**LAPORAN TUGAS BESAR  
KECERDASAN BUATAN**



# OLEH

# Lexy Fadillah N. 1841720097

**Lili Nur Indah Sari 1841720037**

**Raka Arya Alzidan W. 1841720077**

**R.B Dwi Rizky Arifandi 1841720181**

**Thariq Alfa Benriska 1841720165**

**PROGRAM STUDI D4 TEKNIK INFORMATIKA**

**JURUSAN TEKNOLOGI INFORMASI**

**POLITEKNIK NEGERI MALANG**

1. **Latar Belakang**

Pernahkah Anda bertanya-tanya bagaimana menambahkan pengenalan ucapan(speech recognition) ke proyek Python Anda? Jika demikian, kemudian terus membaca! Lebih mudah daripada yang Anda bayangkan. Jauh dari menjadi mode, keberhasilan yang luar biasa dari produk berkemampuan *speech* seperti Amazon Alexa telah membuktikan bahwa beberapa tingkat dukungan *speech* akan menjadi aspek penting dari teknologi rumah tangga untuk masa mendatang. Jika Anda berpikir tentang hal ini, alasan mengapa cukup jelas. Menggabungkan pengenalan suara ke dalam aplikasi Python Anda menawarkan tingkat interaktivitas dan aksesibilitas yang beberapa teknologi dapat cocok.

Perbaikan aksesibilitas sendiri patut dipertimbangkan. Pengenalan ucapan memungkinkan orang tua dan gangguan secara fisik dan penglihatan untuk berinteraksi dengan produk dan layanan yang canggih secara cepat dan alami.

1. **Pendekatan Yang Digunakan**

* Pengguna berbicara ke komputer melalui mikrofon, yang pada gilirannya, mengidentifikasi themdaing kata-kata dan mengirimkannya ke perangkat NLP untuk diproses lebih lanjut. Setelah dikenali, kata-kata tersebut dapat digunakan dalam berbagai aplikasi seperti display, robotika, perintah ke komputer, dan pendiktean. Kata recognizer adalah sistem pengenalan ucapan yang mengidentifikasi kata-kata individual. Sistem perintis awal hanya bisa mengenali huruf dan angka individual. Saat ini, mayoritas sistem pengenalan kata sudah mengalami perkembangan adalah pengenal kata dan memiliki lebih dari 95% pengakuan. Sistem seperti itu mampu mengenali kosa kata kecil dari kata-kata tunggal atau ungkapan-ungkapan sederhana. Seseorang harus mengucapkan informasi masukan dengan kata-kata lisan yang jelas, dengan jeda di antara kata-kata, untuk memasukkan data di komputer. Pengenal ucapan terus menerus mengalami kemajuan, jauh lebih sulit dibangun daripada pengenal kata. Anda berbicara kalimat lengkap ke komputer. Masukan akan dikenali dan kemudian diproses oleh NLP.

1. **Komponen Dalam Program Yang Dibuat**
   * Import random: menyediakan fast pseudorandom number generator berdasarkan algoritma



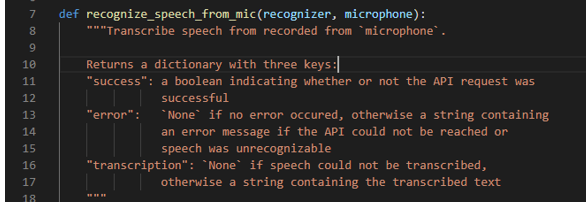
* + Import time: memasukkan yang berisi fungsi dan class untuk operasi tanggal dan waktu.



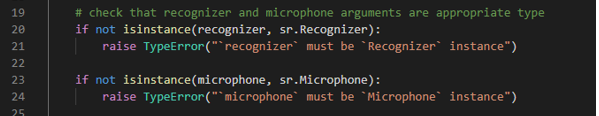
* + Import speech recognition as sr : memasukkan pengenalan ucapan atau Speech Recognition (SR), adalah suatu pengembangan teknik dan sistem yang memungkinkan komputer untuk menerima masukan berupa kata yang diucapkan.



