak_akses = {
    '0': '---',
    '1': '--x',
    '2': '-w-',
    '3': '-wx',
    '4': 'r--',
    '5': 'r-x',
    '6': 'rw-',
    '7': 'rwx'
}

akses = " ".join([hak_akses[digit] for digit in oktal])

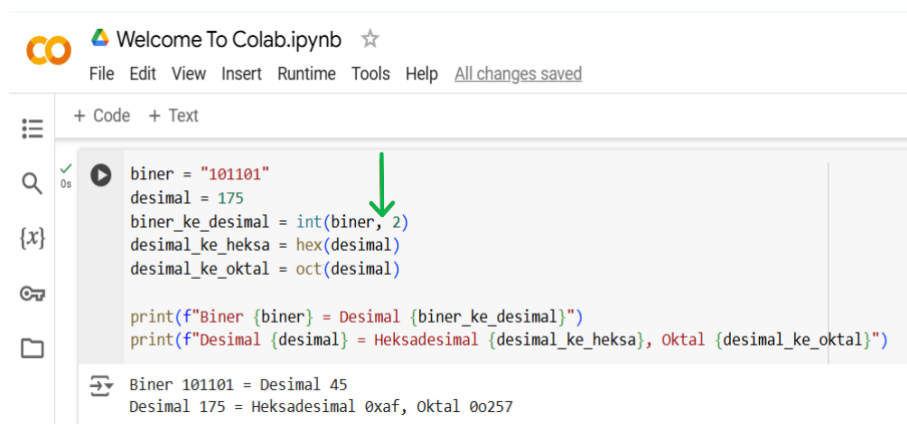
print(f'Oktal {oktal} = Biner {biner}")
print(f'Hak Akses: {akses}")
```

File "<ipython-input-19-818f027a8ecc>", line 16
akses = " ".join([hak_akses[digit] for digit in oktal])
SyntaxError: invalid syntax

Gambar 7.

Tanda panah berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Pada Gambar 6, merupakan contoh dari penggunaan tanda baca titik dengan tepat atau sesuai dengan aturan penulisan kode menggunakan bahasa Python. Sehingga kode pada Gambar 6, dapat terbaca saat dijalankan dan tidak *error*. Gambar 7 merupakan contoh pada saat pemrogram lupa untuk menambahkan tanda baca titik pada kode yang ditulis. Sehingga saat kode pada Gambar 7 dijalankan, hasilnya menjadi *error*.

Studi Kasus 4



Gambar 8.

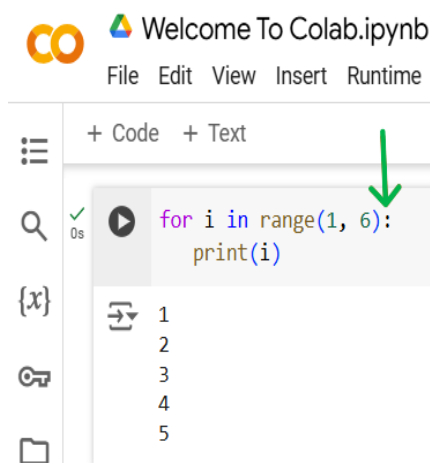


Gambar 9.

Tanda panah berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Gambar 8 menampilkan penulisan kode dengan benar, programmer tidak lupa menuliskan tanda baca koma setelah kata biner. Sehingga kode pada Gambar 8 dapat terbaca dan berjalan sesuai dengan tujuan pengodean. Untuk Gambar 9 merupakan Gambar yang menampilkan penulisan kode yang kurang tepat, karena programmer lupa menuliskan tanda baca koma setelah kata biner. Sehingga kode pada Gambar 9 menjadi *error* saat dijalankan.

programmer atau orang lain yang akan menggunakan hasil dari pengodean tersebut.

Studi Kasus 6



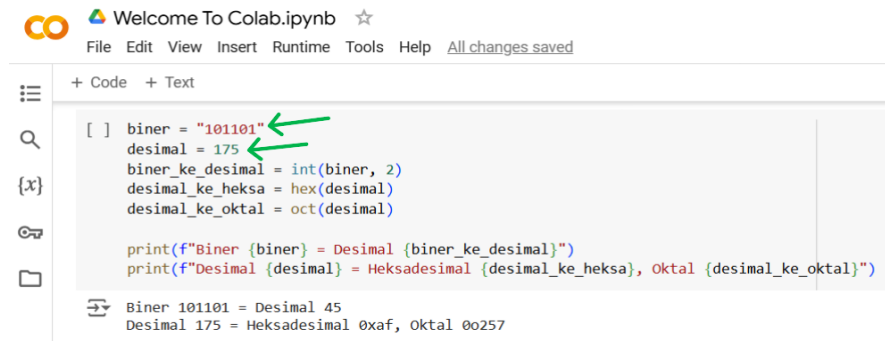
Gambar 10.



Gambar 11

Tanda panah berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Gambar 10 merupakan Gambar yang menampilkan penulisan kode menggunakan bahasa python dengan benar. Dimana programmer tidak lupa untuk menambahkan tanda baca titik dua setelah kode pada baris pertama selesai. Sehingga kode pada Gambar 10 dapat terbaca dan hasilnya sesuai dengan perintah yang diberikan pada baris kedua. Sedangkan pada Gambar 11, terlihat bahwa tidak ada tanda baca titik dua pada akhir penulisan kode di baris pertama. Karena tidak ada tanda baca titik dua tersebut, maka saat kode dijalankan hasilnya akan *error* dan tidak sesuai dengan perintah pada baris kedua.

Studi Kasus 7

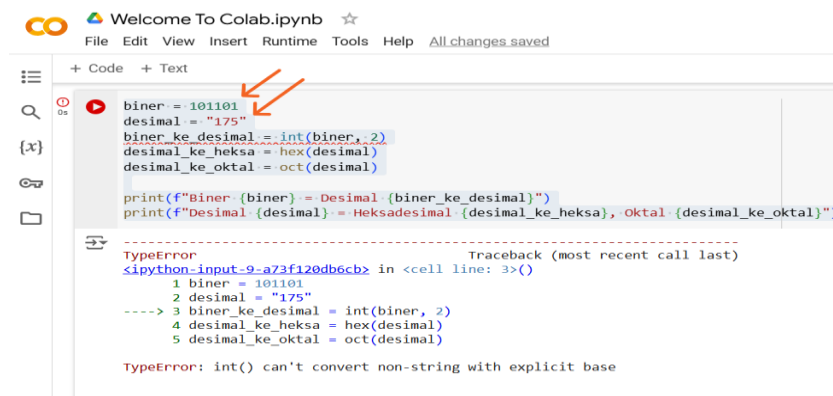


