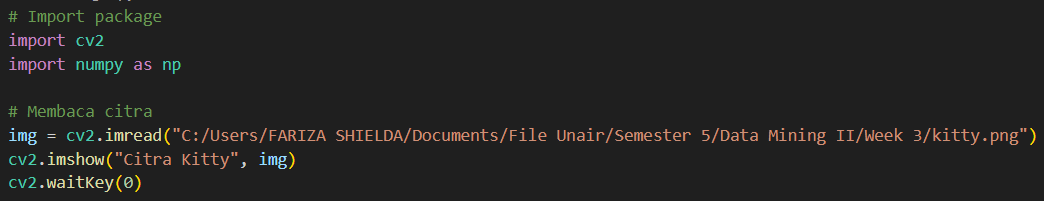
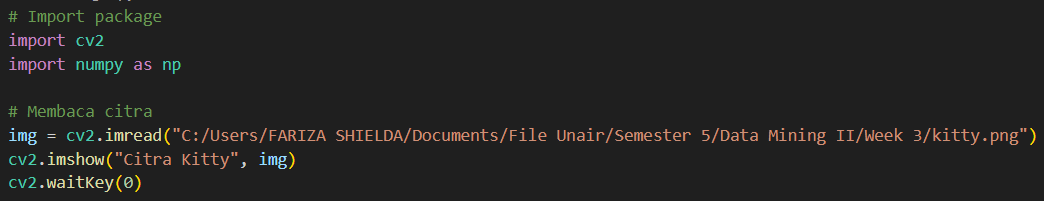
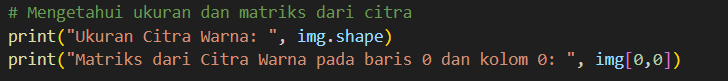
1. **Akuisisi dan Representasi Data Image**
2. Sebelum melakukan akuisisi dan representasi data image, dapat menginstall lalu mengimport library OpenCV CV2 dan Numpy terlebih dahulu.



Untuk membuka citra bernama ‘kitty.png’, maka dapat menggunakan library OpenCV CV2, sebagai berikut:

Berikut adalah hasilnya:

1. Untuk mengetahui ukuran dan matriks dari citra tersebut gunakan kode dibawah ini.



Hasilnya adalah sebagai berikut:



Citra tersebut berukuran (373, 293, 3) yang artinya citra terbagi menjadi 373 baris dan 293 kolom, dimana pada tiap baris dan kolom (pixel) terdapat 3 kanal (channel), dengan urutan Blue, Green, Red, dimana masing-masing kanal memiliki nilai tersendiri. Sedangkan untuk Hasil Nilai Matriks dari Citra Warna adalah [47 43 32], dimana pada baris ke-0 dan kolom ke-0, nilai kanal Blue adalah 47, kanal Green adalah 43, dan Kanal Red adalah 32.

1. Citra berwarna terdiri dari 3 kanal Red, Green, Blue. Lalu memisahkan suatu citra berwarna menjadi komponen Red, Green, dan Blue. Method dalam OpenCV yang dapat digunakan untuk melakukan pemisahan channel adalah split(). Untuk memisahkan ketiga channel, tuliskan kode berikut:



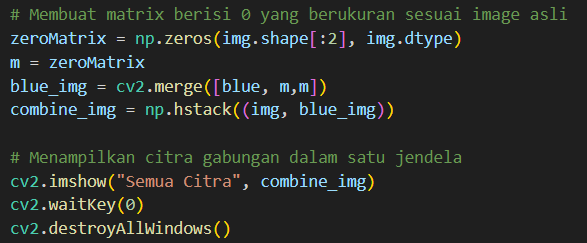
Dengan menggunakan kode diatas, maka telah berhasil memisahkan ketiga channel. Kemudian menampilkan channel Blue dengan menuliskan kode berikut:



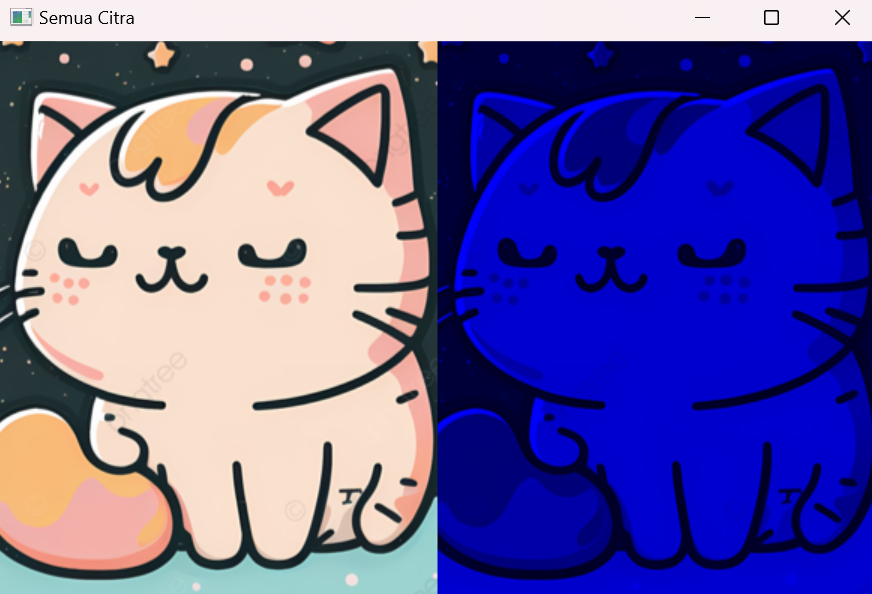
Setelah dipisahkan, komponen Blue akan terlihat seperti gambar dibawah. Karena setiap channel direpresentasikan dalam 1 channel saja, maka tidak terlihat warna Blue dan seperti citra grayscale karena hanya memiliki 1 channel.



Agar channel Blue bisa menampilkan warna sesuai channel-nya maka harus direpresentasikan dalam 3 channel menggunakan method merge(). Kita harus membuat matriks berisi nilai ‘0’ yang memiliki ukuran sama dengan ukuran gambar kita (gambar ‘kitty.png’ memiliki ukuran 373 x 293), kemudian matriks tersebut digunakan untuk mengisi channel Red dan Green.

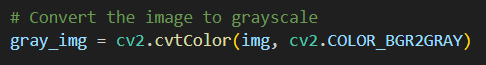


Output dari kode diatas adalah sebagai berikut:

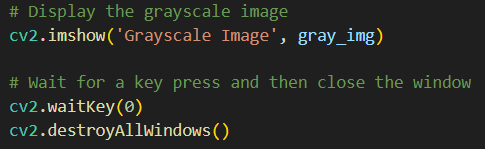


* **Modifikasi kode bagian (a) agar dapat menampilkan citra dalam grayscale.**

Untuk menampilkan citra gambar ‘kitty.png’ dalam grayscale, maka dapat menggunakan kode di bawah ini.



Kemudian, untuk membuka citra dapat menggunakan kode berikut.

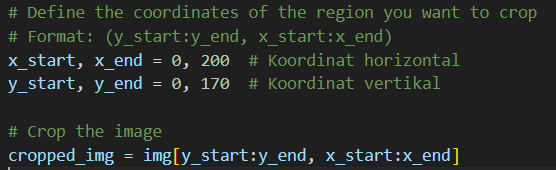


Berikut hasilnya:

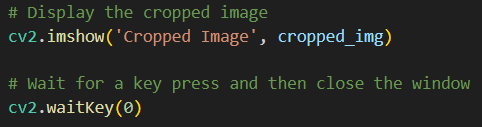


* **Modifikasi kode bagian (a) agar bisa melakukan crop pada citra (terserah dibagian mana saja).**

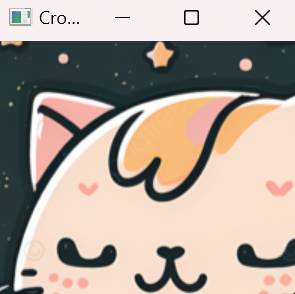
Untuk melakukan crop pada citra dapat menggunakan kode berikut.



Kemudian, untuk menampilkan hasil crop, dapat menggunakan kode di bawah ini.

