

## PANDUAN PENGGUNA

*Endurance Testing*, juga dikenal sebagai *soak testing*, melibatkan pengujian suatu aplikasi dengan memberikan beban secara terus-menerus dalam jangka waktu yang lama. Metode ini bertujuan untuk mengungkap kebocoran memori, kehabisan sumber daya, dan berbagai masalah degradasi kinerja yang mungkin hanya muncul setelah penggunaan dalam waktu yang lama.

*Software Endurance Testing* adalah alat untuk mengevaluasi kinerja dan stabilitas aplikasi web melalui pengujian beban secara berkelanjutan. Aplikasi ini memungkinkan pengguna untuk mengirimkan beberapa permintaan HTTP secara bersamaan ke URL yang ditentukan, sambil memantau berbagai metrik kinerja seperti *load time*, *wait time*, *response time*, *throughput*, dan *error rate*.

### A. Sistem

Berikut adalah panduan penggunaan *Software Endurance Testing*. Sebelum menggunakan *tool* ini, pastikan bahwa koneksi internet dalam kondisi stabil untuk memastikan kelancaran pengujian.

The screenshot shows the 'Endurance Testing' web application interface. At the top, the title 'Endurance Testing' is centered. Below it, there are input fields for 'URL:' (containing 'https://example.com'), 'Number of Request (Min and Max):' (with two empty input boxes), and 'Mode:' (a dropdown menu set to 'Stable'). Below these are 'Timeout Per-Round (In Seconds):' (empty), 'Time in Period:' (empty), and radio buttons for 'Second(s)', 'Minute(s)', and 'Hour(s)'. To the right of these are 'Gemini API Key:' (empty) and 'Discord Webhook URL:' (empty). Below the input fields are two buttons: a green 'Start' button and a red 'Stop' button. In the center, a digital timer displays '00:00:00:00'. Below the timer is a large 'Output:' text area with a scrollable list of results. On the right side of the output area, there are four buttons: 'Help' (teal), 'Info' (blue), 'Clear' (grey), and 'Export' (orange).

1. Masukkan URL target yang akan diuji pada kolom bertanda **“URL”**, dengan memastikan format yang digunakan sudah benar, yaitu menggunakan protokol `http://` atau `https://`.
2. Tentukan jumlah minimum *request* yang akan dikirimkan pada setiap *round* pengujian melalui kolom **“Number of Request (Min and Max)”**, dengan batas maksimum 1000 *request*. Jika memilih mode *Progressive* atau *Fluctuative*, tentukan juga jumlah maksimum *request* per *round* (tidak boleh sama dengan atau kurang dari minimum *request*).
3. Pilih mode pengujian pada menu dropdown **“Mode”**:
  - *Stable*: Mengirimkan jumlah *request* yang tetap pada setiap *round* pengujian.
  - *Progressive*: Secara bertahap meningkatkan jumlah *request* dari waktu ke waktu.
  - *Fluctuative*: Mengirimkan jumlah *request* yang bervariasi secara acak dalam rentang minimum dan maksimum *request* yang telah ditentukan.
4. Tentukan ambang waktu tunggu (*timeout*) untuk setiap *round* pengujian pada kolom **“Timeout Per-Round (In Seconds)”**.
5. Tetapkan durasi pengujian pada kolom **“Time in Period”**, lalu pilih satuan waktu yang sesuai (second, minute, atau hour).
6. (Opsional) Masukkan Gemini API Key pada kolom **“Gemini API Key”** untuk mengaktifkan fitur penyajian ringkasan hasil pengujian secara deskriptif berbasis AI.
7. (Opsional) Masukkan Discord Webhook URL pada kolom **“Discord Webhook URL”** agar mendapatkan notifikasi pada server Discord setelah pengujian selesai secara otomatis.
8. Klik tombol **“Start”** untuk memulai pengujian.
9. Klik tombol **“Stop”** jika ingin menghentikan pengujian ketika waktu hitung mundur belum selesai.
10. Pantau hasil pengujian pada *textbox* **“Output”**.
  - *Textbox* hanya menampilkan hasil dari *round* pengujian terbaru.
  - Untuk melihat data secara keseluruhan, ekspor hasil pengujian atau buka berkas *EnduranceTestLog.txt* di direktori yang sama dengan file *executable software* ini.
11. Setelah pengujian selesai, *software* akan menampilkan ringkasan hasil pengujian yang mencakup:
  - *Total Requests*: Jumlah total *request* yang dikirim selama pengujian.