```
biner = "101101"
desimal = 175
biner_ke_desimal = int(biner, 2)
desimal_ke_heksa = hex(desimal)
desimal_ke_oktal = oct(desimal)

print(f"Biner {biner} = Desimal {biner_ke_desimal}")
print(f"Desimal {desimal} = Heksadesimal {desimal_ke_heksa}, Oktal {desimal_ke_oktal}")
```

Biner 101101 = Desimal 45
Desimal 175 = Heksadesimal 0xaf, Oktal 0o257

Gambar 12



```
biner = 101101
desimal = "175"
biner_ke_desimal = int(biner, 2)
desimal_ke_heksa = hex(desimal)
desimal_ke_oktal = oct(desimal)

print(f"Biner {biner} = Desimal {biner_ke_desimal}")
print(f"Desimal {desimal} = Heksadesimal {desimal_ke_heksa}, Oktal {desimal_ke_oktal}")
```

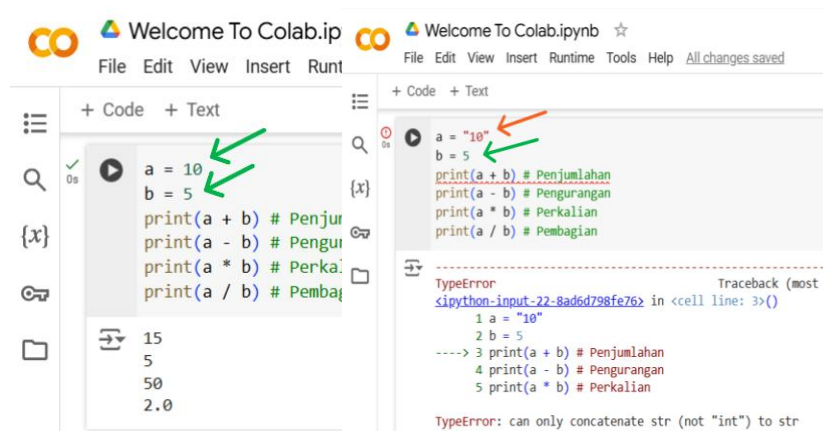
TypeError
Traceback (most recent call last)
<ipython-input-9-a73f120db6cb> in <cell line: 3>()
1 biner = 101101
2 desimal = "175"
----> 3 biner_ke_desimal = int(biner, 2)
4 desimal_ke_heksa = hex(desimal)
5 desimal_ke_oktal = oct(desimal)

TypeError: int() can't convert non-string with explicit base

Gambar 13

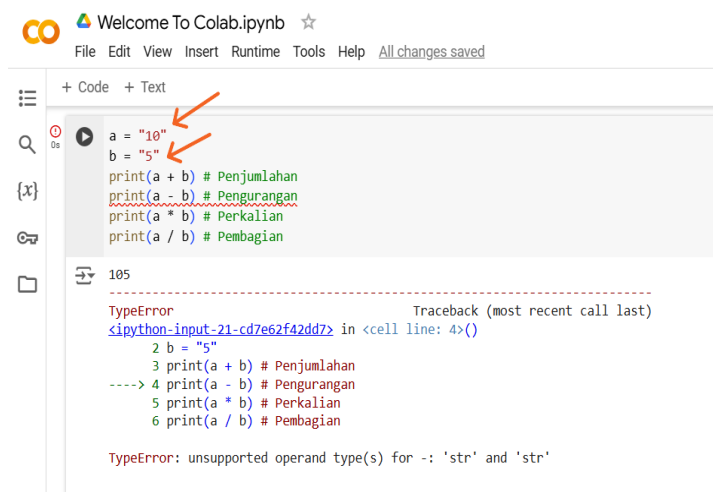
Tanda panah berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Berikutnya adalah contoh studi kasus dari penggunaan tanda baca petik dua. Gambar 12 menampilkan penggunaan yang benar atau penempatan yang tepat dari tanda baca petik dua saat menggunakan bahasa pemrograman Python. Sehingga kode dapat terbaca dan menampilkan hasil sesuai dengan perintah pada baris ke 7 serta 8. Sedangkan Gambar 13, menampilkan contoh yang salah dalam menggunakan atau menempatkan tanda baca petik dua pada saat menggunakan bahasa pemrograman Python. Sehingga kode menjadi *error* dan tidak bisa menampilkan hasil yang sesuai dengan perintah pada baris ke 7 dan 8.