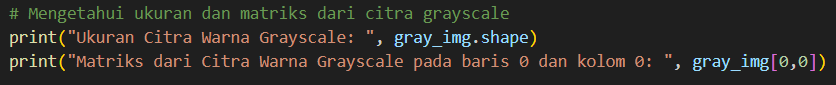


Berikut adalah hasil dari kode di atas:



* **Modifikasi kode bagian (b) agar dapat menampilkan ukuran citra grayscale dari ‘kitty.png’ dan nilai matriks dari citra grayscale pada baris ke-0 dan kolom ke-0. Apakah hasilnya berbeda dengan bagian (b), jelaskan alasannya.**

Untuk mengetahui ukuran dan matriks dari citra grayscale dari ‘kitty.png’ dapat menggunakan kode dibawah ini.



Hasilnya adalah sebagai berikut:



Citra tersebut berukuran (373, 293) yang artinya citra terbagi menjadi 373 baris dan 293 kolom. Sedangkan untuk Hasil Nilai Matriks dari Citra Grayscale pada baris ke-0 dan kolom ke-0 adalah 40.

Hasil Ukuran dan Nilai Matriks pada Citra Warna dan Citra Grayscale berbeda, dimana:

* Ukuran Citra:

1. Citra Warna: Ukuran citra warna adalah (373, 293, 3). Angka-angka ini mengindikasikan bahwa citra ini memiliki 3 dimensi, yaitu lebar (373 piksel), tinggi (293 piksel), dan 3 saluran warna (RGB).
2. Citra Warna Grayscale: Ukuran citra grayscale adalah (373, 293). Ini menunjukkan bahwa citra grayscale hanya memiliki dua dimensi, yaitu lebar (373 piksel) dan tinggi (293 piksel), karena citra grayscale hanya memiliki satu saluran warna.

* Matriks Nilai Piksel:

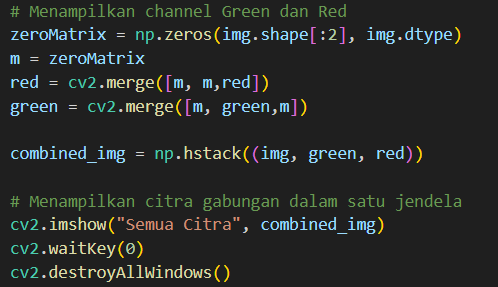
1. Citra Warna: Matriks dari citra warna pada baris 0 dan kolom 0 adalah [47 43 32]. Ini berarti pada koordinat (0, 0) dari citra warna, terdapat tiga nilai warna, masing-masing untuk saluran Red (47), Green (43), dan Blue (32). Oleh karena itu, pada titik ini memiliki informasi tentang warna.
2. Citra Warna Grayscale: Matriks dari citra grayscale pada baris 0 dan kolom 0 adalah 40. Ini berarti pada koordinat (0, 0) dari citra grayscale, terdapat satu nilai tunggal, yaitu tingkat kecerahan atau intensitas cahaya, yang dalam hal ini adalah 40. Citra grayscale hanya menyimpan informasi tentang tingkat kecerahan dan tidak memiliki informasi warna.

* **Modifikasi kode bagian (c) untuk menampilkan channel Green dan Red.**

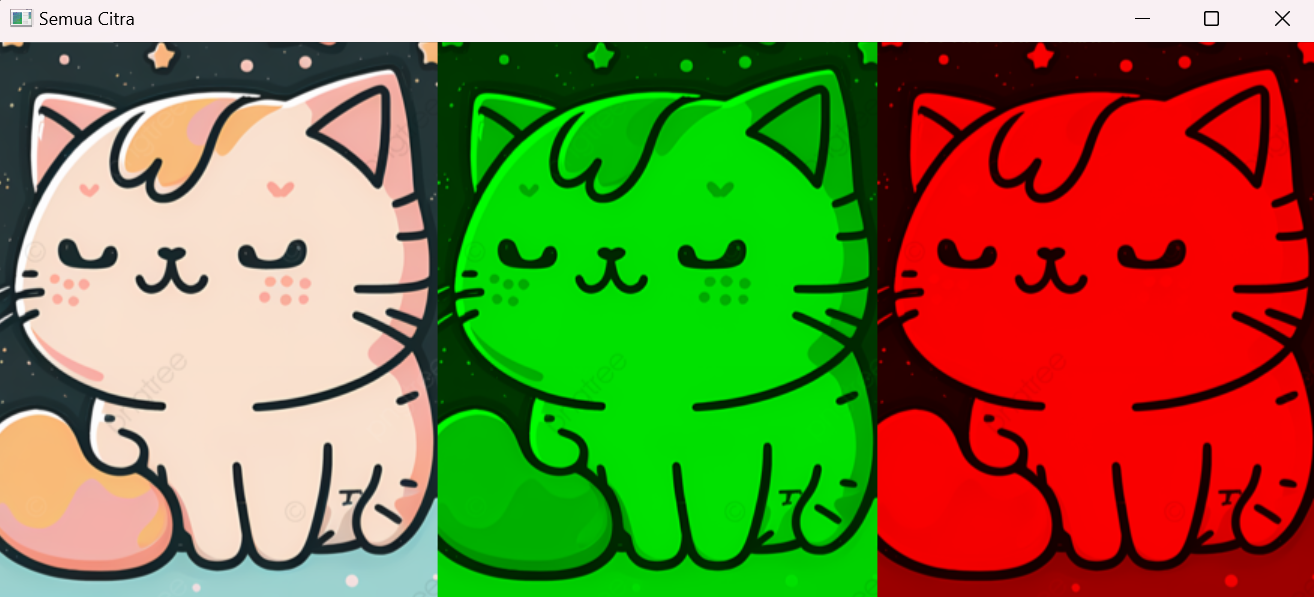
Sebelum menampilkan channel Green dan Red, dilakukan pemisahan ketiga channel seperti yang sudah dilakukan sebelumnya dengan menuliskan kode berikut:



Dengan menggunakan kode diatas, maka telah berhasil memisahkan ketiga channel. Kemudian menampilkan channel Green dan Red dengan menuliskan kode berikut:

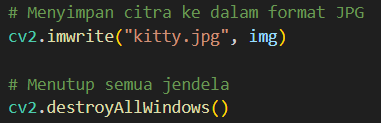
****

Berikut output dari kode di atas.

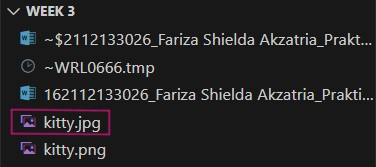


* **Simpanlah file menjadi format selain format awal (PNG jadi JPG, JPG jadi PNG) dengan menggunakan method imwrite pada OpenCV, apakah terdapat perbedaan nilai array pada file citra asli dan file dengan format baru tsb? Jelaskan alasannya.**

Untuk menyimpan file dengan format lain, dapat menggunakan method imwrite() pada OpenCV seperti di bawah ini.

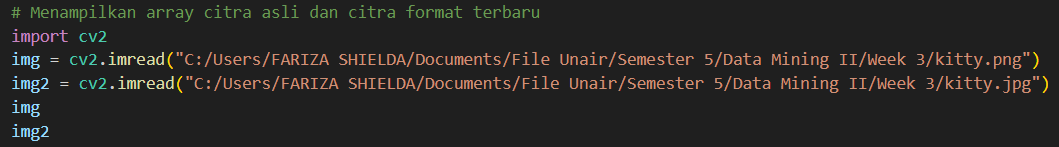


Output dari kode di atas sebagai berikut.



File ‘kitty.png’ telah tersimpan difolder yang sama dengan format yang telah berubah menjadi ‘kitty.jpg’.