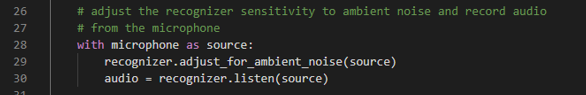
* + Def recognition\_speech\_from\_mic(recognizer, microphone) : transkripsi speech dari recorder dari ‘mikrofon’.



* + Memeriksa recognizer dan argument mikrofon adalah jenis yang sama atau tidak.



* + menyesuaikan sensitivitas pengenal ke kebisingan sekitar dan rekam audio dari mikrofon.

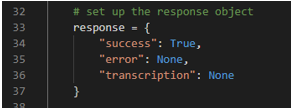


* + mengatur objek respons

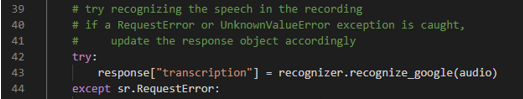
success : true

error : none

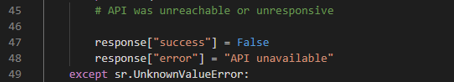
transcription: none



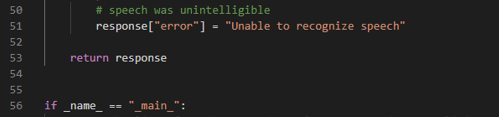
* + mencoba mengenali recognizing dalam rekaman. Jika terdapat RequestError atau UnknownValueError exception, perbarui objek respons sesuai



* + API tidak terjangkau atau tidak responsif



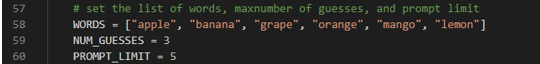
* + ucapan tidak bisa dipahami maka mengeluarkan respon error : unable to recognize speech



* + mengatur daftar kata=apple, banana, grape, orange, mango, lemon.

jumlah tebakan= 3

batas prompt=5



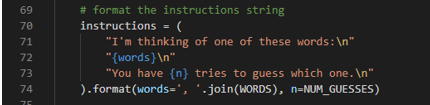
* + buat instance pengenal dan mik



* + mendapatkan kata acak dari daftar(list)



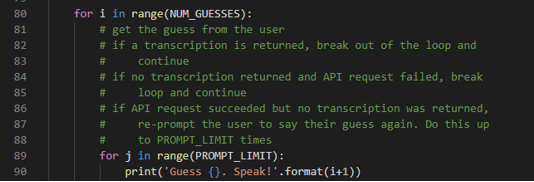
* + format string instruksi

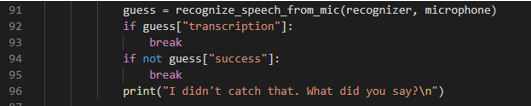


* + menunjukkan instruksi dan tunggu 3 detik sebelum memulai permainan

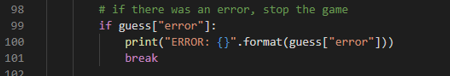


* + dapatkan tebakannya dari pengguna. jika transkripsi dikembalikan, keluar dari loop dan lanjutkan. jika tidak ada transkripsi yang dikembalikan dan permintaan API gagal, putuskan loop dan lanjutkan. jika permintaan API berhasil tetapi tidak ada transkripsi yang dikembalikan, minta kembali pengguna untuk mengatakan tebakan mereka lagi. Lakukan ini hingga PROMPT\_LIMIT kali





* + jika ada kesalahan, hentikan game



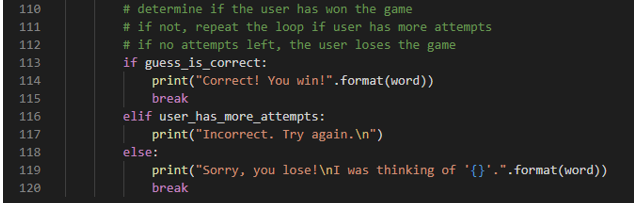
* + meunjukkan pada pengguna transkripsi



