

YouTube Video Converter

Deskripsi

Kumpulan API untuk mengunduh file mp3 maupun mp4 dari URL YouTube yang diberikan. Service akan dijalankan secara asynchronous menggunakan multithreading dan hanya akan bisa didownload ketika hasilnya sudah selesai.

Tingkat Urgensi

Semua orang pasti pernah menggunakan YouTube baik untuk menonton video, maupun untuk mendengarkan musik. Namun, pasti kita pernah ingin menyimpan video atau musik tersebut ke dalam perangkat yang kita miliki. YouTube menyediakan opsi untuk mendownload, tetapi tidak semua video dapat didownload dan bahkan memiliki batas waktu penyimpanan. Disinilah custom video converter hadir untuk menyelesaikan masalah tersebut.

Memang sudah ada beberapa converter yang tersedia, tetapi banyak sekali iklan yang muncul pada website yang ada dan bisa saja tidak aman. Mungkin saja kita diberikan file yang lain berupa virus atau malware dan hal-hal buruk lainnya. Service yang saya sediakan terjamin keamanannya dan dapat dijalankan tanpa adanya iklan sama sekali. Selain cepat, jika video sudah pernah diproses oleh orang lain sebelumnya, tidak perlu dilakukan proses komputasi tambahan.

Dokumentasi Postman

Untuk mencoba seluruh **endpoint** yang tersedia, silakan akses [dokumentasi berikut](#). Disana sudah tersedia variabel **URL** yang dapat dikonfigurasi dan contoh interaksi beserta responnya yang tersedia untuk setiap **endpoint**.

Source Code

Selain terlampir pada slot submisi, saya juga mengunggah hasil pekerjaan saya pada github yang dapat diakses pada [tautan berikut](#).

Terkait Program

Seluruh service dibuat menggunakan **Django REST API**. Konfigurasi *database* akan memanfaatkan **db.sqlite3**. Program memanfaatkan **Multithreading** untuk membuat banyak proses di balik layar secara langsung agar bisa berjalan secara asynchronous.

- Untuk menjalankan program pada mesin lokal, jalankan perintah

```
# Membuat virtual environment
python3 -m venv env
```

Service dapat diakses melalui **localhost:8000**

- Untuk menjalankan program pada server, cukup jalankan

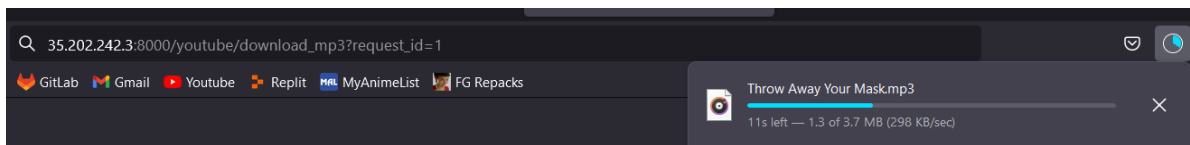
```
bash deploy.sh
```

Service dapat diakses melalui **<URL>:8000**

Service

Silakan akses dokumentasi postman untuk keterbacaan yang lebih baik. Sudah ada contoh interaksi dan response yang dapat menjadi panduan dalam menggunakan service. Dokumentasi postman dapat diakses pada [dokumentasi berikut](#)

Tipe respon untuk video maupun audio berupa **FileResponse** sehingga, ketika dibuka pada browser akan mentrigger-opsi untuk mendownload.



Process Audio

/youtube/process_audio	POST
Memberikan URL YouTube untuk diproses menjadi mp3. Pemrosesan akan dilakukan secara asynchronous	body: url
	response: 200 on success or exists

Check Status

/youtube/status	GET
Mengecek status proses konversi berdasarkan ID yang diberikan pada endpoint sebelumnya	query params: request_id
	response: 200 on success or not done 404 if not exists

Download Audio

/youtube/download_mp3	GET
Mendownload hasil konversi yang telah dilakukan	query params:

berdasarkan ID yang diberikan dalam format mp3	request_id
	response: 200 on success or not done 404 if not exists

Process Video

/youtube/process_video	POST
Memberikan URL YouTube untuk diproses menjadi mp4. Pemrosesan akan dilakukan secara asynchronous	body: url
	response: 200 on success or exists