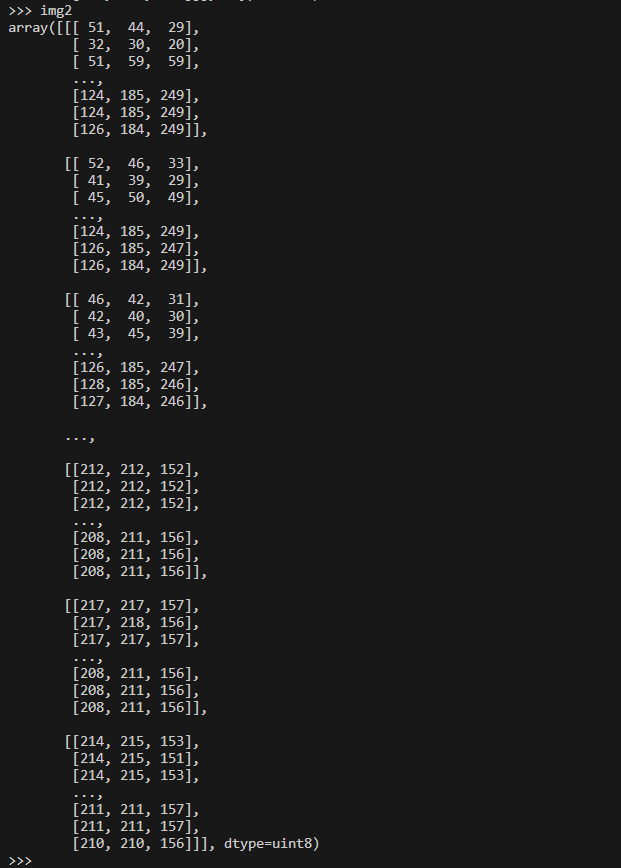
Untuk menampilkan nilai array pada file citra asli dan file dengan format baru, dapat menggunakan kode di bawah ini.



Berikut hasil nilai array pada file citra asli (format png).



Berikut hasil nilai array pada file citra format terbaru (format jpg).



Perbedaan antara kedua array ini dapat dilihat dari nilai-nilai piksel yang berbeda di dalamnya. Di bawah ini adalah perbedaan yang terlihat:

* Warna Piksel:

1. img: Piksel dalam array img memiliki saluran warna RGB yang bervariasi dalam rentang nilai tertentu. Misalnya, dalam citra img, Anda dapat melihat nilai-nilai seperti [47, 43, 32], [35, 32, 22], yang mewakili saluran merah, hijau, dan biru dari setiap piksel.
2. img2: Piksel dalam array img2 juga memiliki saluran warna RGB, tetapi nilainya berbeda dari img. Contohnya, [51, 44, 29], [32, 30, 20].

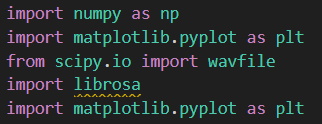
* Kecemerlangan Piksel:

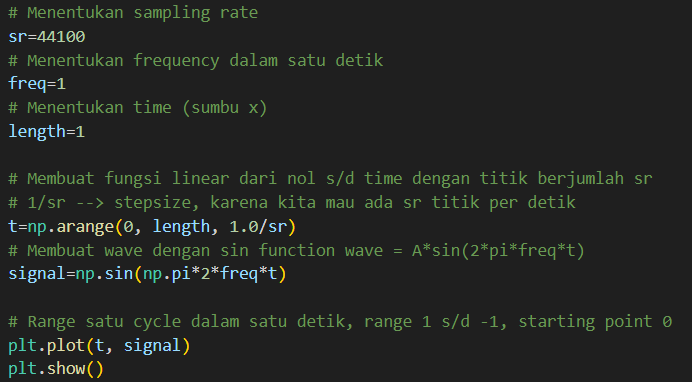
Baik img maupun img2 memiliki citra dengan tingkat kecerahan yang berbeda di beberapa piksel. Misalnya, dalam img, dapat terlihat nilai seperti 127, 185, 249, yang mewakili kecerahan dalam saluran RGB, sedangkan dalam img2, memiliki nilai seperti 124, 185, 249, yang juga mewakili tingkat kecerahan dalam saluran RGB.

* Bentuk Array:

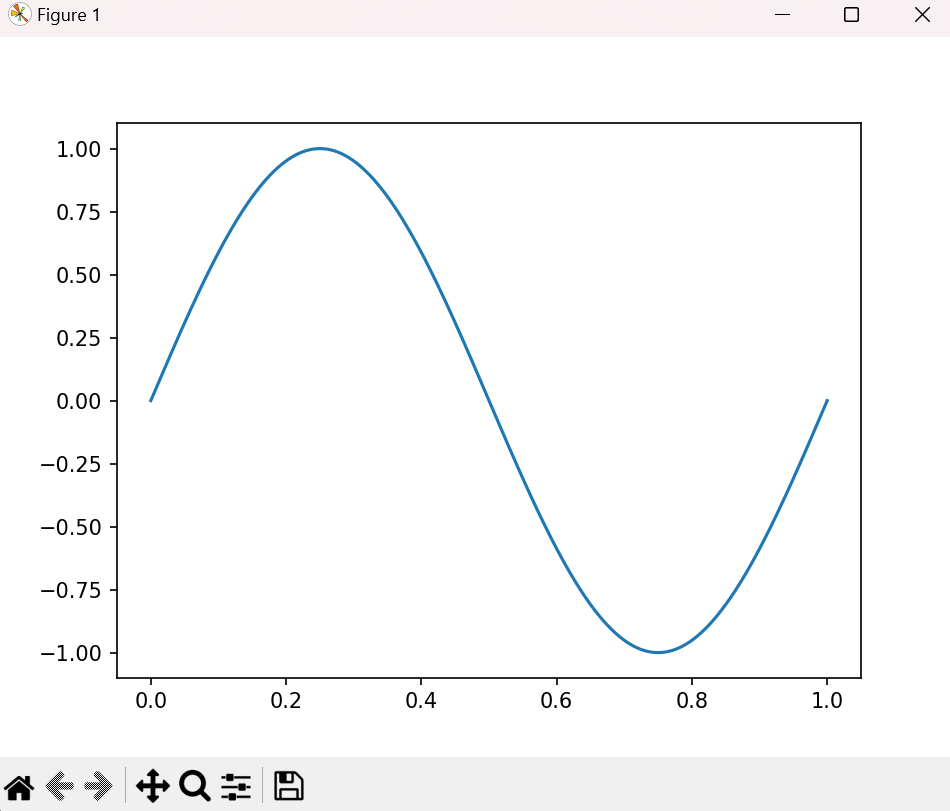
Kedua array ini memiliki bentuk (shape) yang sama, yaitu (373, 293, 3), yang mengindikasikan bahwa keduanya adalah gambar berwarna dengan lebar 373 piksel, tinggi 293 piksel, dan tiga saluran warna RGB. Perbedaan utama di sini adalah warna piksel dan nilai tingkat kecerahan yang berbeda antara kedua gambar. Selain itu, bentuk array mereka tetap sama, menunjukkan bahwa keduanya masih merupakan citra berwarna dengan tiga saluran warna (RGB).

1. **Akuisisi dan Representasi Data Audio**
2. Sebelum membuat sinyal audio berupa gelombang, install dan import beberapa library yang dibutuhkan. Kemudian harus menentukan dahulu sampling rate, frekuensi, dan panjang menggunakan kode di bawah ini.





Hasilnya kode diatas adalah gambar gelombang dibawah ini. Gelombang yang dihasilkan memiliki amplitudo antara 1 dan -1, terdapat 1 gelombang dalam 1 detik (wavelength), dan frekuensinya adalah 1.



Lalu simpan sinyal gelombang ke dalam format wav dengan nama ‘file.wav’ menggunakan kode dibawah ini.