Studi Kasus 8



Gambar 14

Gambar 15

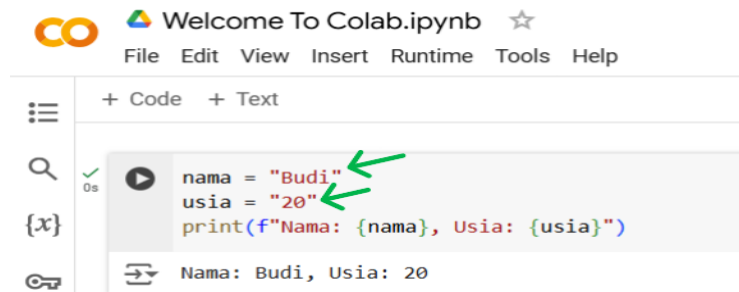


Gambar 16

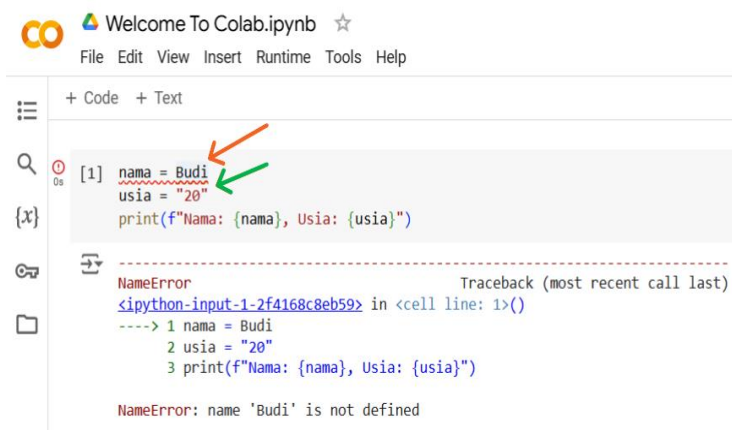
Tanda pada berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Gambar 14, merupakan Gambar yang menampilkan pengkodean yang benar pada studi kasus kali ini. Dimana pada bagian penulisan angka 10 dan 5, ditulis tanpa disertai tanda baca apapun. Sedangkan Gambar 15 dan Gambar 16, merupakan Gambar yang menampilkan pengkodean yang salah. Pada Gambar 15, dalam penulisan angka 10 programmer menambahkan tanda baca petik dua sebelum dan setelah angka 10 sehingga saat kode dijalankan hasilnya menjadi *error*. Sedangkan pada Gambar 16, kedua angka yakni 10 dan 5 dalam penulisannya sama-sama ditambahkan tanda baca petik dua setelah

dan sebelumnya oleh programmer. Sehingga saat kode dijalankan hasilnya akan menjadi *error* atau tidak sesuai dengan perintah yang diberikan pada baris ke 3 sampai 6.

Studi kasus 9



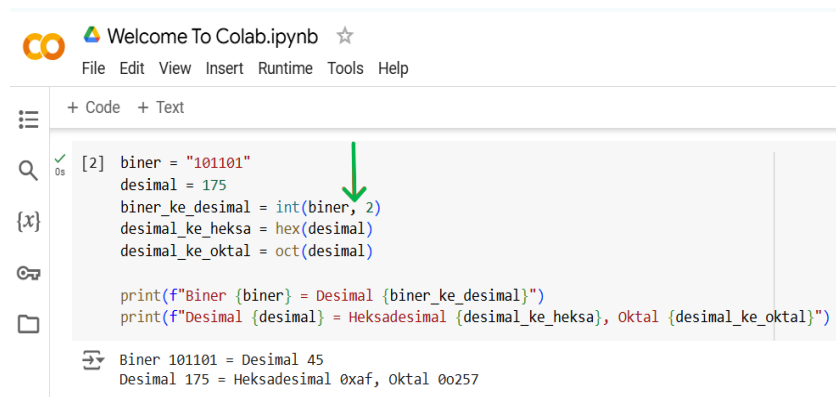
Gambar 17



Gambar 18

Tanda pada berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Pada studi kasus kali ini, Gambar 19 merupakan Gambar yang menampilkan pengodean yang benar. Dimana dalam penulisan nama dan usia, memang sudah seharusnya ditambahkan tanda baca petik dua pada sebelum dan sesudah menuliskan nama dan juga usia. Pada Gambar 18, ditampilkan apabila pengodean nama yang seharusnya sebelum dan sesudah penulisanannya diikuti tanda baca petik dua namun pada kali ini tidak ditambahkan. Maka hasil pengodean pada Gambar 18 akan menjadi *error*.

Studi Kasus 10