Check Status

/youtube/status	GET
Mengecek status proses konversi berdasarkan ID yang diberikan pada endpoint sebelumnya	query params: request_id
	response: 200 on success or not done 404 if not exists

Download Video

/youtube/download_mp4	GET
Mendownload hasil konversi yang telah dilakukan berdasarkan ID yang diberikan dalam format mp3	query params: request_id
	response: 200 on success or not done 404 if not exists

Kesulitan

Service ini dibuat menggunakan beberapa library dan file processing untuk mengatur file yang di download dan disimpan. Proses untuk melakukan download, memastikan proses berjalan dengan lancar tidaklah mudah. Disini penulisan kode tidak dilakukan secara konvensional, melainkan memperhatikan aspek **multithreading** yang dilakukan.

Mengirim file menggunakan API juga tidak mudah. Untuk mengirim baik video, maupun audio harus memberikan informasi tambahan pada header dan mengatur respon agar dapat membawa file yang bersangkutan saat responnya dikirim. Selain itu, program didesain untuk memberikan respon untuk segera mendownload file secara otomatis ketika **endpoint** yang sesuai diakses.

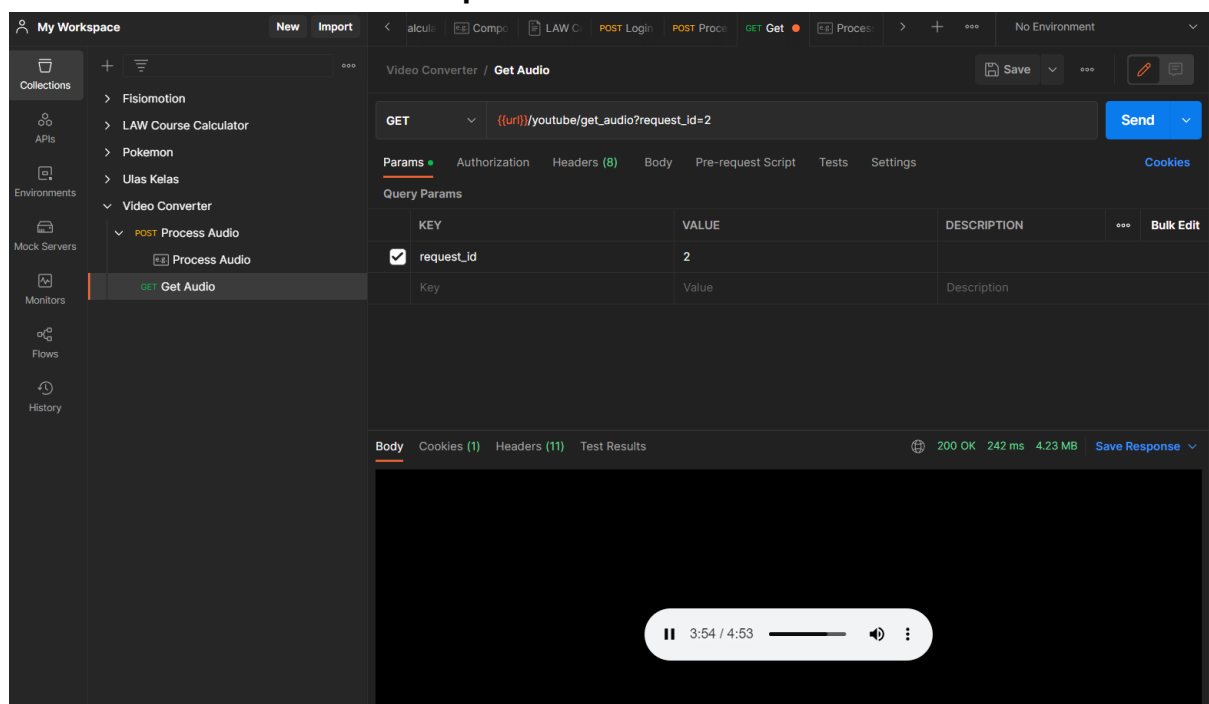
Mengatur filepath, membuka file untuk dikirim juga tidak mudah, harus dilakukan riset terlebih dahulu terkait pemrosesan file dimana seperti yang kita tahu, hampir semua orang tidak suka berurusan dengan **filesystem**.