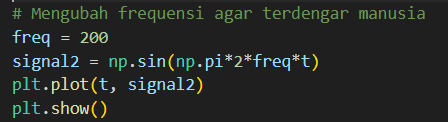


Dalam folder kita akan tersimpan file.wav seperti ini:

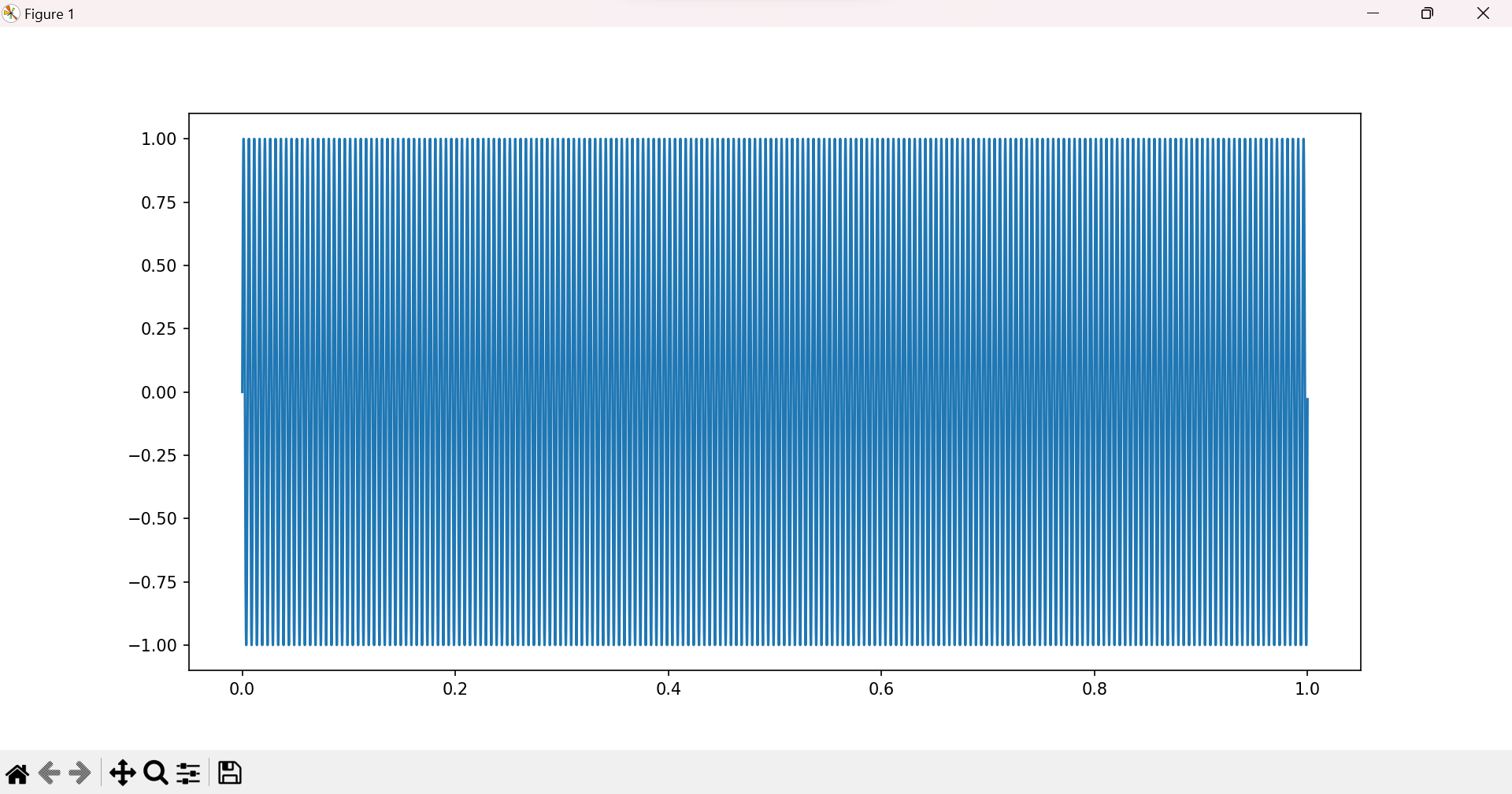


Ketika mendengar file tersebut tidak ada suara yang muncul karena dalam program diatur frekuensinya adalah 1, sementara telinga manusia hanya bisa mendengar bunyi yang memiliki frekuensi 20 hingga 20.000 Hz.

1. Supaya terdengar manusia, harus mengganti frekuensinya. Misalkan, mengubah frekuensinya menjadi 200.



Hasilnya kode diatas adalah gambar gelombang dibawah ini. Gelombang yang dihasilkan memiliki amplitudo antara 1 dan -1, terdapat 200 gelombang dalam 1 detik (wavelength), dan frekuensinya adalah 200.



Lalu simpan sinyal gelombang ke dalam format wav dengan nama ‘file1.wav’ menggunakan kode dibawah ini.



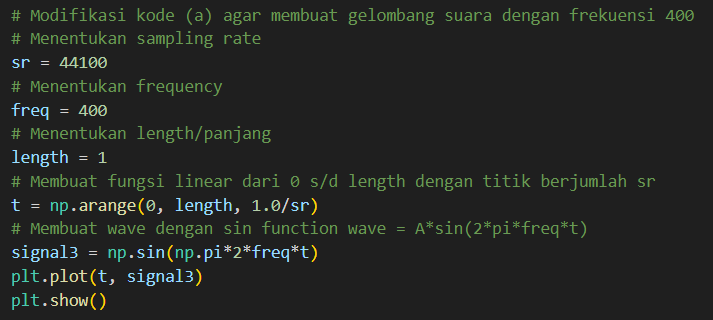
Dalam folder kita akan tersimpan file1.wav seperti ini:



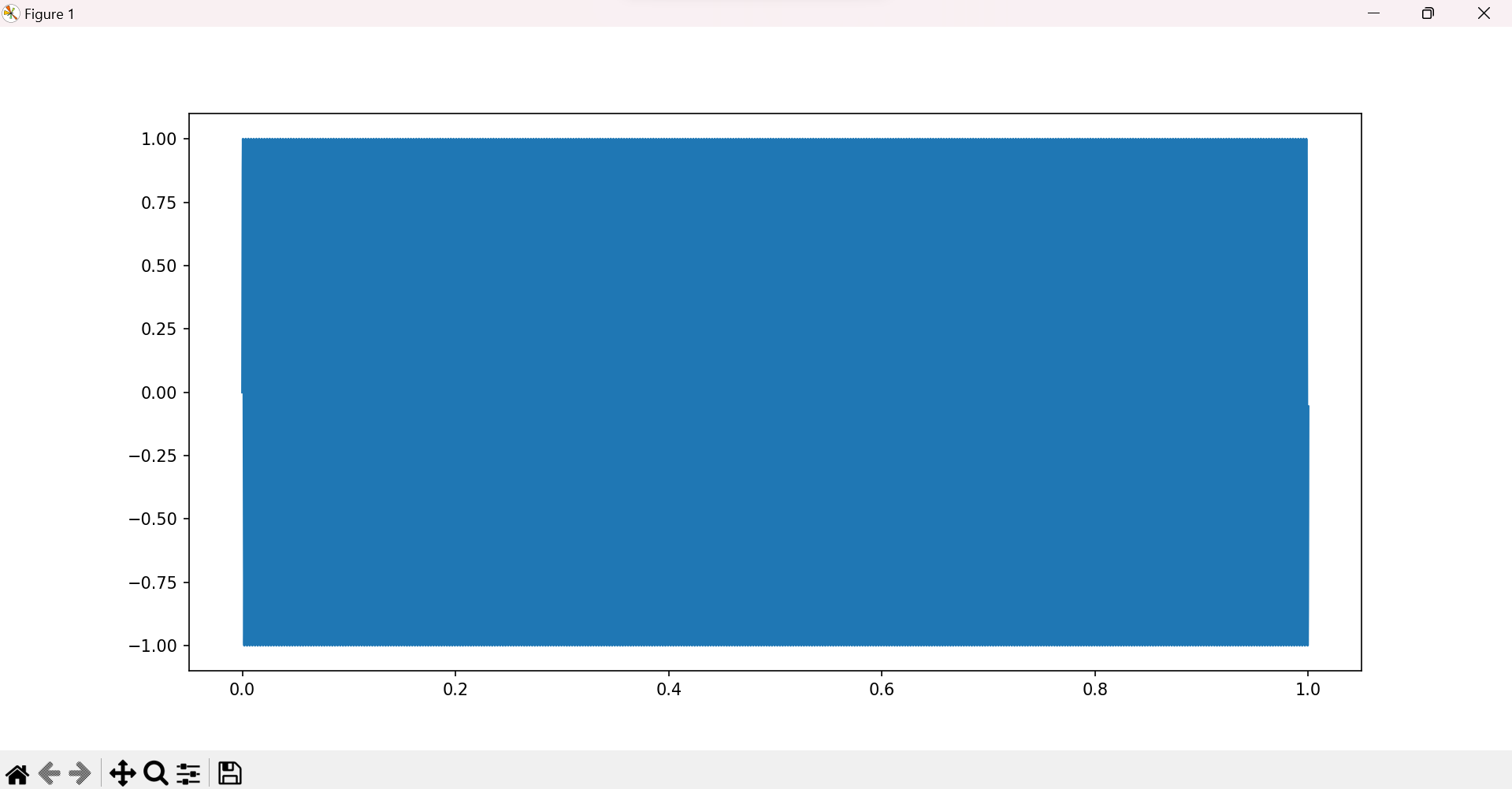
Ketika mendengar file tersebut sudah ada suara yang muncul karena dalam program diatur frekuensinya adalah 200, karena telinga manusia bisa mendengar bunyi yang memiliki frekuensi 20 hingga 20.000 Hz.