```
[2] biner = "101101"
    desimal = 175
    biner_ke_desimal = int(biner, 2)
    desimal_ke_heksa = hex(desimal)
    desimal_ke_oktal = oct(desimal)

    print(f"Biner {biner} = Desimal {biner_ke_desimal}")
    print(f"Desimal {desimal} = Heksadesimal {desimal_ke_heksa}, Oktal {desimal_ke_oktal}")

Biner 101101 = Desimal 45
Desimal 175 = Heksadesimal 0xaf, Oktal 0o257
```

Gambar 19



```
biner = "101101"
desimal = 175
biner_ke_desimal = int(biner: 2)
desimal_ke_heksa = hex(desimal)
desimal_ke_oktal = oct(desimal)

print(f"Biner {biner} = Desimal {biner_ke_desimal}")
print(f"Desimal {desimal} = Heksadesimal {desimal_ke_heksa}, Oktal {desimal_ke_oktal}")

File "<ipython-input-1-e34bf699c864>", line 3
    biner_ke_desimal = int(biner: 2)
                              ^
SyntaxError: invalid syntax
```

Gambar 20

Tanda pada berwarna hijau merupakan petunjuk dari penggunaan tanda baca yang benar. Sedangkan, tanda panah berwarna oranye merupakan petunjuk dari letak kesalahan pada kode. Studi kasus pada kali ini membahas tentang pemilihan tanda baca pada saat menggunakan bahasa pemrograman Python. Dimana pada Gambar ke 19, terlihat bahwa setelah kata biner ditambahkan tanda baca koma. Dimana pemilihan dan penggunaan tanda baca koma tersebut telah sesuai dengan aturan pada saat melakukan pemrograman menggunakan bahasa Python. Sehingga saat kode pada Gambar ke 19 dijalankan, hasilnya tidaklah *error*. Sedangkan pada Gambar ke 20, terlihat bahwa setelah kata biner pemrograman menuliskan tanda baca titik dua. Dimana, penggunaan tanda baca titik dua ini tidaklah sesuai dengan aturan saat melakukan pemrograman menggunakan bahasa Python. Karena adanya kesalahan dalam pemilihan kata, maka saat kode dijalankan hasilnya menjadi *error*.

Dari hasil pembahasan 10 studi kasus diatas, didapatkan bahwa tanda baca sangatlah berpengaruh terhadap hasil dari pemrograman menggunakan bahasa Python. Dimana jika seorang pemrogram salah dalam memilih atau pun salah dalam meletakkan tanda baca, maka hasil dari pemrograman yang telah dilakukan tidak akan sesuai dengan yang diinginkan atau bahkan menjadi *error*. Seorang pemrogram harus lebih teliti dalam menentukan atau memilih tanda baca. Karena, setiap tanda baca dalam bahasa pemrograman Python memiliki letak dan juga fungsi yang berbeda. Walaupun terdapat beberapa waktu dimana tanda baca tidak terlalu berpengaruh, namun tanda baca tetaplah sangat penting untuk diperhatikan. Hal ini karena selain agar kode dapat menghasilkan hasil sesuai yang diinginkan, namun juga agar hasil dari pengodean dapat dipahami oleh setiap pihak yang membutuhkannya dan bahkan dari pihak yang tidak ada hubungannya dengan pemrograman.

KESIMPULAN