Semua **endpoint** terproteksi dari method yang tidak sesuai dan request yang buruk sehingga program sudah pasti memberikan respon yang sesuai dengan interaksi yang diberikan

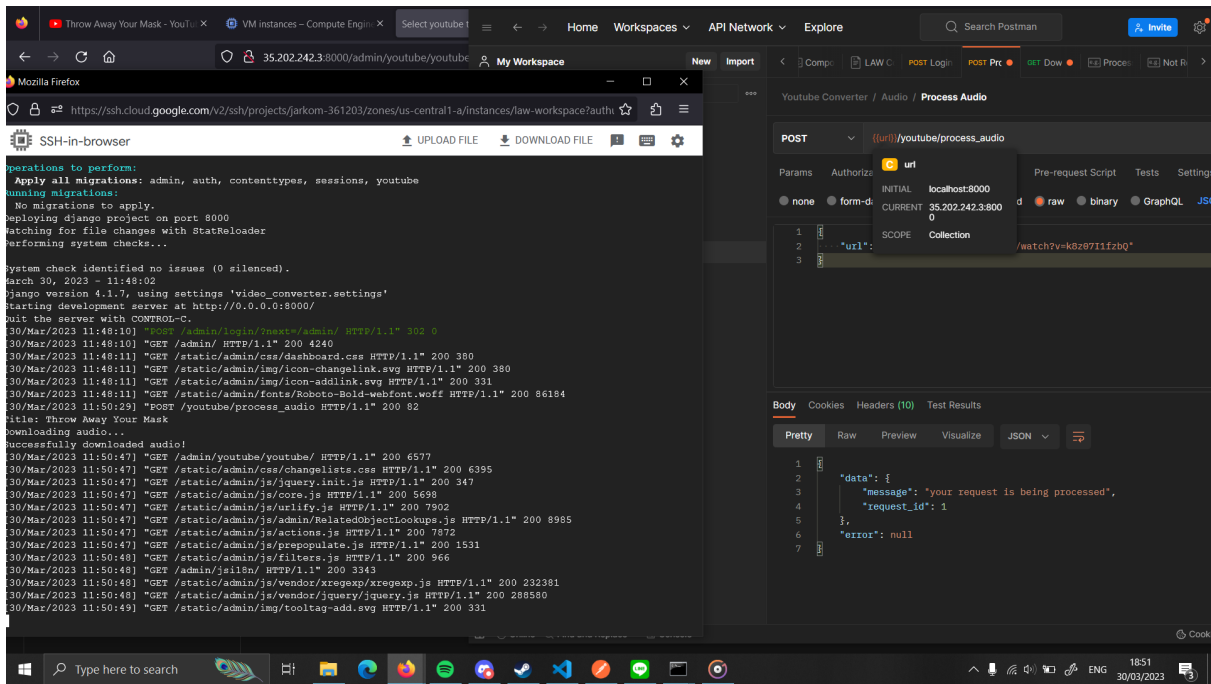
Lampiran

Tidak semua, tetapi saya lampirkan beberapa foto yang sudah cukup representatif untuk memperlihatkan bukti pengerjaan. Pada gambar juga terlihat **URL** yang saya akses (merupakan IP GCP) dan tangkapan layar mesin GCP menggunakan SSH-in browser.