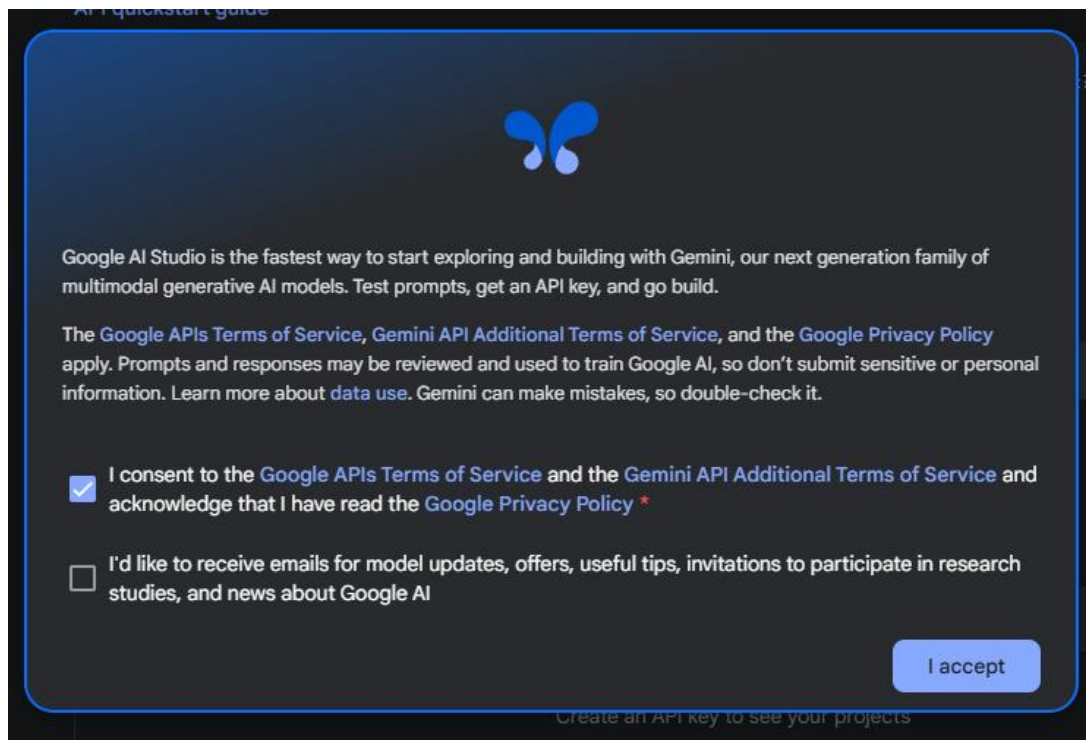
- *Successful Requests*: Jumlah *request* yang berhasil menerima *response* HTTP 200 (OK).
  - *Failed Requests*: Jumlah *request* yang gagal menerima *response* HTTP 200 (OK) atau mengalami waktu habis (*timeout*).
  - *Average Computer's CPU Usage*: Rata-rata penggunaan CPU komputer selama pengujian.
  - *Average Computer's RAM Usage*: Rata-rata penggunaan RAM komputer dalam *megabyte* selama pengujian.
  - *Average Load Time*: Waktu rata-rata yang dibutuhkan untuk menerima *request* secara penuh.
  - *Average Wait Time*: Waktu rata-rata hingga menerima *byte* pertama setelah mengirim *request*.
  - *Average Response Time*: Waktu rata-rata waktu respons untuk semua *request*.
  - *Average Throughput*: Jumlah rata-rata *request* yang diproses per detik.
  - *Average Error Rate*: Persentase *request* yang gagal atau mengalami *timeout*.
  - *Average Round Duration*: Waktu rata-rata untuk menyelesaikan satu *round* pengujian.
  - *AI Analysis*: Jika Gemini API Key dimasukkan, ringkasan hasil pengujian secara deskriptif akan ditampilkan.
12. Klik tombol **“Clear”** untuk mengatur ulang parameter pengujian dan menghapus hasil pengujian.
13. Klik tombol **“Export”** untuk menyimpan hasil pengujian dalam format:
- Excel (.xlsx)
  - CSV (.csv)
  - JSON (.json)
  - HTML (.html)
  - Discord (ringkasan hasil pengujian saja)