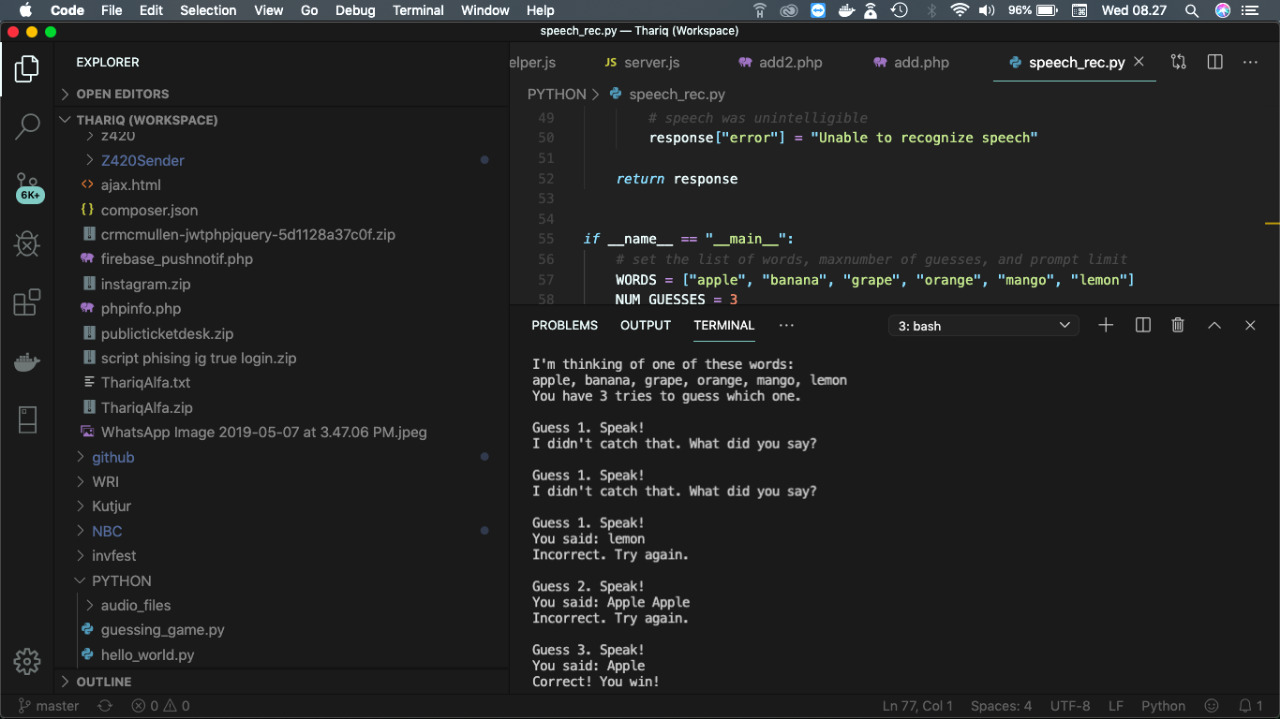
* + menententukan apakah tebakan itu benar dan jika ada upaya yang tetap



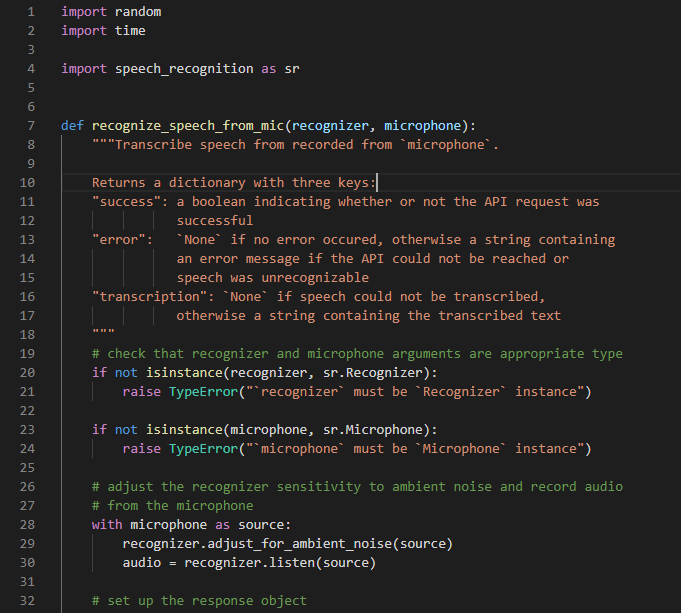
* + tentukan apakah pengguna telah memenangkan permainan. jika tidak, ulangi loop jika pengguna memiliki lebih banyak upaya. jika tidak ada upaya yang tersisa, pengguna kehilangan permainan