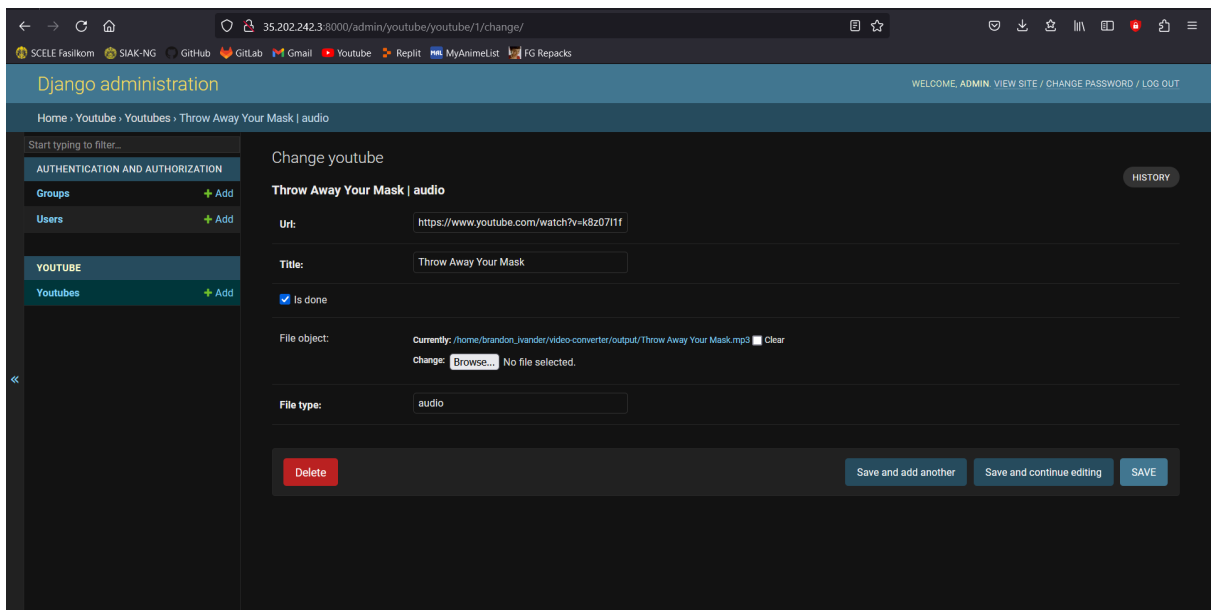
1. File Audio berhasil didapatkan dari server



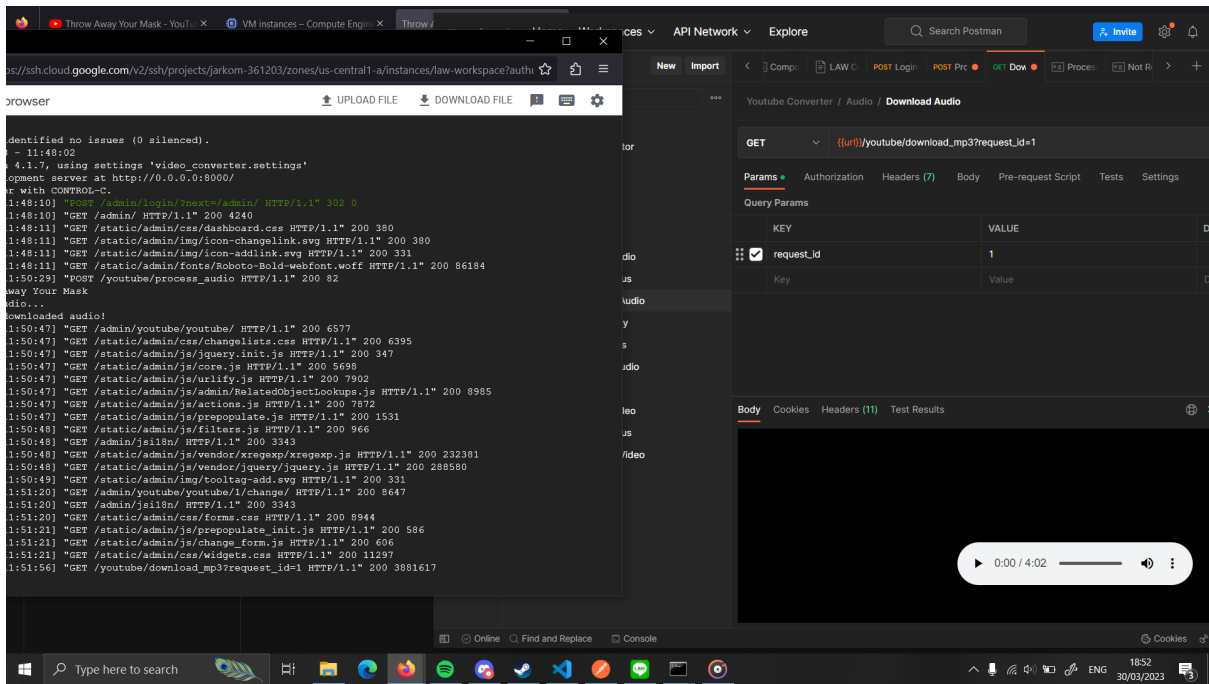
2. Pembuatan Request dan tampilannya pada mesin GCP