## B. Gemini API Key

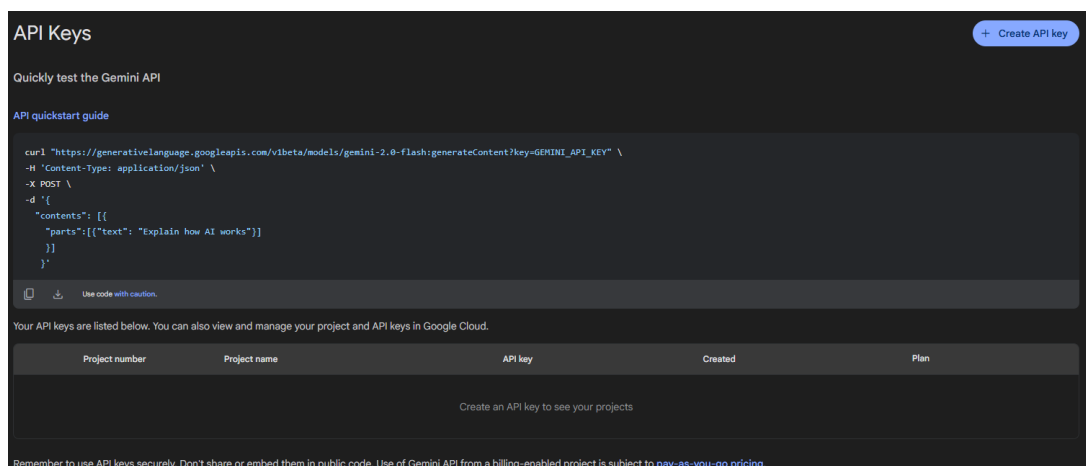
Berikut adalah panduan mendapatkan Gemini API Key:

1. Buat akun google terlebih dahulu jika belum memiliki akun. Kemudian, kunjungi laman web <https://aistudio.google.com/u/3/apikey>, lalu centang pilihan **“I consent to the Google APIs Terms of Service and the Gemini API Additional Terms of**

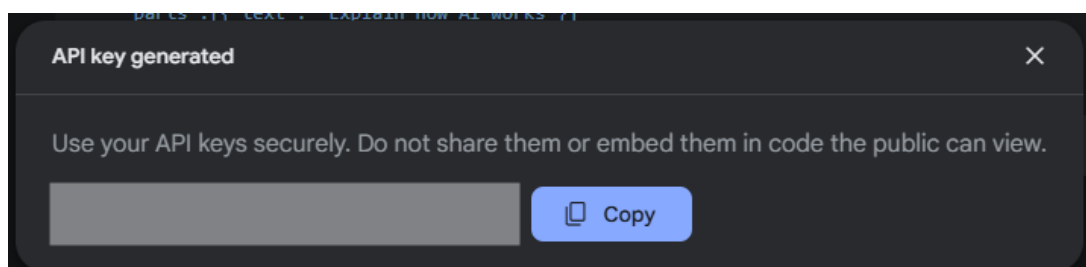
Service and acknowledge that I have read the Google Privacy Policy \*” dan klik tombol “I accept”.



2. Klik tombol “Create API key” dan tunggu proses *loading* selesai.



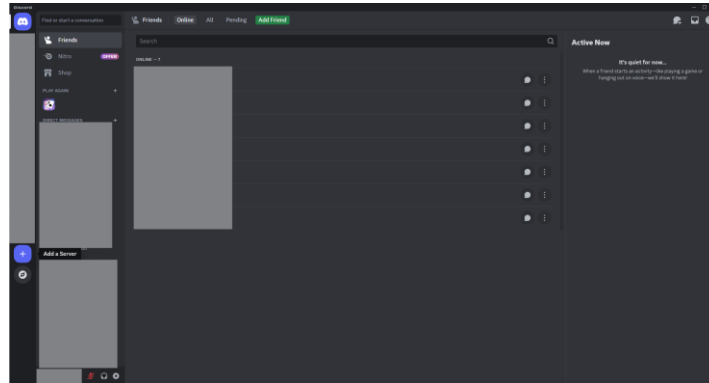
3. API key siap disalin dan digunakan.



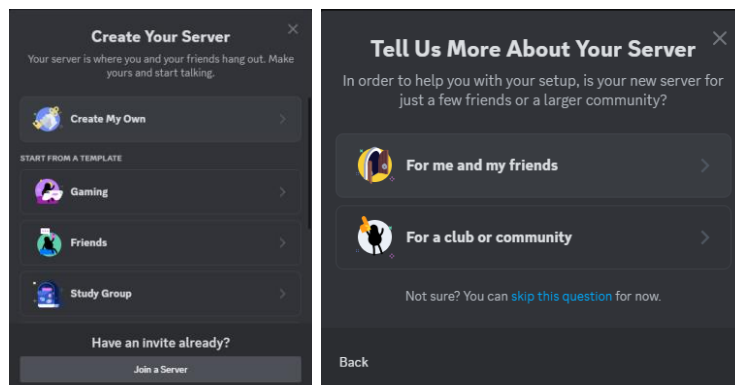
### C. Discord Webhook URL

Berikut adalah panduan mendapatkan Discord Webhook URL:

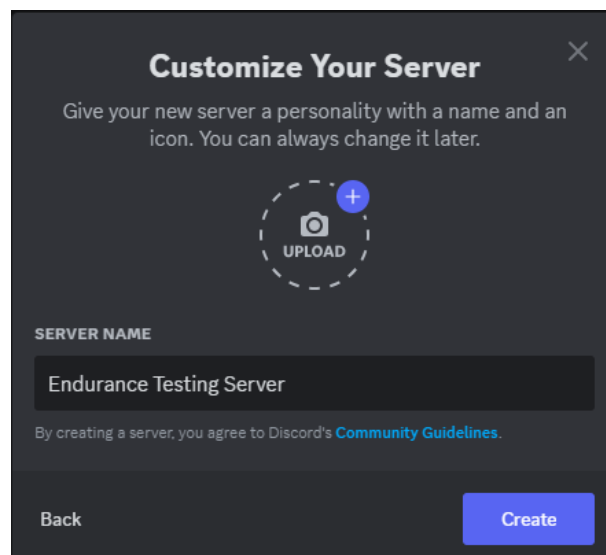
1. Buat server baru dengan menekan ikon **“Add a Server”** di sebelah kiri layar (langkah ini dapat dilewati jika sudah memiliki server).



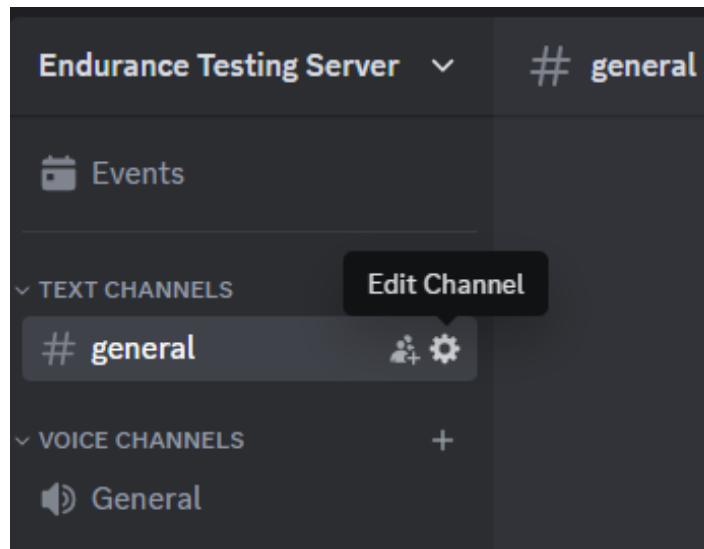
2. Pilih **“Create My Own”** dan **“For me and my friends”** atau sesuaikan dengan kebutuhan anda.



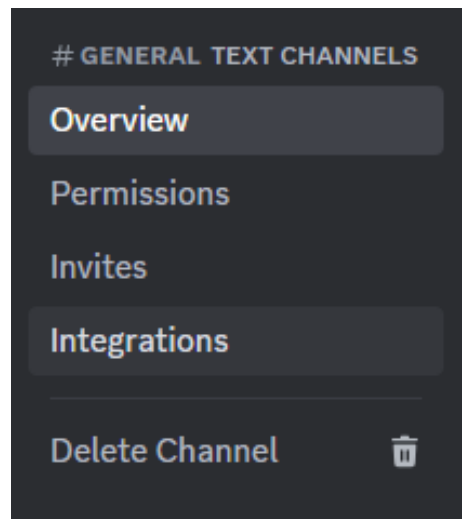
3. Berikan nama pada server anda dan klik **“Create”**.



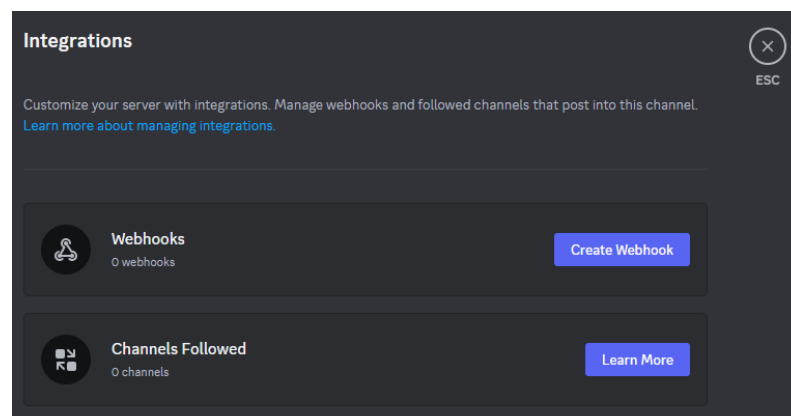
4. Klik ikon roda gigi pada bagian kiri server.



5. Pilih tab “Integrations”.



6. Klik “Create Webhook”.



## 7. Webhook URL siap disalin dan digunakan