* **Modifikasi kode bagian (a) agar membuat gelombang suara dengan frekuensi 400. Berapa panjang gelombangnya sekarang? Apakah file suara yang dihasilkan frekuensi 400 berbeda dengan file suara yang dihasilkan kode bagian (b)? jelaskan alasannya.**

Menentukan sampling rate, frekuensi, dan panjang menggunakan kode di bawah ini.



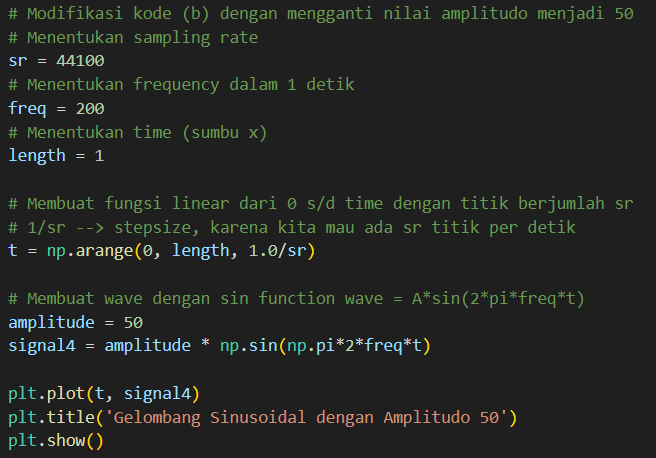
Hasilnya kode diatas adalah gambar gelombang dibawah ini. Gelombang yang dihasilkan memiliki amplitudo antara 1 dan -1, terdapat 1 gelombang dalam 1 detik (wavelength), dan frekuensinya adalah 400.



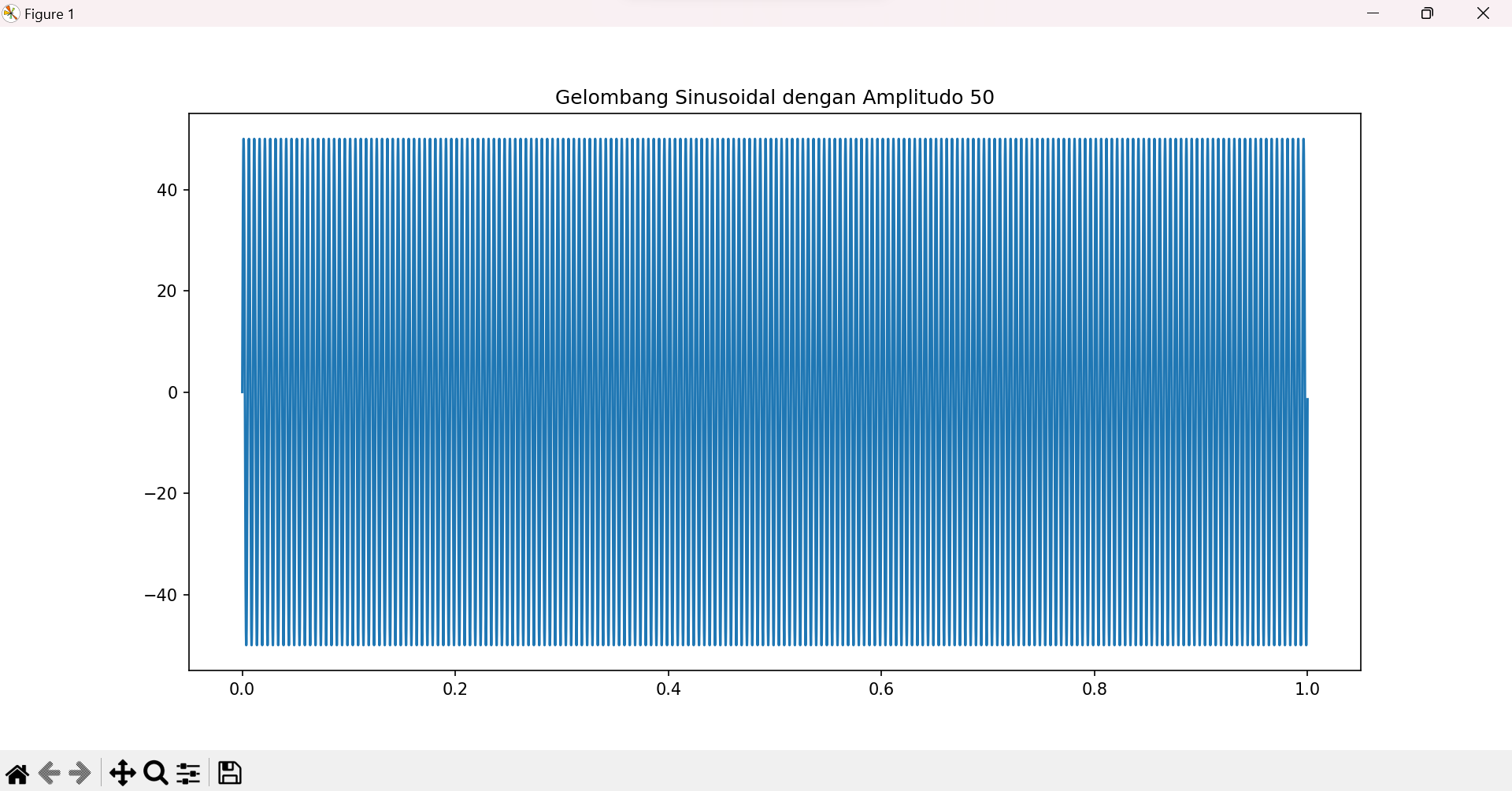
File suara yang dihasilkan dengan frekuensi 400 berbeda dengan hasil kode bagian (b) yang memiliki frekuensi 200. Perbedaan dalam frekuensi menghasilkan perbedaan dalam tinggi rendahnya nada. Frekuensi 400 akan menghasilkan nada yang lebih tinggi dibandingkan dengan frekuensi 200. Dengan kata lain, meskipun panjang gelombang (waktu) tetap sama, perbedaan dalam frekuensi menghasilkan perbedaan dalam tinggi nada. Frekuensi adalah ukuran berapa kali gelombang lengkap berulang dalam satu detik (Hz). Ketika frekuensi meningkat (misalnya dari 200 Hz menjadi 400 Hz), gelombang lengkap berulang dua kali lebih cepat dalam satu detik, sehingga nada yang dihasilkan terdengar lebih tinggi.

* **Modifikasi kode bagian (b) dengan mengganti nilai amplitudo menjadi 50. Pada output gelombang, berapakah panjang dan tinggi gelombang sekarang? Apakah terdapat perbedaan bunyi dengan hasil suara dari kode bagian (b)? jelaskan alasannya.**

Membuat gelombang suara sinusoidal dengan frekuensi yang sama (400 Hz) tetapi dengan nilai amplitudo yang berbeda (50) dapat menggunakan kode berikut.