Tanda baca berpengaruh dalam pemrograman, terutama dalam bahasa pemrograman. Setiap bahasa pemrograman memiliki standar penulisannya masing-masing, oleh karena itu setiap orang yang ingin menggunakan bahasa pemrograman haruslah mempelajari standar atau aturan penulisan dari bahasa pemrograman yang ingin di gunakan. Tujuan dari mempelajari terlebih dahulu standar atau aturan penulisan dari suatu bahasa pemrograman adalah agar saat melakukan pemrograman kita dapat mengetahui penempatan dan pemilihan tanda baca yang tepat. Sehingga, kode yang telah pengguna tuliskan dapat berjalan dengan lancar dan menampilkan hasil yang sesuai dengan kemauan pengguna.

DAFTAR PUSTAKA

- Braun, V., & Clarke, V. (2006). Menggunakan analisis tematik dalam psikologi. *Penelitian Kualitatif dalam Psikologi*, 3(2), 77-101.
- Creswell, J. W. (2014). *Desain Penelitian: Pendekatan Kualitatif, Kuantitatif, dan Metode Campuran*. Sage Publications.
- Fadilah Aulia Dan Yahfizham (2024), "Mengenal Bahasa Pemrograman Pada Algoritma Pemrograman", *Journal Of Informatics And Busines*, Vol. 1 No. 4: 223 – 228
- Ledjap, A. M. B., Rochmawati, F. P., Marsanda, D. A. E., & Sari, A. P. (2024). Pemanfaatan Natural Language Processing Untuk Pengecekan Ejaan Sesuai Kbbi. *Jurnal Jamastika*, 3(2).
- Maesaroh, S., Afiyati, L., Hakim, L., & Sari, Y. S. (2024). *Bahasa Pemrograman Python*. Sada Kurnia Pustaka.
- Mohammad Farid Naufal (2018). "Analisa Teknik Pembelajaran Dan Pengajaran Pemrograman Pada Universitas Dan Industri", *Jurnal Informatika & Multimedia*, Vol. 10, No. 2, Pp. 1 - 8
- Muhammad Romzi Dan Budi Kurniawan (2020), "Pembelajaran Pemrograman Python Dengan Pendekatan Logika Algoritma", *Jtim: Jurnal Teknik Informatika Mahakarya*, Vol. 3, No. 2, Pp. 37- 44.
- Nazar, R. (2024). Implementasi Pemrograman Python Menggunakan Google Colab. *Jurnal Informatika Dan Komputer*, 15(1).

- Nova, S., Khotimah, N., & Wahyuningrum, M. Y. A. (2024). Pemanfaatan Chatbot Menggunakan Natural Language Processing Untuk Pembelajaran Dasar-Dasar Gui Tkinter Pada Bahasa Pemrograman Python. *Jurnal Juit*, 3(1), 58-65.
- Sabban, M. M., & Kasdam, F. (2024). Analisis Kesalahan Penggunaan Ejaan Dan Tanda Baca Dalam Jurnal Paradigma Vol. 10, No 1, Tahun 2024 Universitas Banda Naira. *Paradigma: Jurnal Ilmu Pendidikan Dan Humaniora*, 10(2).
- Saptadi, N. T. S. (2024). Bahasa Pemrograman Python: Paket Standar Python.
- Saptadi, N. T. S. (2024). Dasar Pemrograman Dengan Bahasa Pemrograman Python: Dasar Data Science (Import Data).
- Sembiring, F. (2020). Bahasa Ular Untuk Pemrograman Python.
- Sutriawan, S. (2024). Langkah Awal Menguasai Bahasa Pemrograman Python. Brin Press.
- Ua, A., Marpaung, E., Ong, J., Savinka, M., Nurhaliza, P., & Ningsih, R., (2023). Penggunaan Bahasa Pemrograman Python Dalam Analisis Faktor Penyebab Kanker Paru-Paru. *Jurnal Publikasi Teknik Informatika*, 2(2), 88-99.