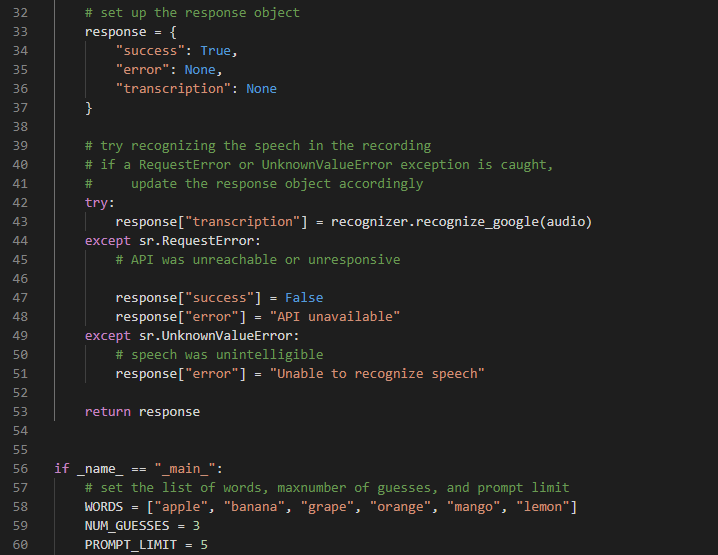


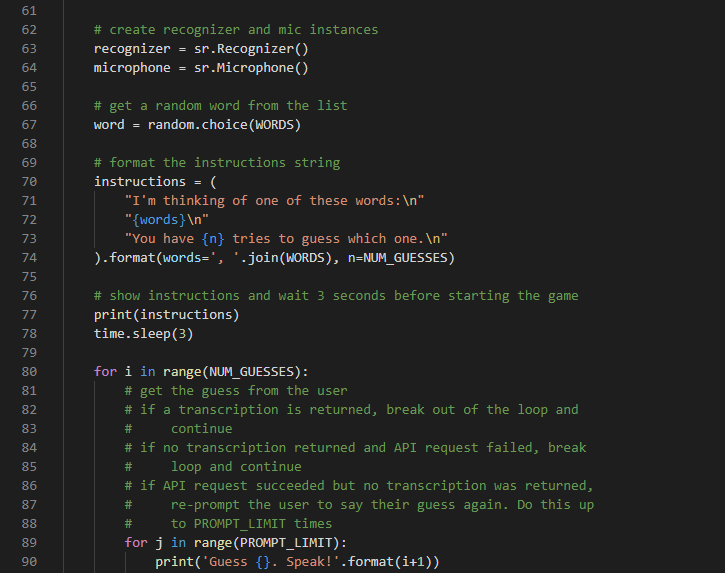
1. **Output**

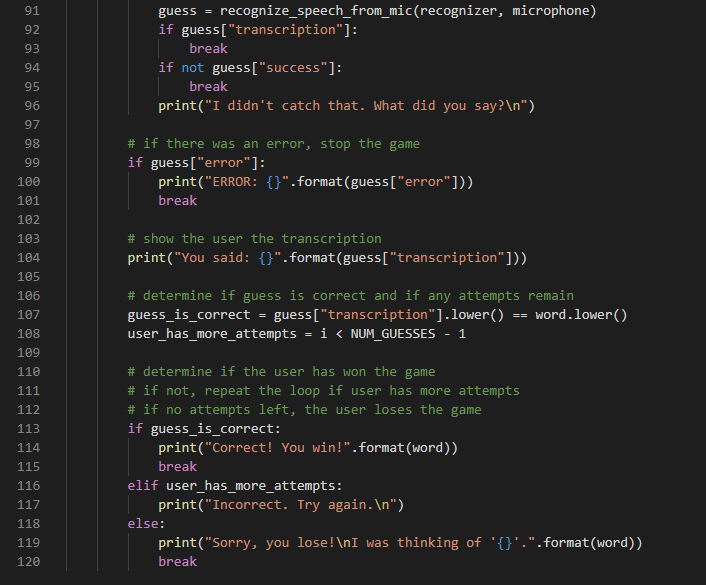


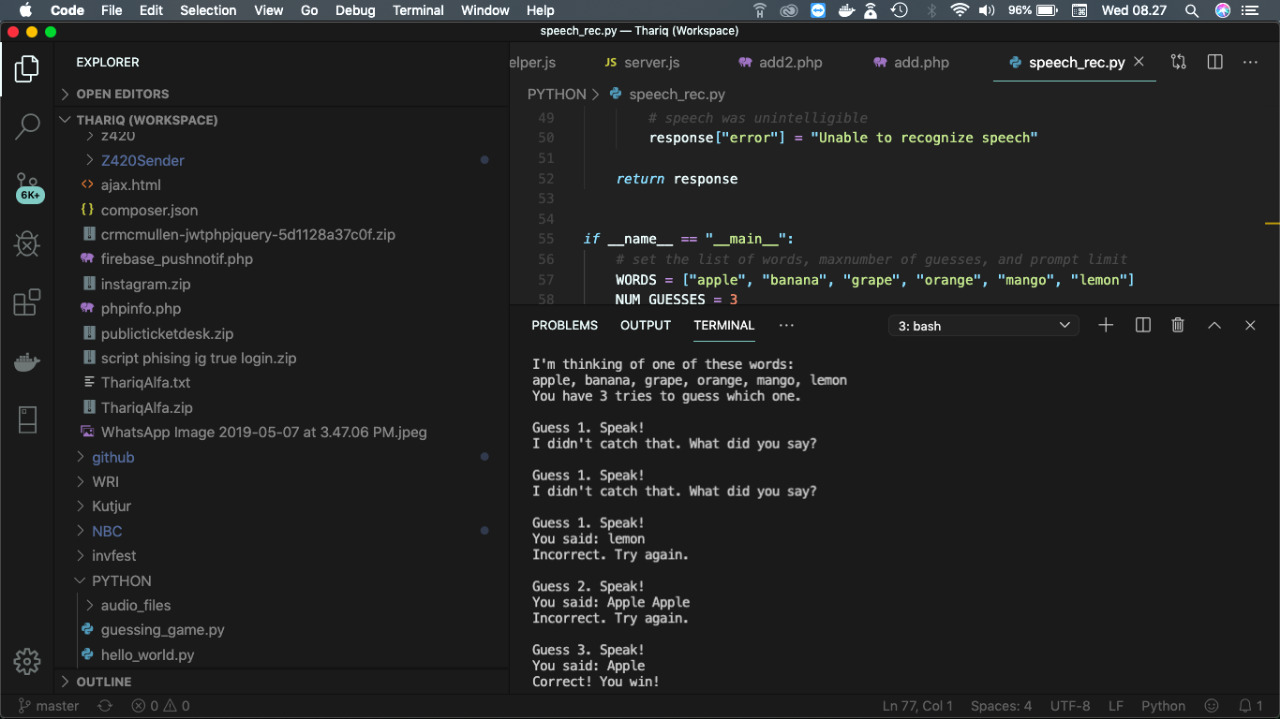
1. **Implementasi**











1. **Kesimpulan dan Saran**

**Kesimpulan:**

1. Aplikasi ini harus menggunakan suara yang jelas, keras.
2. Aplikasi ini memiliki beberapa daftar kata yaitu apple, banana, grape, orang, mango, lemon.
3. Aplikasi ini memiliki jumlah tebakan yaitu 3.
4. Aplikasi ini memiliki batas prompt yaitu 5.

**Saran :**

1. Aplikasi ini harus menggunakan suara yang jelas, keras. Karena jika tidak demikian maka muncul tampilan incorrect atau I didn’t catch that.
2. Aplikasi ini diharapkan berkembang lebih baik.
3. **Peran Anggota**

****

Nama : Lexy Fadillah Nanlohy

NIM : 1841720097

Peran : membuat project

****

Nama : Lili Nur Indah Sari

NIM : 1841720037

Peran : membuat laporan

****

Nama : Raka Arya Alzidan W.

NIM : 1841720077

Peran : membuat laporan

****

Nama : R.B Dwi Rizky A.

NIM : 1841720181

Peran : membuat laporan

****

Nama : Thariq Alfa B.

NIM : 1841720165

Peran : membuat project