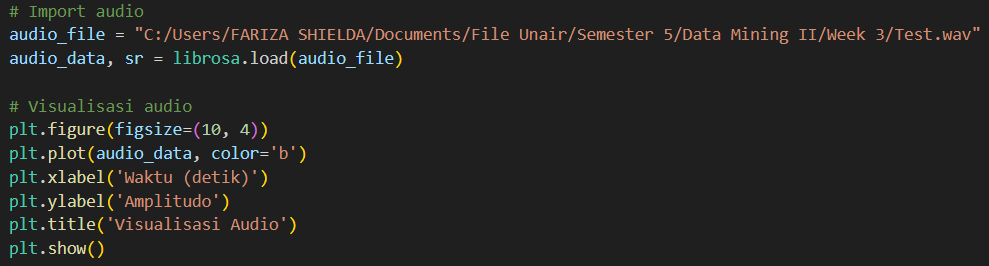
Berikut output dari kode di atas:



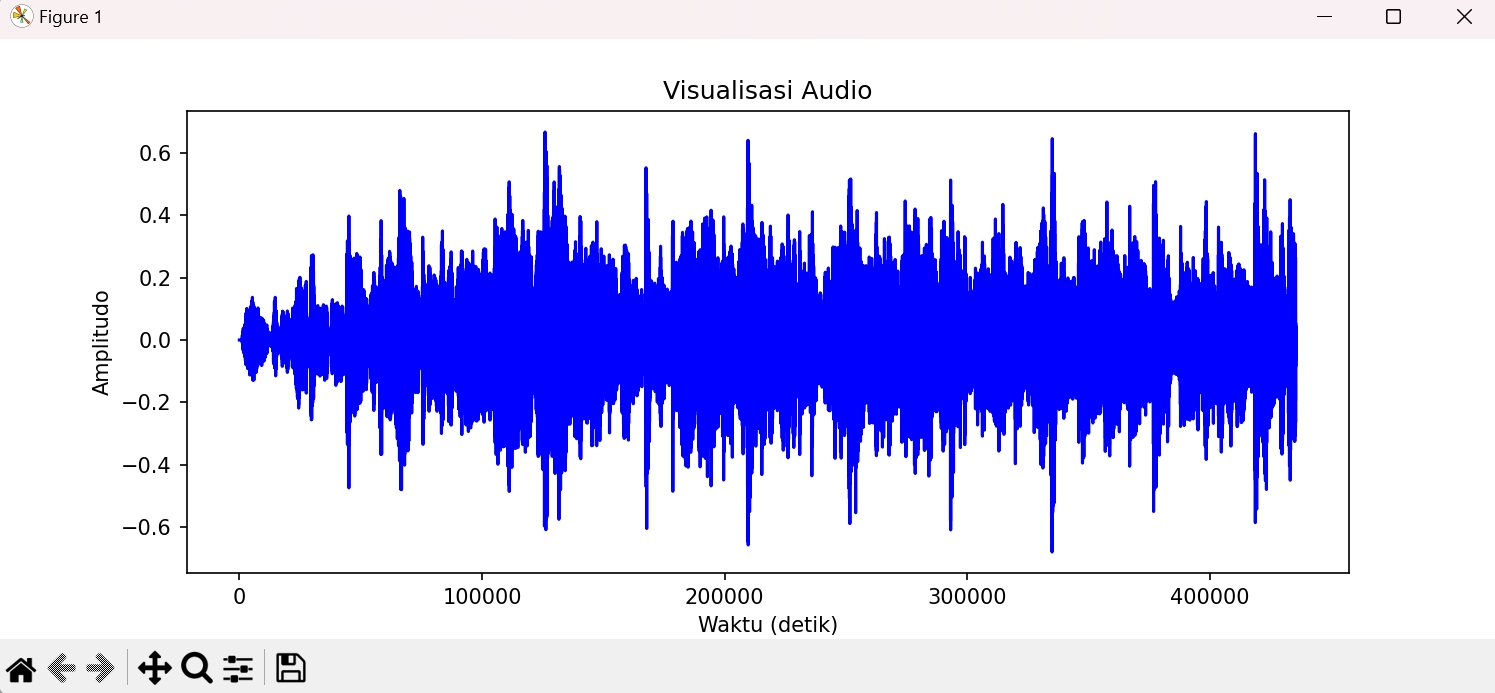
Panjang gelombang tetap tidak berubah karena tidak mengubah nilai variabel length masih tetap 1 detik. Tinggi gelombang (amplitudo) sekarang adalah 50. Ini berarti tinggi gelombang yang dihasilkan sekarang lebih besar daripada yang dihasilkan dalam kode sebelumnya (dengan amplitudo 1). Perubahan amplitudo akan memengaruhi tinggi rendahnya volume suara. Dalam kode ini, tinggi gelombang yang lebih besar (50) akan menghasilkan suara yang lebih keras atau lebih nyaring dibandingkan dengan hasil suara dari kode sebelumnya (amplitudo 1). Ini berarti ada perbedaan bunyi yang signifikan antara hasil suara dari kode ini dengan hasil suara dari kode (b) karena perubahan dalam amplitudo. Tinggi gelombang yang lebih besar menghasilkan suara yang lebih keras atau lebih nyaring.

* **Rekam suara baru dari device anda, lakukan visualisasi suara tersebut**

Setelah melakukan perekaman suara, maka untuk memunculkan visualisasinya dapat menggunakan kode berikut.

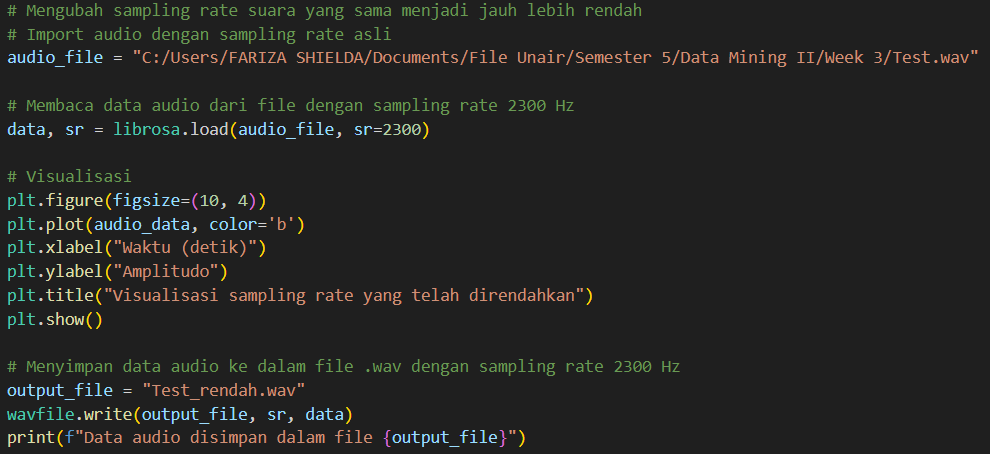


Output dari kode di atas adalah sebagai berikut:

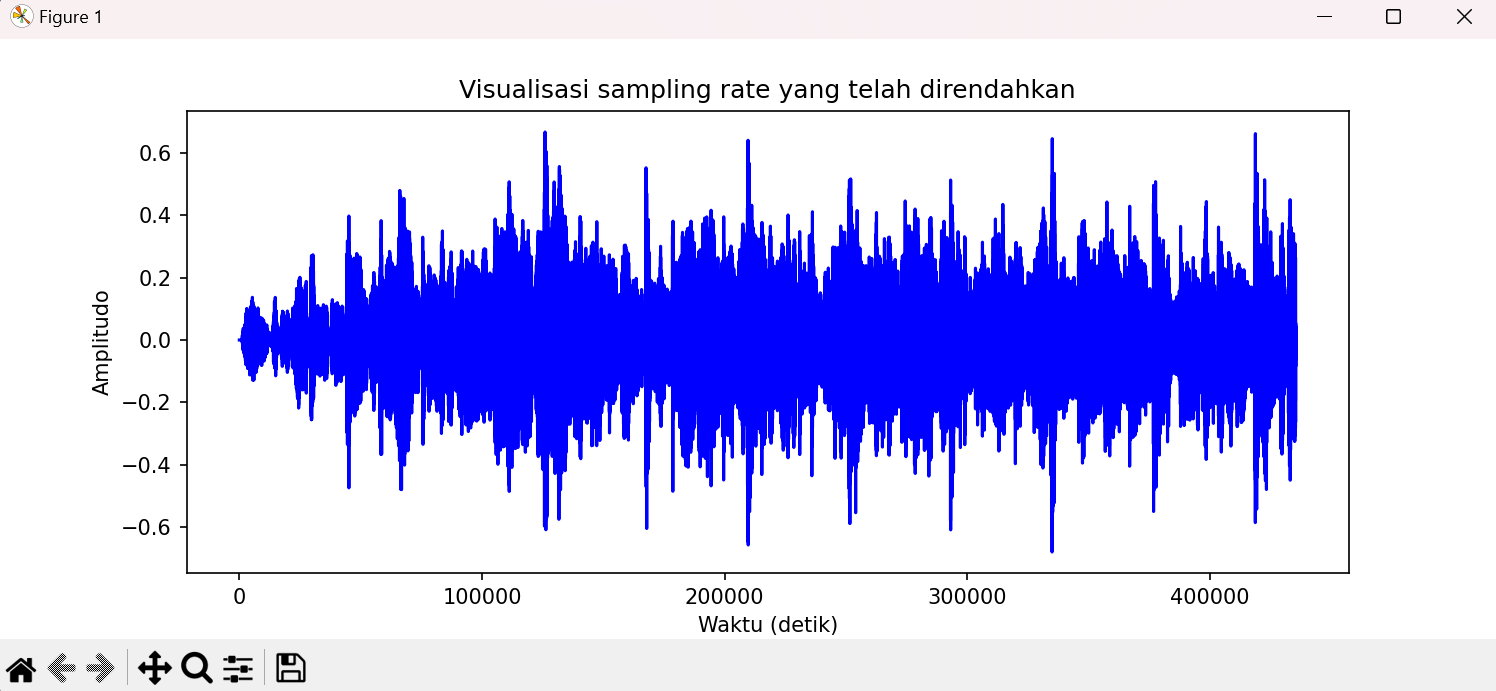


* **Coba untuk ubah sampling rate suara yang sama menjadi jauh lebih rendah, dengarkan, lalu apa efeknya? Jelaskan alasannya**