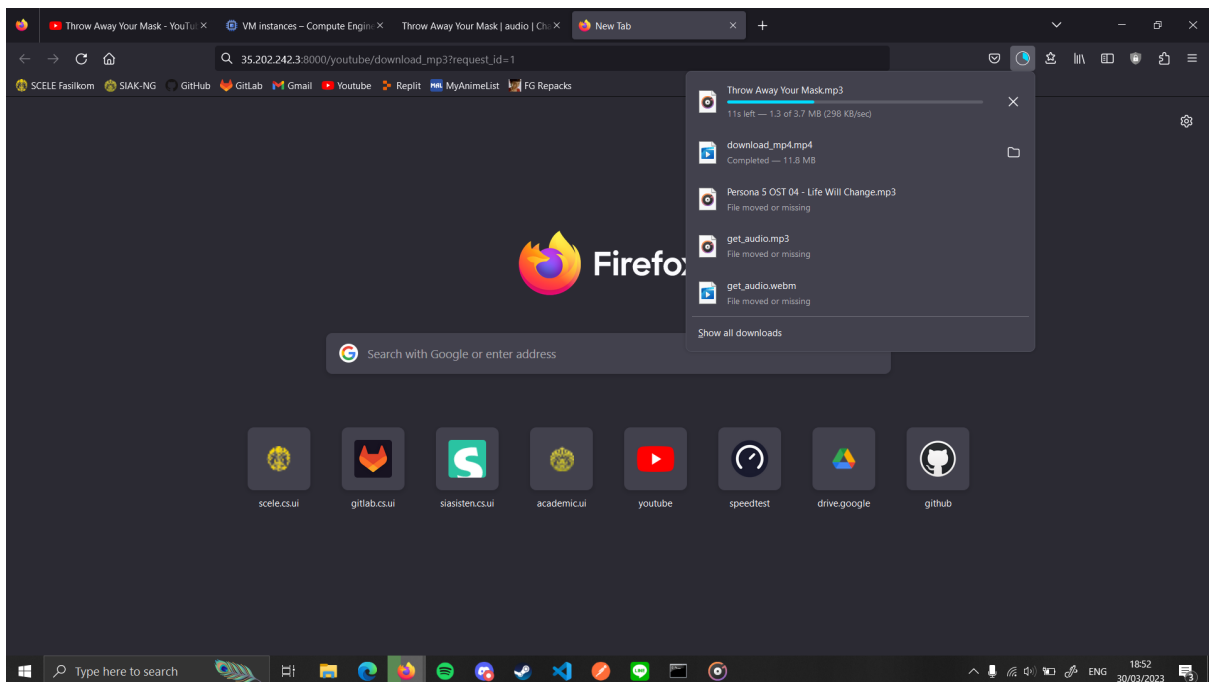
3. Hasil yang tersimpan pada Django Admin



4. Output Audio dari Request



5. Auto-download Ketika Diakses Menggunakan Browser



6. File Audio yang Tersimpan di VM

The image displays two side-by-side terminal windows. The left window, titled 'SSH-in-browser', shows a system check with no issues, a Django version 4.1.7, and a starting development server at http://0.0.0.0:8000/. It then shows a series of GET requests to various endpoints like /static/admin/js/jquery.init.js, /static/admin/js/core.js, /static/admin/js/uriutils.js, /static/admin/js/actions.js, /static/admin/js/prepopulate.js, /static/admin/js/filters.js, /static/admin/js/i18n/, /static/admin/js/vendor/jquery/jquery.js, /static/admin/js/tooltag-add.svg, /static/admin/js/change_form.js, /static/admin/js/widgets.js, and a download request for mp3request_id=1. The right window, also titled 'SSH-in-browser', shows system information as of Thu Mar 30 11:53:41 UTC 2023, including system load, memory usage, and swap usage. It also lists management and support URLs, and mentions expanded security maintenance for applications, which is not enabled. Finally, it shows a last login time and a series of commands related to downloading and managing video converter packages.