Mengubah sampling rate suara menjadi jauh lebih rendah dapat menggunakan kode berikut.



Berikut output dari kode di atas.



Setelah mendengarkan file audio terbaru dengan sampling ratenya yang telah direndahkan, suara menjadi terdengar lebih kasar, dan kurang jelas. Terlebih lagi, hal ini dapat mengakibatkan distorsi audio, di mana komponen frekuensi tinggi dalam audio mungkin terdengar sebagai frekuensi yang lebih rendah, menciptakan efek suara yang tidak diinginkan. Hasilnya, suara tersebut dapat terdengar seperti suara yang dihasilkan oleh komputer atau robot, yang sangat berbeda dari karakter suara aslinya. Mengubah sampling rate suara menjadi jauh lebih rendah akan mengakibatkan pengurangan kualitas audio secara signifikan. Hal ini disebabkan oleh fakta bahwa dengan mengurangi sampling rate, dapat menghilangkan banyak informasi yang relevan dalam audio.

1. **Akuisisi dan Representasi Data Text**
2. ASCII

Fungsi ‘ord’ pada python digunakan untuk mengubah karakter menjadi ASCII code, sedangkan fungsi ‘char’ digunakan untuk mengubah ASCII code menjadi karakter.



Berikut output dari kode di atas.



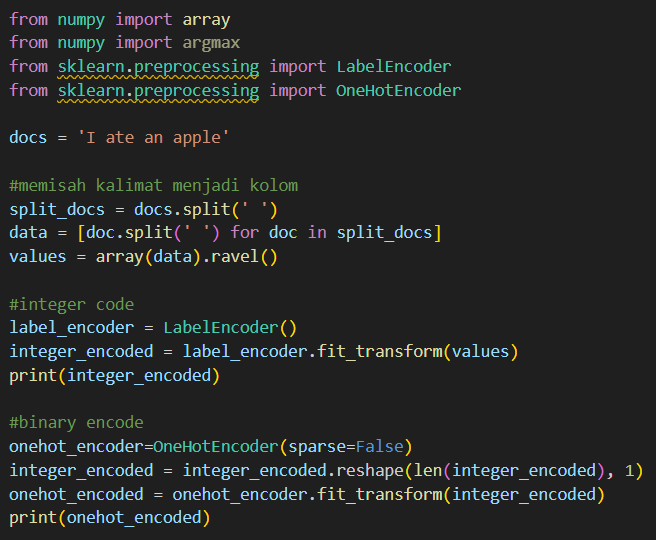


Berikut output dari kode di atas.



1. One-Hot Encoding

Kode berikut digunakan untuk melakukan one-hot encoding pada suatu kalimat.



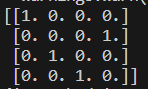
Berikut index dari tiap kata dalam kalimat:



Yang memiliki arti:

|  |  |  |  |
| --- | --- | --- | --- |
| I | ate | an | apple |
| 0 | 3 | 1 | 2 |

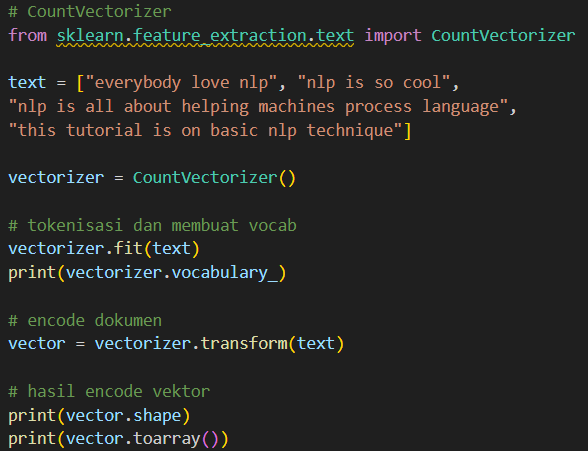
Berikut adalah hasil one-hot encoding untuk kalimat “I ate an apple”.



Sehingga [[1 0 0 0] [0 0 0 1] [0 1 0 0] [0 0 1 0]] merupakan one-hot encoding dari masing-masing kata “I ate an apple”.

1. CountVectorizer

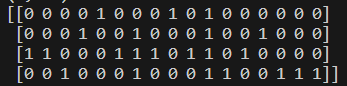
Kode berikut digunakan untuk menghitung CountVectorizer pada suatu korpus.



Daftar vocabulary atau kata dari dokumen adalah sebagai berikut:

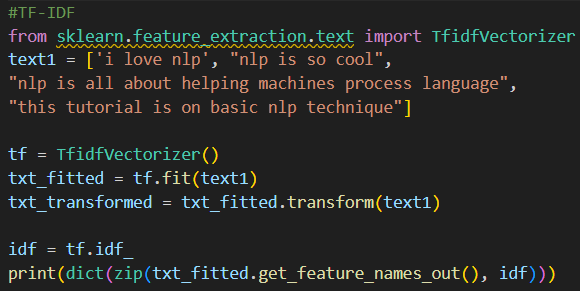
Ukuran vektor adalah **(4,16)**, karena dalam korpus terdapat **4** dokumen, dimana terdapat **16** kata unik dalam korpus tersebut.

Sehingga, CountVectorizer output pada korpus tersebut adalah sebagai berikut:

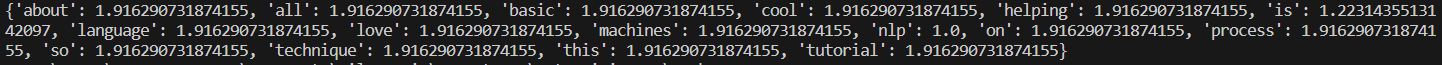


1. TF-IDF

Kode berikut digunakan untuk menghitung TF-IDF pada suatu korpus.

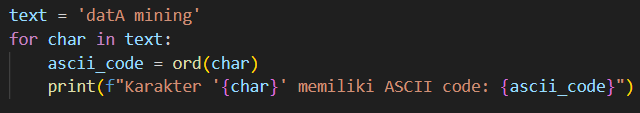


Berikut adalah hasilnya:

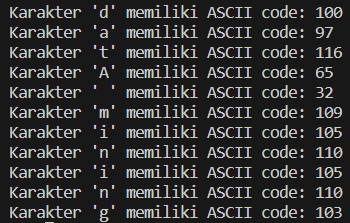
Kata ‘nlp’ memiliki nilai paling kecil karena kata tersebut merupakan kata yang paling sering muncul dalam korpus, terdapat kata ‘nlp’ pada tiap kalimat. Diikuti dengan stopword ‘is’ dengan nilai 1.2231435513142097, karena kata ‘is’ muncul di tiga kalimat dalam korpus.

* **Modifikasi kode bagian (a) agar bisa menampilkan ASCII code untuk kata ‘datA mining’**

Untuk memodifikasi bagian (a) agar dapat menampilkan ASCII code pada kata ‘datA mining’ dapat menggunakan kode berikut:

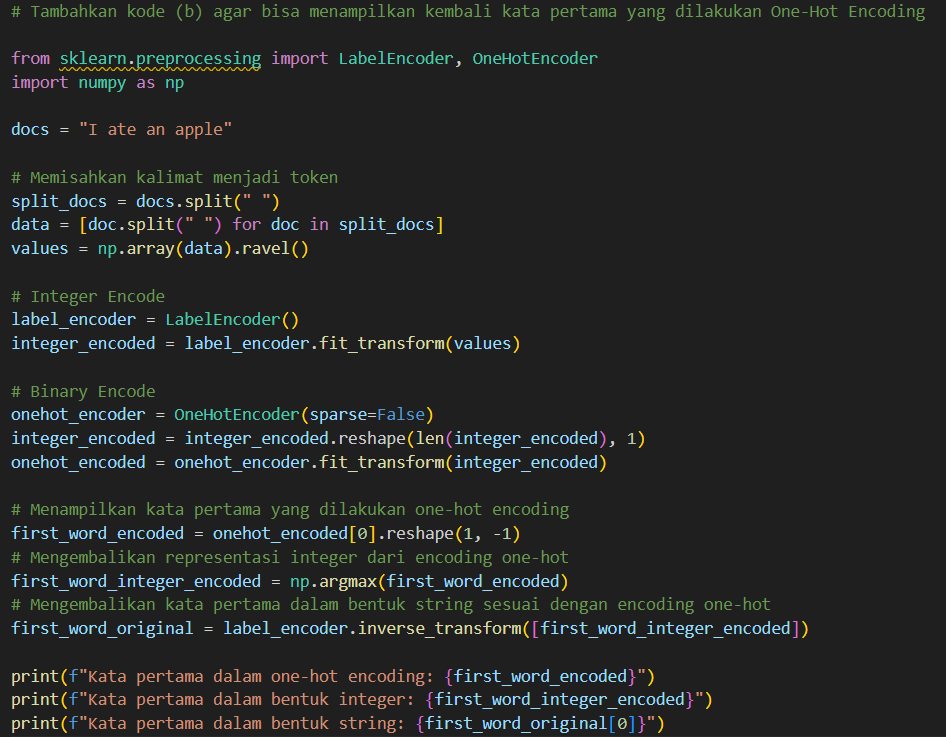


Output dari kode di atas adalah:



* **Tambahkan kode bagian (b) agar bisa menampilkan kembali kata pertama yang di lakukan one-hot encoding**

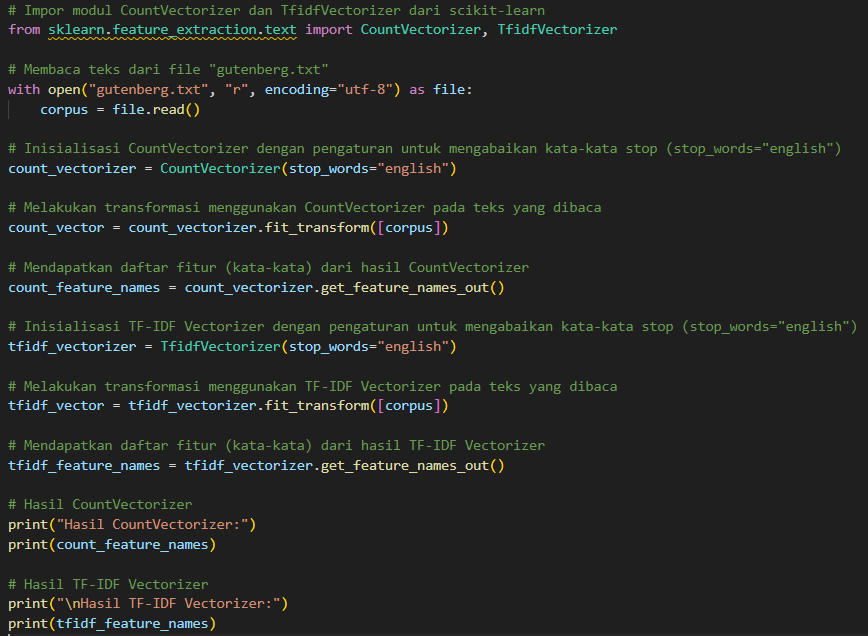
Untuk menambahkan kode bagian (b) agar bisa menampilkan kembali kata pertama yang di lakukan one-hot encoding dapat menggunakan kode di bawah ini.



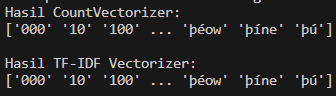
Berikut output dari kode di atas.



* **Download file https://www.gutenberg.org/files/16328/16328-0.txt, kemudian lakukan CountVectorizer dan TF-IDF pada korpus tersebut. Jelaskan hasil yang didapatkan.**



Berikut output yang dihasilkan.



Output yang dapat terlihat adalah daftar kata atau fitur yang telah diekstrak dari korpus teks setelah menerapkan CountVectorizer dan TF-IDF Vectorizer. Di sini, setiap kata atau fitur dalam teks diperlakukan sebagai kolom dalam matriks yang dihasilkan oleh CountVectorizer dan TF-IDF Vectorizer.

* Hasil CountVectorizer: Daftar ini berisi kata-kata unik yang ditemukan dalam korpus setelah menerapkan CountVectorizer. CountVectorizer menghitung berapa kali setiap kata muncul dalam dokumen dan kemudian membuat matriks di mana setiap baris mewakili dokumen, dan setiap kolom mewakili kata. Nilai dalam sel matriks adalah jumlah kemunculan kata dalam dokumen. Jadi, daftar ini berisi semua kata unik yang ditemukan dalam dokumen, termasuk kata-kata umum seperti "the," "and," dan lainnya.
* Hasil TF-IDF Vectorizer: Daftar ini adalah daftar kata unik yang ditemukan dalam korpus setelah menerapkan TF-IDF Vectorizer. TF-IDF Vectorizer mengukur pentingnya setiap kata dalam dokumen dengan mempertimbangkan seberapa sering kata tersebut muncul dalam dokumen dan seberapa umum kata tersebut dalam seluruh korpus. Daftar ini mencakup kata-kata yang memiliki nilai TF-IDF yang lebih tinggi, yang berarti kata-kata ini dianggap lebih penting atau lebih unik dalam konteks dokumen tersebut.

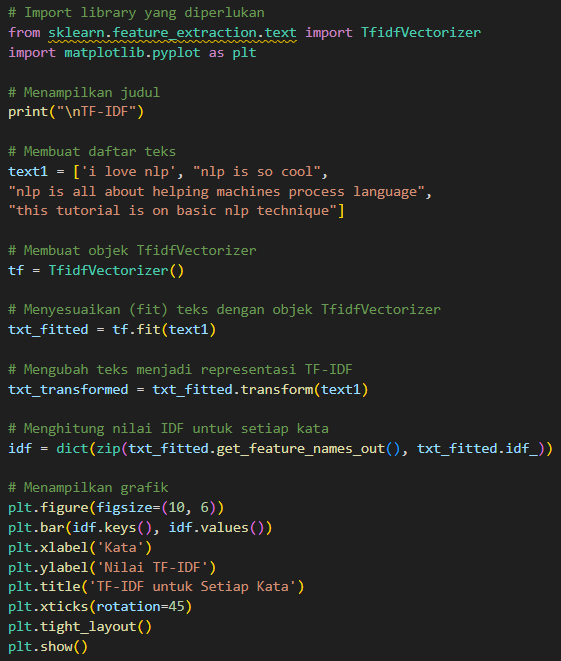
Jadi, perbedaan antara keduanya adalah dalam cara penghitungan pentingnya kata-kata dalam dokumen. Hasil dari CountVectorizer mencakup semua kata dalam dokumen, sementara hasil dari TF-IDF Vectorizer mencakup kata-kata yang memiliki tingkat penting yang lebih tinggi dalam konteks dokumen tersebut. Dalam banyak kasus, hasil dari TF-IDF Vectorizer lebih berguna untuk analisis teks karena mereka memberikan penekanan pada kata-kata yang lebih informatif.

* **Modifikasi kode bagian (d) agar bisa menampilkan grafik dari tiap kata. contohnya seperti gambar di bawah.**

Chart, bar chart

Description automatically generated

Untuk memodifikasi kode bagian (d) agar bisa menampilkan grafik dari tiap kata dapat menggunakan kode di bawah ini:

****

Berikut output yang dihasilkan:

