# AUTOMATED PERFORMANCE TESTING MENGGUNAKAN

**K6 (Studi Kasus : Komiqus.com)**

SKRIPSI

Sebagai salah satu syarat

untuk memperoleh gelar Sarjana Teknologi Informasi pada Program Studi Sarjana Teknologi Informasi Institut Bisnis dan Informatika Kesatuan

## Oleh:

**DENNY DOLOK PARTALA**

NPM: 192310004



## PROGRAM STUDI SARJANA TEKNOLOGI INFORMASI FAKULTAS INFORMATIKA DAN PARIWISATA INSTITUT BISNIS DAN INFORMATIKA KESATUAN

## BOGOR

**2023**

# AUTOMATED PERFORMANCE TESTING MENGGUNAKAN

**K6 (Studi Kasus: Komiqus.com)**

## SKRIPSI

Sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknologi Informasi

pada Program Studi Teknologi Informasi Institut Bisnis dan Informatika Kesatuan

## Oleh:

**DENNY DOLOK PARTALA**

NPM : 192310004

## PROGRAM STUDI TEKNOLOGI INFORMASI FAKULTAS INFORMATIKA DAN PARIWISATA INSTITUT BISNIS DAN INFORMATIKA KESATUAN

## BOGOR

**2023**

# AUTOMATED PERFORMANCE TESTING MENGGUNAKAN

**K6 (Studi Kasus : Komiqus.com)**

## SKRIPSI

Telah diuji dan disetujui pada Sidang Sarjana Institut Bisnis dan Informatika Kesatuan, pada:

|  |  |  |
| --- | --- | --- |
| Hari | : | Jumat |
| Tanggal | : | 18 Agustus 2023 |

Mengetahui,

|  |  |
| --- | --- |
| Dekan Fakultas Informatika dan Parawisata | Ketua Program Studi Teknologi Informasi |
| Dr. Ir. Jan Horas V. Purba, M.Si., CBPA®, CPSP® | Edi Nurachmad, S. Kom., M. Kom |

# AUTOMATED PERFORMANCE TESTING DENGAN

**MENGGUNAKAN K6 (Studi Kasus : Komiqus.com)**

## SKRIPSI

Telah disetujui oleh Pembimbing:

Febri Damatraseta, S.T., M.Kom.

Telah diujikan pada Sidang Sarjana dan dinyatakan LULUS Pada tanggal seperti tertera di bawah ini

Bogor, 18 Agustus 2023

Dosen Penguji

|  |  |
| --- | --- |
| Penguji I, | Penguji II, |
| Dr. Ir. Nusa Muktiadji, M.M. | Septian Cahyadi, S.Kom., M.Kom. |

# SURAT PERNYATAAN

Yang bertanda tangan dibawah ini:

|  |  |  |
| --- | --- | --- |
| Nama | : | Denny Dolok Partala |
| NPM | : | 192310004 |
| Program Studi | : | Teknologi Informasi |
| Judul Skripsi | : | Automated Performance Testing Menggunakan  K6 (Studi Kasus: Komiqus.com) |

Menyatakan dengan sebenar – benarnya bahwa segala pernyataan dalam Skripsi ini adalah hasil perkerjaan saya sendiri, dengan bimbingan Ketua Pembimbing, kecuali yang jelas ditunjukan rujukannya. Skripsi ini belum pernah diajukan untuk memperoleh gelar pada program sejenis di Perguruan Tinggi lain. Semua sumber data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Jika ternyata terbukti pernyataan ini tidak benar, sepenuhnya menjadi tanggung jawab saya.

Bogor, 18 Agustus 2023 Yang Menyatakan,

Denny Dolok Partala

# ABSTRAK

DENNY DOLOK PARTALA, NPM: 192310004 (2023), Automated Performance Testing Menggunakan K6. Dibawah Bimbingan FEBRI DAMATRASETA.

Komiqus merupakan salah satu website yang bergerak dibidang sosial katalog, dimana tantangan terbesar dalam membangun sebuah website sosial media ialah menangani performa stabilitas websitenya untuk menjaga jumlah akses dalam waktu tertentu agar dapat terus berjalan dan melayani setiap permintaan. Metode *Automated Performance Test* digunakan untuk menguji performa website yang bersifat *non-functional*. Model *Load Testing* digunakan untuk mengukur seberapa baiknya website Komiqus bekerja jika memiliki beban kerja yang cukup banyak dalam waktu tertentu. K6 digunakan dalam penelitian ini sebagai media *software tools* untuk model *Load Testing*. Pengujian ini menggunakan jumlah beban VUs (*Virtual Users*) dimulai dari 1, 30, 50, 100 dan 300 yang dilakukan pada tiga halaman website Komiqus yaitu halaman utama, halaman katalog dan halaman profil kreator. Dari hasil penelitian ini dapat disimpulkan bahwa performa website Komiqus memiliki kriteria buruk, nilai ini diperoleh berdasarkan *GMatrix report* dengan nilai F pada halaman utama, D pada halaman katalog dan E pada halaman profil kreator dengan waktu respon tertinggi yang didapat adalah 1 menit pada saat pengujian dengan beban 300 VUs yang dilakukan pada halaman profil kreator dan waktu tercepat yang didapat adalah dibawah 2 detik pada saat pengujian dengan beban 1 VUs pada semua halaman yang diuji, hasil tersebut didapati melalui pengujian menggunakan K6.

Kata Kunci: Performance Testing, Load Testing, K6, Automated Performance Testing

# ABSTRACT

DENNY DOLOK PARTALA, NPM: 192310004 (2023), Automated Performance Testing Using K6, Under Guidance of FEBRI DAMATRASETA.

*Komiqus is a website engaged in the social catalog, where the biggest challenge dalam building a social media website is handling the stability of the pekerjaan to maintain the number of accesses in a certain time so that it can continue to run and serve every request. The Automated Performance Test method is used to test the performance of non-functional websites. The Load Testing model is used to measure how well the Komiqus website works if it has quite a lot of workload in a certain time. K6 is used in this study as a media software tool for the Load Testing model. This test uses the number of VUs (Virtual Users) loads starting from 1, 30, 50, 100 and 300 which are carried out on three Komiqus website pages, namely the main page, catalog page and creator profile page. From the results of this study it can be concluded that Komiqus website performance has poor criteria, this value is obtained based on the GMatrix report with a value of F on the main page, D on the catalog page and E on the creator's profile page with the highest response time obtained is 1 minute during testing with a load of 300 VUs was carried out on the creator's profile page and the fastest time obtained was under 2 seconds during testing with a load of 1 VUs on all pages tested, these results were found through testing using K6.*

Keywords: *Performance Testing, Load Testing, K6, Automated Performance Testing*

# KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kehadirat Tuhan Yang Maha Esa atas berkat rahmat dan kasih karuniaNya sehingga penulis dapat meyelesaikan penulisan skripsi yang berjudul “Automated Performance Testing Menggunakan K6 (Studi Kasus : Komiqus.com)” ini dengan baik dan tepat waktu.

Pada kesempatan kali ini, penulis menyampaikan terima kasih kepada:

1. Bapak Prof. Dr. Bambang Pamungkas, Ak,. MBA, CA, CPA, CPA (Aust), ASEAN CPA, CIMBA, CSFA, CFrA, CGAE. selaku Rektor IBI Kesatuan.
2. Bapak Dr. Ir, Jan Horas Veryady Purba, M.Si., CBPA®, CPSP®. selaku Dekan Fakultas Informatika dan Pariwisata IBI Kesatuan.
3. Bapak Edi Nurachmad, S.Kom., M.Kom. selaku Ketua Program Studi Teknologi Informasi IBI Kesatuan Bogor.
4. Bapak Ferdiyan Feisal, S.Kom. selaku CEO website Komiqus.
5. Bapak Febri Damatraseta, S.T., M.Kom. selaku Dosen pembimbing penulis.
6. Bapak Septian Cahyadi, S.Kom., M.Kom. dan Bapak Dr. Ir. Nusa Muktiadji,

M.M. selaku Penguji Sidang Skripsi.

1. Staff Tata Usaha dan Perpustakaan Institut Bisnis dan Informatika Kesatuan atas pengabdiannya pada fakultas tercinta.
2. Rekan-rekan kelas karyawan angkatan 2019 yang selalu membantu satu sama lain selama masa penyusunan skripsi.

Terakhir, penulis menyampaikan terima kasih kepada kedua orang tua penulis, mendiang Bapak Kisa Silaen dan Ibu Ria Sitorus, yang telah memberikan perhatian kepada saya.

Kemudian, Penulis menyadari bahwa skripsi ini masih sangat jauh dari kata sempurna. Oleh karena itu, kritik dan saran yang membangun akan penulis terima demi kesempurnaan skripsi ini.

Bogor, Agustus 2023

Penulis

# DAFTAR ISI

|  |  |  |
| --- | --- | --- |
| ABSTRAK | .................................................................................................. | i |
| *ABSTRACT* | .................................................................................................. | ii |
| KATA PENGANTAR .................................................................................... | | iii |
| DAFTAR ISI .................................................................................................. | | iv |
| DAFTAR TABEL ........................................................................................... | | vi |
| DAFTAR GAMBAR ...................................................................................... | | vii |
| BAB I PENDAHULUAN ............................................................................. | | 1 |
| 1.1 Latar Belakang Penelitian......................................................... | | 1 |
| 1.2 Rumusan Masalah .................................................................... | | 2 |
| 1.3 Tujuan Penelitian...................................................................... | | 2 |
| 1.4 Manfaat..................................................................................... | | 2 |
| 1.5 Batasan Masalah ....................................................................... | | 3 |
| BAB II TINJAUAN PUSTAKA .................................................................... | | 4 |
| 2.1 Dasar Teori ............................................................................... | | 4 |
| 2.1.1 Performance Testing........................................................ | | 4 |
| 2.1.2 STLC (*Software Testing Life Cycle*) ............................... | | 5 |
| 2.1.3 Komiqus .......................................................................... | | 6 |
| 2.1.4 Go .................................................................................... | | 6 |
| 2.1.5 K6 .................................................................................... | | 7 |
| 2.1.6 Cloudlets.......................................................................... | | 7 |
| 2.2 Penelitian Sebelumnya ............................................................. | | 7 |
| BAB III METODOLOGI PENELITIAN......................................................... | | 10 |
| 3.1 Waktu dan Tempat Penelitian .................................................. | | 10 |
| 3.2 Alat dan Bahan ......................................................................... | | 10 |
| 3.3 Prosedur Kerja .......................................................................... | | 11 |
| 3.3.1 Metode Penelitian ............................................................ | | 11 |
| 3.3.2 Metode *Load Testing*....................................................... | | 14 |
| BAB IV PEMBAHASAN HASIL PENELITIAN........................................... | | 15 |
| 4.1 Pendahuluan ............................................................................. | | 15 |
| 4.2 *Requirement Analysis* ............................................................... | | 15 |
| 4.2.1 Objek Penelitian .............................................................. | | 16 |
| 4.3 Analisa *Test Planning*............................................................... | | 24 |
| 4.3.1 Analisa Data Input ........................................................... | | 24 |
| 4.3.2 Arsitektur K6 ................................................................... | | 25 |
| 4.4 *Test Case Development & Environment Setup*......................... | | 26 |
| 4.4.1 Input................................................................................. | | 26 |
| 4.4.2 Algoritma K6...................................................................  4.4.3 Pengaturan Pengujian *Automated Performance*  *Testing* ............................................................................. | | 28  28 |
| 4.3 Test Execution .......................................................................... | | 32 |
| 4.3.1 Evaluasi Hasil Pengujian *Local Environment* ................. | | 32 |
| 4.3.2 Evaluasi Hasil Pengujian *Real-World Environment*........ | | 34 |

* 1. [Test Closure 36](#_TOC_250005)
  2. [Rangkuman 41](#_TOC_250004)

[BAB V KESIMPULAN DAN SARAN 42](#_TOC_250003)

* 1. [Kesimpulan 42](#_TOC_250002)
  2. [Saran 43](#_TOC_250001)

[DAFTAR PUSTAKA 45](#_TOC_250000)

**DAFTAR TABEL**

Tabel 2.1 Penelitian Sebelumnya 8

Tabel 3.1 Waktu Penelitian 10

Tabel 3.2 Test Planning 12

Tabel 3.3 Test Case Development 13

Tabel 4.1 Atribut Halaman Utama 18

Tabel 4.2 Atribut Halaman Katalog 20

Tabel 4.3 Atribut Halaman Profil Kreator 23

# DAFTAR GAMBAR

Gambar 2.1 STLC 5

Gambar 2.2 Halaman Utama Komiqus.com 6

Gambar 3.1 Flowchart Metode Penelitian 11

Gambar 4.1 Diagram Package Proses Bisnis Halaman Utama 16

Gambar 4.2 Gambar Halaman Utama Komiqus.com 17

Gambar 4.3 Diagram Package Proses Bisnis Halaman Katalog 18

Gambar 4.4 Halaman Katalog Komik Komiqus.com 19

Gambar 4.5 Proses Bisnis Halaman Profil Kreator 21

Gambar 4.6 Gambar Halaman Profil Kreator Komiqus.com 22

Gambar 4.7 Arsitektur K6 25

Gambar 4.8 Laporan Hasil Load Test 26

Gambar 4.9 Screenshot syntax kode input VUs 27

Gambar 4.10 Screenshot syntax kode input URL 27

Gambar 4.11 Syntax Kode Load Test 28

Gambar 4.12 Screenshot Hasil Pengujian Halaman Utama 29

Gambar 4.13 Screenshot Hasil Pengujian Halaman Katalog 30

Gambar 4.14 Screenshot Hasil Pengujian Halaman Profil Kreator 31

Gambar 4.15 Hasil Pengujian Halaman Utama 33

Gambar 4.16 Hasil Pengujian Halaman Katalog 33

Gambar 4.17 Hasil Pengujian Halaman Profil Kreator 34

Gambar 4.18 Diagram Activity Program 35

Gambar 4.19 Hasil Testing Real-world Halaman Utama Dengan 1 VUs 36

Gambar 4.20 Hasil Testing Real-world Halaman Utama Dengan 30 VUs 36

Gambar 4.21 Hasil Testing Real-world Halaman Utama Dengan 50 VUs 37

Gambar 4.22 Hasil Testing Real-world Halaman Utama Dengan 100 VUs 37

Gambar 4.23 Hasil Testing Real-world Halaman Utama Dengan 300 VUs 37

Gambar 4.24 Hasil Testing Real-world Halaman Katalog Dengan 1 VUs 38

Gambar 4.25 Hasil Testing Real-world Halaman Katalog Dengan 30 VUs 38

Gambar 4.26 Hasil Testing Real-world Halaman Katalog Dengan 50 VUs 38

Gambar 4.27 Hasil Testing Real-world Halaman Katalog Dengan 100 VUs 39

Gambar 4.28 Hasil Testing Real-world Halaman Katalog Dengan 300 VUs 39

Gambar 4.29 Hasil Testing Real-world Halaman Profil

Kreator Dengan 1 VUs 39

Gambar 4.30 Hasil Testing Real-world Halaman Profil

Kreator Dengan 30 VUs 40

Gambar 4.31 Hasil Testing Real-world Halaman Profil

Kreator Dengan 50 VUs 40

Gambar 4.32 Hasil Testing Real-world Halaman Profil

Kreator Dengan 100 VUs 40

Gambar 4.33 Hasil Testing Real-world Halaman Profil

Kreator Dengan 300 VUs 41

**BAB I PENDAHULUAN**

## Latar Belakang Penelitian

Dalam membangun sebuah *social media platform* banyak hal yang harus diperhatikan dari segi keamanan, kenyamanan dan lain – lain. Dan salah satu diantaranya adalah performa. Performa dari sebuah *social media platform* mempunyai pengaruh yang besar karena apabila *social media platform* tidak bisa melayani pengguna yang banyak disaat bersamaan pastinya kenyamanan pengguna dan proses bisnis akan terganggu juga.

Komiqus adalah website *social cataloguing* dan *social media platform* yang berfokus pada peminat serta penulis buku, komik dan novel di Indonesia. Komiqus sendiri dibuat untuk mengatasi masalah yang yang muncul karena judul-judul buku dan komik buatan anak bangsa yang tidak dikenal oleh banyak orang, tidak adanya sosialisasi mengenai hasil karya anak bangsa dan tidak adanya tempat untuk menampung info-info mengenai buku-buku serta komik-komik yang sudah ada. Komiqus bisa diakses melalui link <http://komiqus.com/>.

Dalam pembuatan sebuah aplikasi yang akan digunakan oleh konsumen atau pengguna, diperlukan berbagai jenis tes uji yang dapat memastikan atau menjamin bahwa aplikasi tersebut bisa dan dapat digunakan dengan baik. Salah satunya adalah *Performance Testing,* tes ini berguna untuk mengukur performa sebuah website aplikasi sehingga bisa diketahui apakah website aplikasi yang telah dibuat memiliki performa yang baik dan tidak akan menggangu pengguna ketika menggunakan aplikasi yang telah dibuat. Oleh karena itu penulis memutuskan untuk melakukan pengujian *Peformance Testing* pada website Komiqus untuk mengetahui performanya bila dibawah tekanan banyak pengguna yang mengunjungi dalam satu waktu, sehingga apabila ditemukan masalah pada bagian performa, proses mitigasi bisa dipersiapkan.

## Rumusan Masalah

Rumusan masalah penelitian ini adalah:

* + 1. Bagaimana performance dari website Komiqus jika dalam satu halaman dikunjungi oleh banyak pengguna ?
    2. Berapa lama waktu yang dibutuhkan dalam memproses satu halaman Komiqus dalam satu kali permintaan atau banyak permintaan?
    3. Bagaimana website Komiqus menjaga kestabilan sistemnya jika terjadi naik turunnya pengunjung secara terus menerus ?

## Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk :

* + 1. Mengetahui Stabilitas dan performa sistem ketika diakses oleh banyak pengguna.
    2. Menemukan batas performa sistem ketika berada dibawah tekanan sistem.
    3. Mengetahui daya tahan sistem dibawah tekanan beban dalam jangka waktu yang panjang*.*

## Manfaat

Manfaat dari penelitian ini adalah:

* + 1. Membantu dalam menemukan bug yang hanya bisa ditemukan ketika sistem diakses oleh banyak pengguna secara bersamaan di website Komiqus.
    2. Mengidentifikasi hambatan kinerja website Komiqus.
    3. Mengetahui batas performa dari website Komiqus.
    4. Mengembangkan pengetahuan penulis dalam menjalankan pengujian

*Performance Testing*.

* + 1. Mengembangkan pengalaman penulis dalam penggunaan bahasa pemrograman Go.

## Batasan Masalah

Beberapa batasan masalah yang harus ditetapkan dalam penelitian ini adalah sebagai berikut:

* + 1. Jumlah batas pengguna untuk Load testing hanya 300 pengunjung website berdasarkan kemampuan hardware penguji.
    2. Penerapan *Performance testing* hanya akan dilakukan pada halaman utama, halaman penulis dan halaman katalog.
    3. Waktu pengujian yang dilakukan untuk setiap testing adalah 5 menit hingga 15 menit.
    4. Pengujian performa pada aplikasi ini tidak membahas performa dari sisi

*security*.

# BAB II TINJAUAN PUSTAKA

## Dasar Teori

* + 1. **Performance Testing**

*Performance testing* merupakan sebuah proses menjalankan aplikasi atau sistem dengan melakukan simulasi pengguna sebenarnya dengan alat yang bisa membuat beban terhadap sistem agar bisa menemukan batasan dari sistem tersebut. Tujuan utama dari tes tersebut adalah untuk menemukan skalabilitas, ketersediaan dan juga performa dari sisi hardware maupun dari sisi software. Performance testing bisa juga digunakan digunakan dalam menemukan kejanggalan dalam functional yang hanya bisa ditemukan ketika sistem berada dibawah tekanan (Bhawna Jyoti, 2016). Dalam melakukan performance testing, aspek seperti *network bandwidth*, penggunaan CPU, penggunaan memori dan yang lain – lain bisa juga di monitor sebagai bagian dalam laporan performance testing. Ditambah lagi waktu respon, dan juga penggunaan yang berhubungan dengan web server, app server juga bisa dipertimbangkan sebagai performance testing. Jika digabungkan, performa sistem yang bisa dianggap sebagai bagian dari sudut respon waktu, jumlah data yang dikirim, ketersediaan, keandalan, keamanan, skalabilitas dan ekstensibilitas (H. Sarojadevi, 2011). Terdapat berbagai jenis performance testing yaitu : Stress testing, Spike Testing, Load testing, dan yang lain – lain (Doni Andriansyah, 2019).

*Automation Performance Testing* adalah tes yang dilakukan secara otomatis yang berguna untuk mengukur kecepatan halaman website selesai dimuat, respon waktu, keandalan dan juga skalabilitas dari sebuah aplikasi di bawah beban kerja yang diharapkan. Dengan dilakukan test ini secara otomatis maka para pengembang bisa dengan mudah dan cepat menemukan batasan performa dari sebuah aplikasi mengoptimalkan aplikasi tersebut. Banyak aplikasi yang bisa digunakan untuk melakukan *Automation Performance Testing* beberapa diantaranya adalah Apache Jmeter dan BlazeMeter.

*Load testing* merupakan salah satu tipe dari *non-functional testing* dan juga bagian dari *performance testing* yang dilakukan untuk mengetahui kinerja sebuah aplikasi / sistem ketika diakses oleh pengguna dengan jumlah yang banyak dan juga bagaimana aplikasi mengatasi trafficnya (Siddhant Shrivastava, 2020). *Load testing* secara sederhana digunakan untuk menentukan respon waktu sistem ketika test sedang dilakukan. Konkurensi isu beserta error fungsional ketika sistem sedang berada dibawah tekanan akan terdeteksi. Load testing dilakukan dengan cara memberikan sistem tugas tekanan yang berat sehingga batasan *hardware* / *software* dari utilitas sumber daya bisa ditemukan tanpa mengorbankan kinerja.

## STLC (Software testing Life Cycle)

Software testing life cycle merupakan sekuel dari aktivitas yang dilakukan dalam melakukan software testing secara sistematis dan terencana. Didalam STLC aktifitas – aktifitas berbeda dilakukan untuk meningkatkan kualitas dari sebuah produk. STLC sendiri merupakan bagian dari SDLC (Software Development Life Cycle). Fase – fase yang ada di dalam STLC adalah sebagai berikut (Dimas Rizky, 2019).

1.

Requirement Analysis

3. Test Case Development

& Environtment Setup



* + 1. Test Closure

2. Test Planning

4. Test Execution

Gambar 2.1 STLC

## Komiqus

Komiqus merupakan website social cataloguing yang berfokus pada peminat serta penulis buku, komik dan novel di Indonesia. Komiqus sendiri dibuat untuk mengatasi masalah yang yang muncul karena judul – judul buku dan komik buatan anak bangsa yang tidak dikenal oleh banyak orang, tidak adanya sosialisasi mengenai hasil karya anak bangsa dan tidak adanya tempat untuk menampung info – info mengenai buku – buku serta komik – komik yang sudah ada. Komiqus sendiri memiliki banyak fitur yang mendukung perkembangan serta penyebaran hasil karya buku dan komik buatan masyarakat Indonesia seperti halaman review, halaman penulis, serta tempat social media platform untuk para pecinta pembaca buku dan komik local. Website Komiqus ini bisa diakses melalui <http://komiqus.com/>.



Gambar 2.2 Halaman Utama Komiqus.com

## Go

Merupakan bahasa pemrograman yang dibuat oleh Google. Bahasa pemrograman ini berfokus pada proses software development. Bahasa pemrograman ini efisien, tersedia secara Cuma – Cuma dan memiliki keunggulan dalam pengerjaan project oleh banyak programmers secara bersamaan. Go memiliki banyak keunggulan dan fitur – fitur yang membantu pada saat pengembangan salah satunya adalah *concurrency*, *libraries*, dan *garbage collector*. Go sendiri dibuat oleh Google untuk mengatasi masalah dalam pembuatan aplikasi memanfaatkan multi-core system. Go sendiri masih popular dikalangan para pengembang karena

libraries yang bisa mempermudah dan mempersingkat pengembangan aplikasi (Rob Pike, 2014).

## K6

K6 adalah tools buatan *Grafana Labs* yang bisa digunakan dalam membantu membangun script untuk melakukan testing website aplikasi. K6 bisa digunakan untuk *performance test*, *regression testing* maupun *smoke test*. K6 juga mendukung integrasi dan ekstensi yang bisa digunakan diberbagai platform dan aplikasi seperti Postman, Visual Studio dan lain – lain. K6 sendiri memiliki halam *website* yang bisa dikunjungi dengan link [https://k6.io](https://k6.io/).

## Cloudlets

*Cloudlets* unit pengukuran terkecil yang ada pada cloud. Sebuah *Cloudlets* setara dengan 128 MiB RAM dan 400 MHz kekuatan CPU. Pengukuran ini membantu sebuah sistem cloud dalam mengalokasikan kapasitas secara tepat untuk setiap bagian pada lingkungan *Cloud* (Tetiana Fydorenchyk, 2017)*.*

## Penelitian Sebelumnya

Tabel berikut menampilkan daftar penelitian – penelitian sebelumnya yang diambil oleh Penulis sebagai acuan dalam penulisan skripsi ini. Penelitian – penelitian yang dipilih diambil berdasarkan metode yang digunakan yaitu *load testing* atau *performance testing*. Tahun penelitian yang diambil adalah dari tahun 2018 keatas. Bahasa yang dicari adalah bahasa indonesia namun ada satu penelitian sebelumnya yang menggunakan bahasa inggris.

Tabel 2.1 Penelitian Sebelumnya

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tahun** | **Judul Penelitian** | **Penulis** | **Metode** | **Hasil Penelitian** |
| 2022 | Pengujian Performa Pada Website Lomba Nasional Kreativitas Mahasiswa | Hilman Nuril Hadi, Addin Aditya, Febry Eka Purwianton o | Performance Testing, JMeter | Beberapa halaman website dapat diakses hingga 500  pengunjung dalam satuan waktu 1 detik secara bersamaan |
| 2021 | Analisis Kinerja Website Pemerintah  Kota Pekalongan | Mohammad Reza Maulana, Eko Budi Susanto, Sattriedi Wahyu Binabar | Performance Testing, Load Testing, JMeter | Website hanya bisa menangani 16.8  Permintaan setiap menitnya |
| 2021 | Pengujian Aplikasi Greenwallet Dengan Metode Load Testing Dan Apache Jmeter | Cahaya Putri Agustika, Wahyu SJ Saputra, Mohammad Idhom | Load Testing, JMeter | Dari semua pengujian yang dilakukan respon waktu yang didapat dari server ketika diakses |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  | oleh seorang pengguna adalah kurang dari 1 detik |
| 2020 | Performance Testing On The Shopee Website In The Pandemic Period Of Covid-19 | Aulia Zahra Musthafawi  , Aulia Mas’adah, Sukmadinin gtyas | Performance testing, JMeter | Performa website semakin buruk ketika semakin banyak pengguna yang menggunakan aplikasi. |
| 2018 | Pengujian Sistem Informasi Penjualan dan Persediaan Barang Dengan Web Application Load Stress and Performance Testing (WAPT) | Herlamban g Brawijaya, Slamet Widodo, Samudui | WAPT (Web  Application Load stress dan Performance Testing) | Avg 90% response time, sec, avg:  nilai terhadap 1 pengguna dalam 1x request pada 1 halaman |

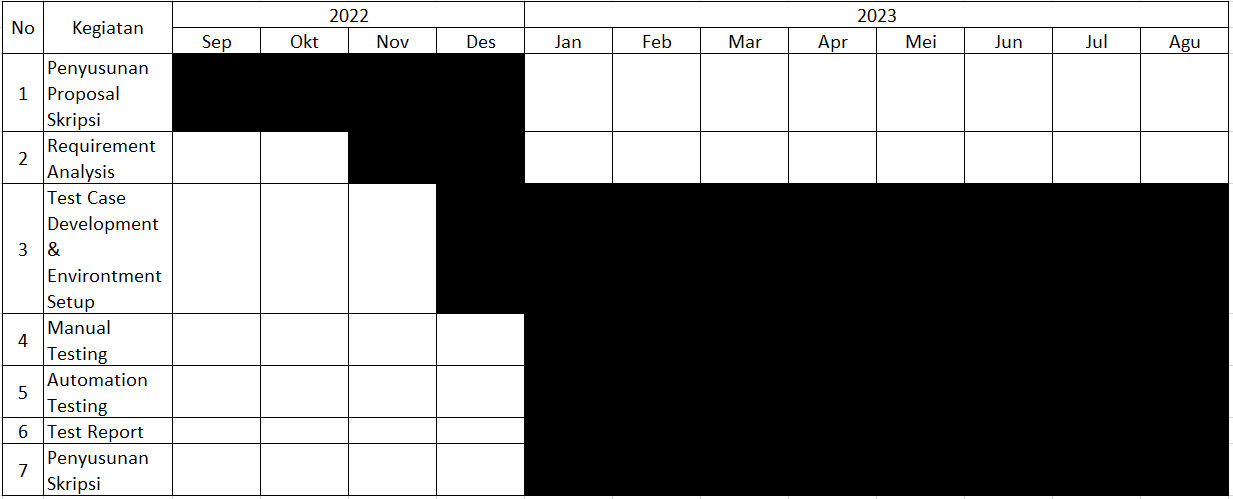
Tabel diatas merupakan daftar penelitan – penelitan sebelumnya yang menjalankan test yang sama, dengan mayoritas menggunakan JMeter sebagai alat penggujian. Penulis memilih untuk menggunakan K6 dibandingan dengan JMeter karena K6 bisa menjalankan tes jauh lebih ringan dalam hal banyaknya *virtual users* dibandingkan dengan JMeter (Giorgo Diprima, 2022).

# BAB III METODOLOGI PENELITIAN

## Waktu dan Tempat Penelitian

Tempat penelitian akan dilaksanakan di Gedung Telkom, Jl. Pajajaran No.39, RT.04/RW.06, Babakan, Kecamatan Bogor Tengah, Kota Bogor, Jawa Barat 16128. Dan waktu penelitian akan dilakukan sesuai dengan tabel berikut.

Tabel 3.1 Waktu Penelitian



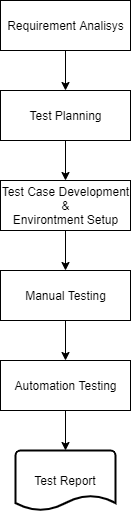
## Alat dan Bahan

Dalam Penelitian ini akan digunakan hardware yaitu satu buah laptop yang memiliki spesifikasi RAM 8GB dengan 2GB dari total 8GB ram dialokasi ke sistem, memori penyimpanan 1 TB HDD, prosesor AMD Athlon 300u. Penulis juga akan menggunakan software yaitu Visual Studio Code untuk membuat program untuk menjalankan otomatisasi proses K6. K6 sendiri membutuhkan Aplikasi K6 tersinstall pada laptop yang akan menjalankannya.

## Prosedur Kerja

* + 1. **Metode Penelitian**

Metode penelitian ini akan dilaksanakan dengan menggunakan metode STLC (Software Testing Life Cycle) yang sebelumnya sudah disesuaikan dengan penelitian ini, yaitu :



Gambar 3.1 Flowchart Metode Penelitian

## Requirement Analysis

Pada tahap ini akan dilakukan analisis terhadap website Komiqus yang nanti nya akan diuji. Untuk skripsi ini modul yang akan diuji adalah halaman utama, halaman katalog dan halaman *creator*. Ketiga halaman tersebut dipilih karena merupakan halaman yang paling mudah diakses oleh pengunjung dalam hal ini tidak perlu memerlukan kebutuhan tambahan seperti akun.

## Test Planning

Di dalam Tahap kedua ini akan dilakukan perencanaan test yang nanti akan dijalankan berdasarkan requirement analysis yang telah dilakukan seperti estimasi waktu dilakukannya pengujian dan sumber daya yang diperlukan saat pengujian website komiqus nanti.

Table 3.2 Test Planning

|  |  |  |
| --- | --- | --- |
| **Modul** | **Virtual Users** | **Jenis Tes** |
| Halaman Utama | 30 - 300 | Load Test |
| Halaman Katalog | 30 - 300 | Load Test |
| Halaman *Creator* | 30 - 300 | Load Test |

Sesuai dengan tabel diatas, tes yang akan direncanakan akan dilakukan dengan memberikan beban 30 hingga 300 *Virtual Users* pada setiap modul dan jenis tes yang akan dilaksanakan adalah *Load Test* dan *Performance Test* sesuai dengan batasan masalah.

## Test Case Development & Environment Setup

Pada tahap ini *test case* akan dibuat beserta test data yang nantinya akan digunakan pada saat melakukan test terhadap modul – modul atau fitur website komiqus. Kemudian dilanjutkan dengan Environment Setup dimana pada tahap ini akan mempersiapkan modul – modul website komiqus yang akan diuji coba seperti memastikan bahwa modul yang akan diuji berfungsi atau tidak.

Table 3.3 Test Case Development

|  |  |
| --- | --- |
| **Modul** | **Data** |
| Halaman Utama | https://komiqus.com/ |
| Halaman Katalog | <https://komiqus.com/browse/comics> |
| Halaman *Creator* | https://komiqus.com/andik.prayogo |

Tabel diatas menampilkan data yang dibutuhkan ketika ingin menjalankan tes, data ini merupakan URL untuk mengakses halaman modul yang ingin diuji yang nantinya akan digunakan oleh K6 pada saat tes dijalankan. Halaman utama website Komiqus terdapat beberapa fitur seperti daftar buku komik yang sedang *trending*, daftar testimoni – testimoni dari pengguna Komiqus, serta data kreator yang ada pada website Komiqus. Pada halaman katalog fitur – fitur yang ada adalah daftar buku yang baru terbit, daftar buku yang baru saja diperbaharui, daftar buku terpopuler dan daftar buku yang memili rating terbaik.

## Manual Testing

Kemudian manual testing akan dilakukan berdasarkan test case yang telah dibuat terhadap modul – modul yang sudah dipersiapkan, gunannya untuk memastikan pengujian berjalan dengan baik sebelum dilakukannya automation testing. Pada tahapan ini setiap modul dikunjungi secara manual untuk memastikan bahwa halaman yang akan diuji bisa diakses.

## Automation Testing

Dari manual testing yang sudah berhasil dijalankan, kemudian akan dilakukan *automation testing* untuk mempermudah dan juga mempercepat pengujian website komiqus. Automation testing ini sendiri akan menggunakan Framework golang K6 yang bisa membantu dalam pembuatan script automation testing. Pada tahap ini tes akan dijalankan pada setiap modul – modul yang sebelumnya sudah disiapkan, mulai dari halaman utama, halaman katalog dan halaman *creator* dan jenis tes yang dijalankan adalah *Load Test*.

## Test Report

Dari *Automation testing* yang sudah selesai akan menghasilkan test report yang nantinya akan diberikan kepada pihak Komiqus sebagai evaluasi untuk digunakan dalam pengembangan website Komiqus dimasa depan. Laporan tes yang akan dihasilkan akan dibuat oleh K6 dalam bentuk HTML dan juga akan ditampilkan lewat terminal dimana program dijalankan ketika proses tes selesai. Laporan tes ini akan menampilkan beberapa data seperti, waktu terlama yang dibutuhkan untuk memuat halaman, waktu terendah, waktu rata-rata dan yang lain – lain.

* + 1. **Metode *Load Testing***

Load testing dilakukan dengan mempersiapkan *script* dengan data testing berupa *link* halaman yang akan diuji, jumlah user yang diinginkan. Pada load testing di penilitian ini jumlah pengguna maksimal adalah 300 pengguna dan halaman yang akan diuji adalah halaman utama, katalog dan penulis. Seperti yang telah dijelaskan pada bab kedua, *Load Testing* merupakan salah satu tipe dari non-functional testing dan yang dilakukan untuk mengetahui kinerja sebuah aplikasi / sistem ketika diakses oleh pengguna dengan jumlah yang banyak dan juga bagaimana aplikasi mengatasi trafficnya (Siddhant Shrivastava, 2020).

# BAB IV PEMBAHASAN HASIL PENELITIAN

## Pendahuluan

Bab ini akan membahas tentang hasil penelitian yang telah dijalankan. Dimulai dengan objek yang diteliti dalam hal ini adalah *website* Komiqus, seperti infrastruktur dari Komiqus sendiri, halama – halaman yang akan diuji, kenapa memilih halaman tersebut. Lalu dilanjut dengan pembahasan proses bisnis didalam halaman yang akan diuji. Kemudian dilanjut dengan software dibuat oleh penguji, seperti bahasa pemrogramman yang digunakan, algoritma yang digunakan dan cara penggunaanya. Lalu dilanjut dengan hasil hasil testing yang dilakukan, mulai dari data latih dalam rupa excel dan data latih tersebut akan di evaluasi dan dicari hasil terbaiknya. Kemudian dilanjut dengan evaluasi hasil pengujian yang telah dilakukan di dunia nyata. Terakhir rangkuman mengenai keseluruhan dari bab 4.

## Requirement Analysis

Komiqus merupakan website social cataloguing yang berfokus pada peminat serta penulis buku, komik dan novel di Indonesia. Komiqus sendiri dibuat untuk mengatasi masalah yang yang muncul karena judul – judul buku dan komik buatan anak bangsa yang tidak dikenal oleh banyak orang, tidak adanya sosialisasi mengenai hasil karya anak bangsa dan tidak adanya tempat untuk menampung info – info mengenai buku – buku serta komik – komik yang sudah ada. Komiqus dibuat dengan menggunakan bahasa pemrograman *CodeIgniter*, *AngularJS*, *JQuery*, *Python*, *Bootstrap*, dengan *MySQL* sebagai databasenya. Untuk arsitektur server, Komiqus menggunakan Apache sebagai host dengan beberapa node server yang apabila ditotalkan menjadi processor maksimal 8GHz, RAM maksimal 2.5 GiB, dan cloudlets maksimal 20.

## Object Penelitian

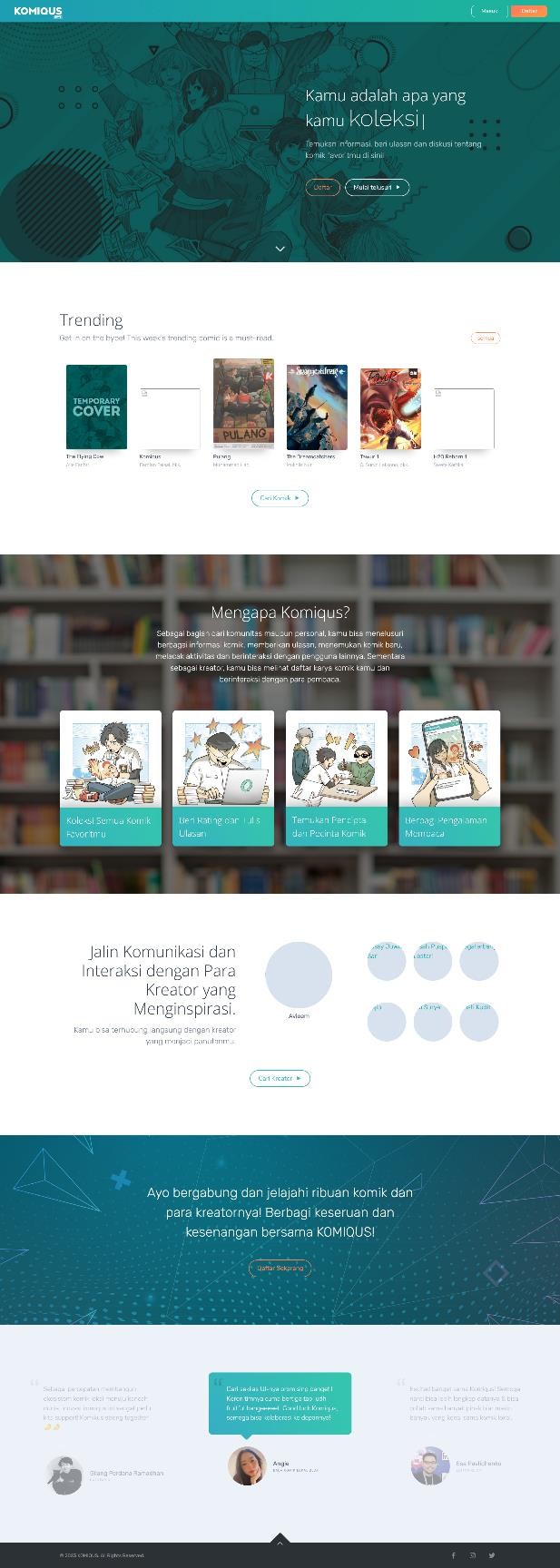
* + - 1. **Halaman Utama**



Gambar 4.1 Diagram Package Proses Bisnis Halaman Utama

Halaman utama dipilih menjadi salah satu halaman yang di uji coba karena halaman tersebut adalah halaman yang pertama kali akan dikunjungi oleh pengguna. Gambar diatas menampilkan diagram package untuk halaman utama *website* Komiqus.com. Halaman ini memiliki 3 buah proses bisnis yaitu Katalog Buku Trending, Daftar Testimonial dan Katalog Creator. Katalog Buku Trending yang berisikan data singkat mengenai buku

– buku yang sedang trending pada minggu ini. Daftar Testimonial berisikan Testimoni – testimoni dari pengunjung Komiqus sebelumnya. Dan terakhir Katalog Creator yang menampilkan beberapa creator di website Komiqus. Sebagai halaman utama, pengunjung akan secara otomatis akan diarahkan ke halaman ini ketika menggunakan link Komiqus.com. Pada halaman ini juga akan ditampilkan bagian singkat yang berisikan alasan – alasan kenapa Komiqus.com merupakan hal yang baik untuk pencinta komik atau kreator komik.



Gambar 4.2 Gambar Halaman Utama Komiqus

Gambar diatas merupakan halaman utama Komiqus dan merupakan halaman yang akan ditemui oleh pengguna jika mengakses website aplikasi Komiqus. Pada halaman ini terdapat beberapa informasi seperti informasi –

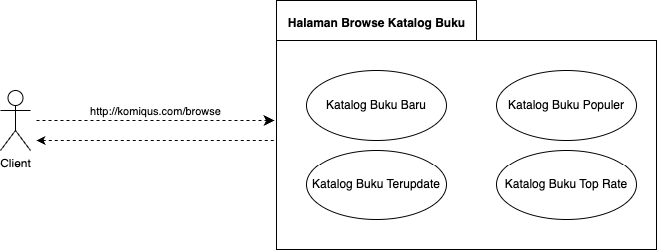
informasi singkat mengenai komiqus, daftar – daftar beberapa penulis yang ada pada Komiqus dan testimoni dari pengguna – pengguna Komiqus. Dari halaman ini Pengguna bisa memiliki pilihan untuk masuk kedalam halaman katalog buku komiqus, masuk kedalam halaman daftar penulis atau author dan juga bisa mendaftarkan diri sebagai pengguna komiqus atau masuk sebagai pengguna apabila sudah memiliki akun.

Tabel 4.1 Atribut Halaman Utama

|  |  |
| --- | --- |
| **Jenis** | **Jumlah** |
| Gambar | 33 |
| *Font* | 9 |
| *Script* | 73 |

Tabel diatas menampilkan banyaknya gambar, *font* serta *scripts* yang dimuat pada halaman utama *website* Komiqus.com. Halaman ini memuat 33 total gambar, 9 buah *font* dan 73 total *scripts*. Dari total script diatas 5 diantaranya adalah HTML, 16 CSS dan 52 Javascript. Pada halaman utama terdapat 9 total font yang digunakan yaitu OpenSans-Light, Rubik- Light, Nunito-Regular, Bootstrap-icon, Raleway-ExtraLight, Rubik-Italic, Rubik-Regular, Nunito-Light, dan Robot-Light.

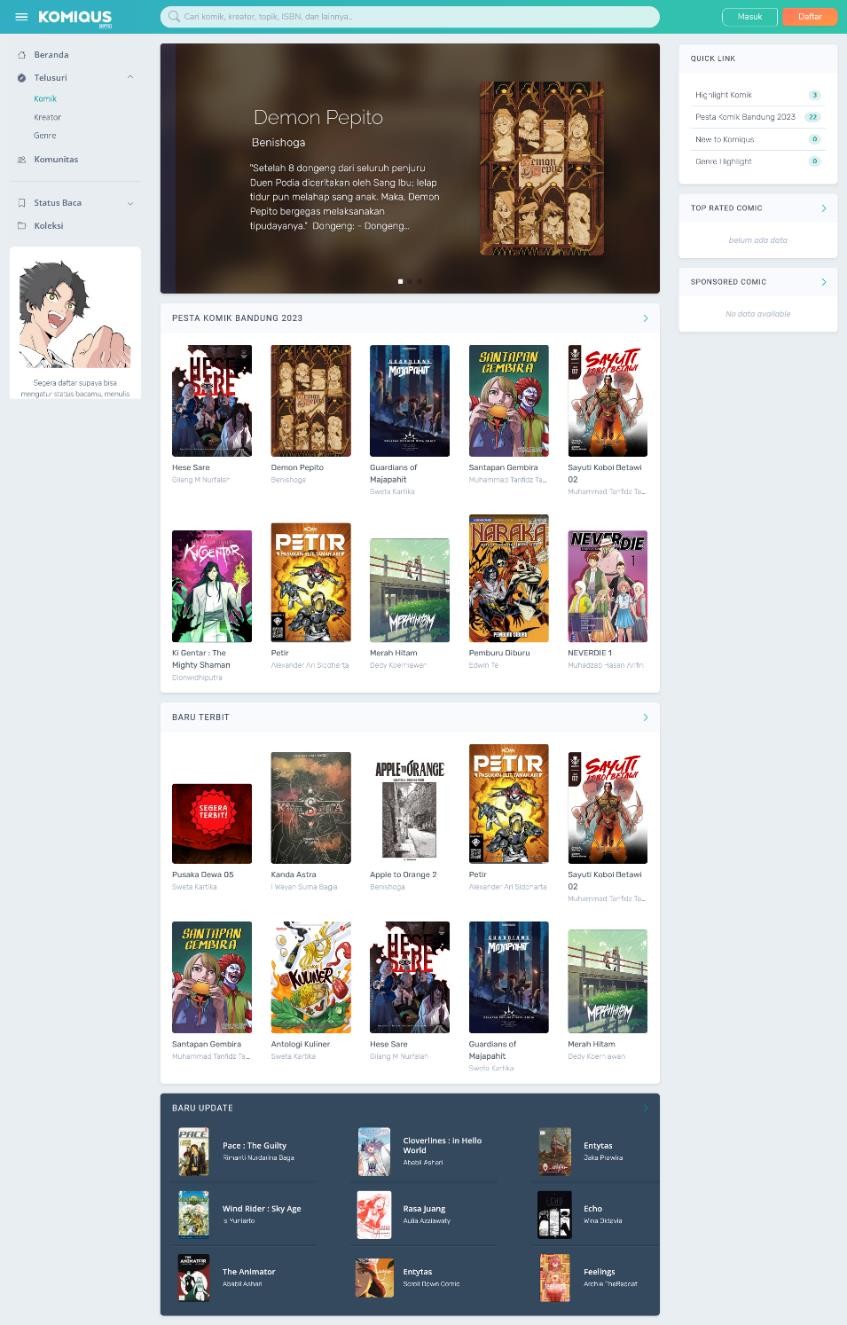
## Halaman Katalog



Gambar 4.3 Diagram Package Proses Bisnis Halaman Katalog

Halaman kedua yang dipilih adalah Halaman katalog. Halaman ini dipilih karena ketika pengguna dari halaman utama akan diarahkan masuk kehalaman ini. Dihalaman ini memiliki beberapa proses bisnis yaitu Katalog

Buku Baru, Katalog Buku Terupdate, Katalog Buku Terpopuler, dan Katalog Buku Terpopuler. Katalog buku baru menampilkan buku – buku yang baru terbit. Katalog buku populer menampilkan data – data buku yang sedang populer. Katalog top rate berisikan buku – buku yang memiliki rating tertinggi. Terakhir Katalog buku terupdate berisikan buku – buku yang akhir – akhir ini baru saja diupdate. Dihalaman ini sama seperti halaman pertama memiliki banyak gambar, font dan script – script yang bisa membebankan lamanya waktu memuat halaman.



Gambar 4.4 Halaman Katalog Komik Komiqus.com

Gambar diatas merupakan tampilan untuk halaman katalog Komiqus.com. Pada halaman ini terdapat berbagai jenis bagian, contohnya

seperti daftar buku yang baru terbit, daftar buku yang baru saja diperbaharui, dan daftar buku yang yang ada pada saat festival komik bandung 2023. Pada halaman ini pengunjung *website* Komiqus.com juga bisa mengakses beberapa menu atau pilihan yang memiliki berbagai fungsi contohnya, menu pencarian untuk mencari komik, topik, ISBN atau kreator. Pengunjung juga bisa berpindah ke halaman lain dengan menggunakan menu telusuri untuk mengunjungi berbagai halaman lain seperti halaman kreator, halaman *genre*, beranda dan juga komunitas. Pilihan untuk masuk ke menggunakan akun atau mendaftarkan akun baru juga tersedia melalui halaman ini.

Tabel 4.2 Atribut Halaman Katalog

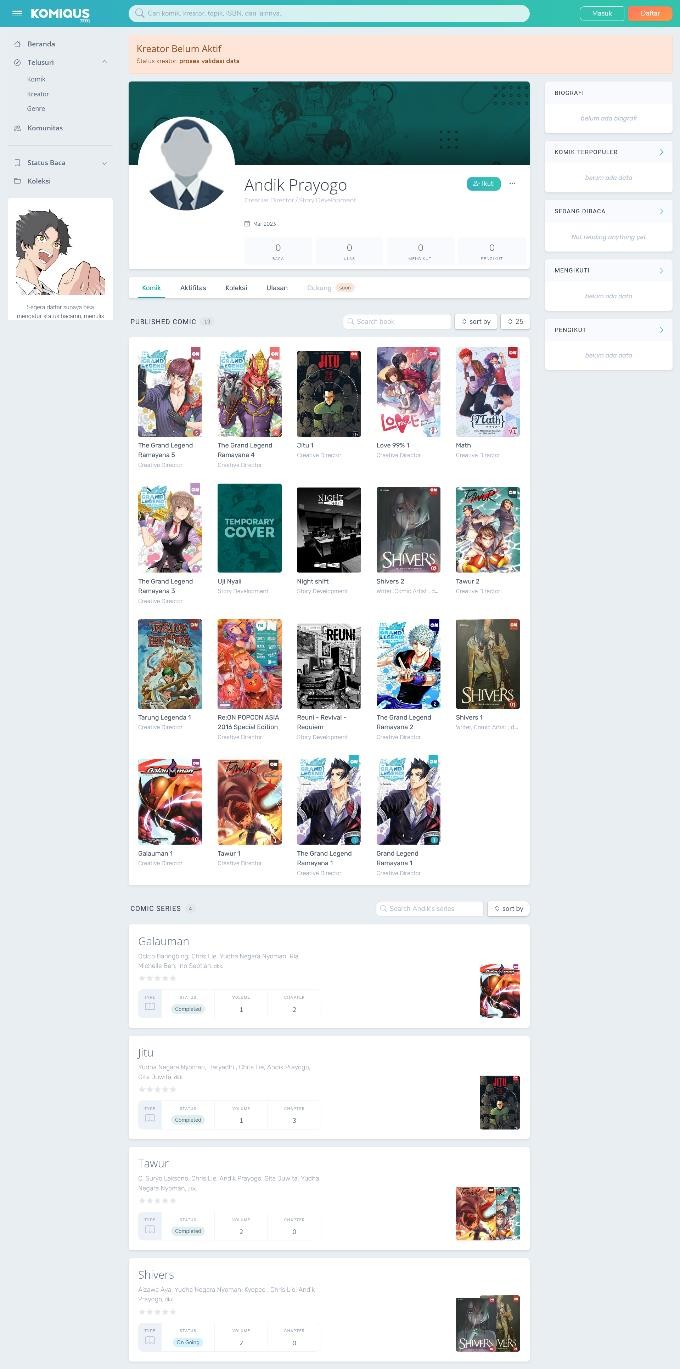
|  |  |
| --- | --- |
| **Jenis** | **Jumlah** |
| Gambar | 42 |
| *Font* | 12 |
| *Script* | 97 |

Tabel diatas menampilkan banyaknya gambar, *font* serta *scripts* yang dimuat pada halaman katalog *website* Komiqus.com. Halaman ini memuat 49 total gambar, 4 buah *font* dan 97 total *scripts*. Dari 97 total jumlah *script* 21 diantaranya adalah HTML, 16 CSS *script*, 60 *javascript*. Pada halaman katalog digunakan 12 total font yaitu Nunito-Regular, bootstrap-icons, OpenSans-SemiBold, Nunito-Light, Roboto-Regular, Rubik-Italic, Rubik-Light, OpenSans-Light, OpenSans-Regular, Raleway- ExtraLight, Rubik-Regular, dan OpenSans-Medium.

## Halaman Profil Kreator

Gambar 4.5 Proses Bisnis Halaman Penulis

Halaman ketiga dan terakhir yang dipilih sebagai halaman yang akan di uji coba adalah halaman Penulis atau Halaman Profil Kreator. Halaman ini dipilih karena pengunjung setelah sampai di halaman katalog buku, atau dari halaman utama bisa dialihkan ke halaman penulis. Di Halaman ini memiliki proses bisnis yang lebih banyak dibandingkan dengan halaman – halaman sebelumnya yaitu Katalog Buku Creator, Katalog Buku Terpopuler, Katalog Album Buku, Daftar Followers, Profil Biodata Creator dan Daftar Following. Katalog Buku Creator menampilkan daftar buku – buku atau komik yang telah dibuat oleh creator dari halaman yang dikunjungi. Katalog Buku terpopuler menampilkan buku – buku atau komik terpopuler yang dibuat oleh creator. Daftar followers menampilkan jumlah pengguna komiqus yang memiliki akun yang mengikuti creator tersebut, profil biodata creator berisikan profil singkat mengenai creator dan terakhir daftar following menampilkan angka dan creator – creator yang di ikuti oleh creator halaman yang di kunjungi. Pada halaman ini juga memiliki gambar, font dan script – script yang bisa memperlama lamanya waktu memuat halaman. Halaman ini juga memiliki beberapa fitur lain seperti apabila pengunjung sudah memiliki akun pada website Komiqus, pengunjung dapat mengikuti kreator yang diinginkan, info singkat mengenai kapan kreator tersebut bergabung dengan Komiqus, dan juga bisa melihat ulasan – ulasan komik yang telah dibuat oleh kreator dari halaman itu apabila ada.



Gambar 4.6 Halaman Kreator Komiqus.com

Gambar diatas merupakan gambar tampilan dari halaman kreator pada *website* Komiqus.com. Halaman ini akan menampilkan profil singkat mengenai kreator tersebut dan juga menampilkan daftar komik atau buku yang telah dibuat oleh kreator. Pengguna bisa mengakses berbagai menu seperti melihat koleksi dari kreator, melihat ulasan – ulasan yang dibuat oleh kreator jika sudah ada dan juga bisa mencari buku berdasarkan judul yang dibuat oleh kreator tersebut. Pada halaman ini juga akan tampil data singkat seperti jumlah pengikut dan kreator yang diikuti. Seperti pada halaman katalog, pengunjung website komiqus.com bisa mengakses halaman lain

dengan menggunakan menu telusuri untuk berpindah halaman seperti ke halaman *genre*, katalog dan halaman kreator. Sama seperti halaman katalog juga, pengunjung bisa mengakses menu untuk mendaftarkan akun baru atau masuk kedalam akun yang sudah ada.

Tabel 4.3 Atribut Halaman Profil Kreator

|  |  |
| --- | --- |
| **Jenis** | **Jumlah** |
| Gambar | 31 |
| *Font* | 13 |
| *Script* | 92 |

Tabel diatas menampilkan data – data mengenai atribut – atribut yang akan muncul pada halam kreator. Total gambar yang tampilkan dihalaman kreator yang diuji ini adalah 31 sedangkan total *font* adalah 13 dan total *script* yang dimuat adalah 92. Jumlah gambar dihalaman ini bervariasi sesuai dengan jumlah buku yang sudah dibuat oleh kreator halaman yang dikunjungi. Dari total 92 *scripts* dihalaman ini, 16 diantaranya adalah HTML, dan 16 lagi adalah CSS dan 60 sisanya adalah *javascript*.

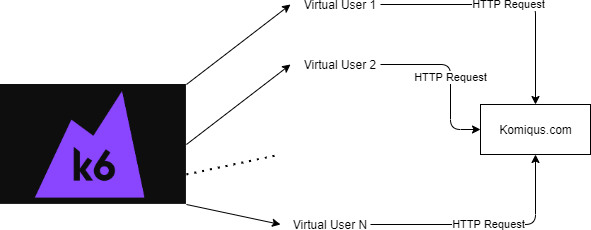
## Analisa Test Planning

* + 1. **Analisa Data Input**

Aplikasi yang dibuat menggunakan bahasa Go sebagai bahasa utama pembuatan program automated web performance testing. Aplikasi ini menggunakan K6 untuk melakukan test – test uji coba yang akan dilakukan, sehingga hanya perlu memasukan jumlah Virtual User yang akan digunakan dan juga link URL halaman web yang akan diuji. Virtual User sendiri berarti adalah pengguna simulasi yang menjalankan iterasi terpisah dan bersamaan dari skrip pengujian. Contohnya jika pada program dimasukkan 5 Virtual User atau VU maka, 5 pengguna simulasi akan memuat halaman URL yang diberikan. K6 sendiri menggunakan Golang dan Javascript sebagai pembuatan scriptnya. Untuk algoritma yang digunakan adalah Load Generation. Load Generation merupakan algoritma yang digunakan oleh K6 menghasilkan jumlah permintaan (requests) yang ditujukan ke website yang sedang diuji. Algoritma ini dapat berupa pembebanan berdasarkan jumlah pengguna virtual (VUs). Pada program yang telah dibuat VU di input oleh pengguna saat sebelum proses testing dimulai. Jumlah VU yang diterima oleh aplikasi digunakan oleh K6 sebagai acuan seberapa banyak beban yang akan dibuat dan dikirim ke target website yang akan di uji nanti.

Setelah Pengguna menjalankan program dan setelah URL dan jumlah VUs selesai dimasukkan, program akan secara otomatis menjalankan Load Testing. Ketika pengujian selesai, hasil testing akan dibuat kedalam dua buah file yaitu JSON dan HTML. File ini akan berisikan laporan hasil testing yang telah dilakukan. Berikut peragaan cara penggunaan aplikasi.

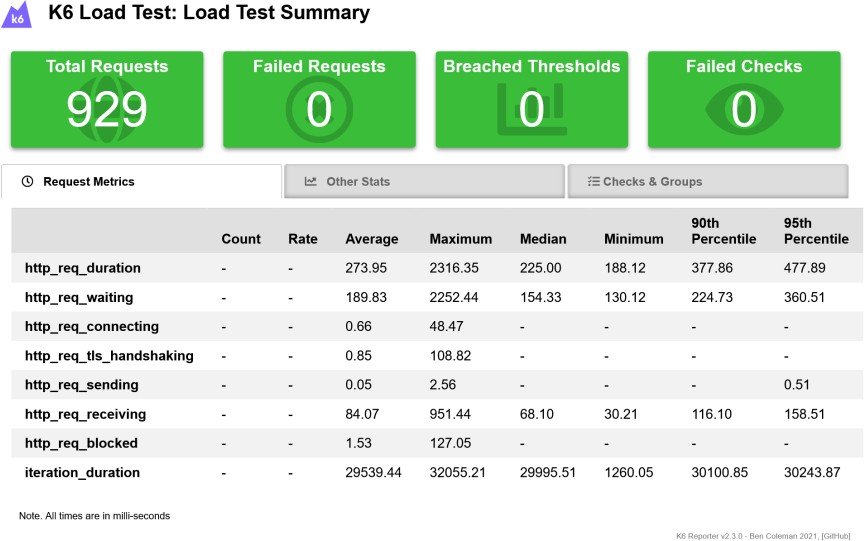
## Arsitektur K6



Gambar 4.7 Arsitektur K6

Gambar diatas merupakan cara kerja K6 dalam melakukan testing. K6 akan mensimulasikan *virtual users* berdasarkan jumlah yang diberikan ketika tes akan dimulai kemudian *virtual users* tersebut akan melakukan *HTTP Request* menuju URL yang telah diberikan. Kemudian K6 akan mencatat waktu yang dibutuhkan oleh tiap tiap *virtual users* dalam mengakses halaman yang di uji coba kemudian akan membuat laporan yang kemudian ditampilkan, laporan ini bisa dalam rupa HTML atau ditampilkan secara langsung pada terminal dimana K6 dijalankan.

Load Testing Tools merupakan komponen utama pada K6 yang digunakan dalam menjalankan *Load Test* dan memvisualiasikan hasil dalam bentuk laporan HTML. K6 juga bisa menggunakan aplikasi – aplikasi *open- source* untuk mengirimkan data hasil pengujian kepada aplikasi pihak ketiga yang berfungsi untuk sebagai aplikasi analisis. Ketika load testing menjadi tes yang sering diuji atau penting, tim *developers* biasanya membutuhkan pengembangan tes lebih lanjut yang memiliki kemampuan untuk menjalankan tes yang lebih besar dan mendistribusikan tes eksekusi di beberapa perangkat, meningkatkan keandalan dan kegunaan solusi.



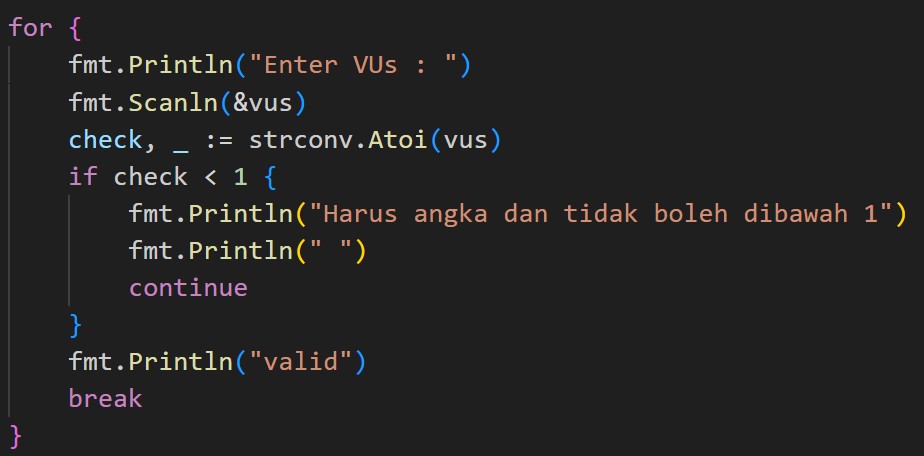
Gambar 4.8 Laporan Hasil Load Test

Gambar diatas merupakan contoh hasil laporan pengujian *load test*. Laporan akan menampilkan berbagai data mengenai hasil test yang baru saja dijalankan, Seperti banyaknya jumlah permintaan yang gagal, total permintaan yang dijalankan, lama waktu yang dibutuhkan untuk mengirim permintaan ke halaman *website* yang diuji. Pada bagian *Request Metrics* angka yang ditampilkan adalah waktu dalam satuan mili detik.

## Test Case Development & Environment Setup

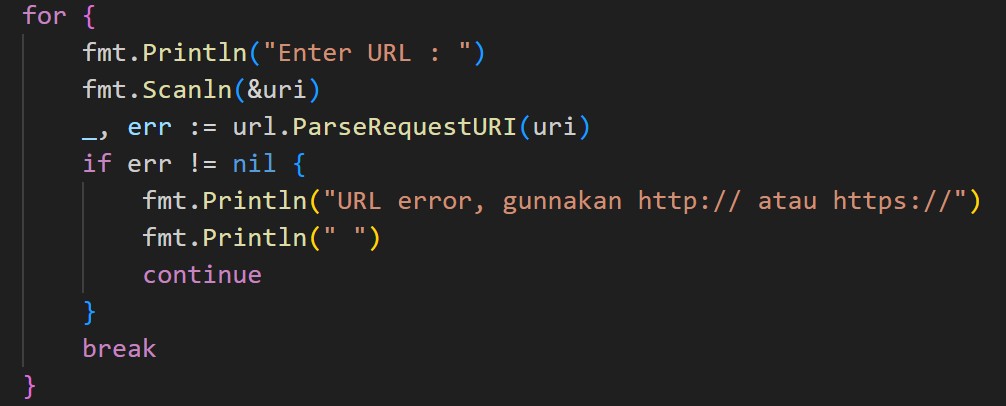
* + 1. **Input**

Seperti yang telah dijelaskan pada poin sebelumnya, pada saat pengujian aplikasi akan membutuhkan dua buah jenis input, VUs (Virtual User) dan juga URL. Keduanya akan digunakan pada saat melakukan testing, VUs sebagai jumlah beban yang akan diberikan dan URL sebagai tujuan website yang akan di uji coba.



Gambar 4.9 Screenshot syntax kode input VUs

Gambar diatas merupakan syntax kode yang digunakan oleh penulis untuk menerima masukan jumlah Virtual User yang diterima oleh pengguna aplikasi nanti, yang nantinya akan digunakan oleh K6 acuhan seberapa banyak beban yang akan dibuat dan dikirim ke target website yang akan di uji nanti.



Gambar 4.10 Screenshot syntax input URL

Gambar diatas merupakan syntax yang digunakan untuk menyimpan URL yang diinput. URL tersebut nantinya akan digunakan oleh K6 sebagai tujuan *HTTP Request* untuk setiap *Virtual Users.* Pada saat memasukkan URL, apabila URL tidak valid maka proses akan diulang kembali hingga URL yang dimasukkan sudah valid atau program dihentikan.

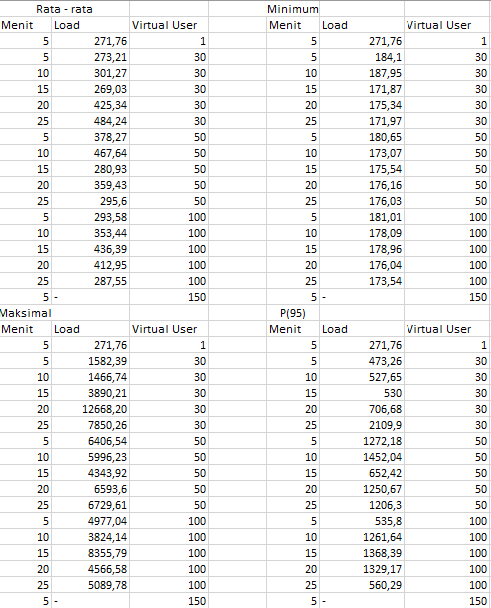
## A black screen with colorful text Description automatically generatedAlgoritma K6

Gambar 4.11 Syntax Kode Load Test

Gambar diatas merupakan syntax yang digunakan untuk memanggil K6 untuk menjalankan *Load Test*. URL dan *Virtual Users* yang sudah diinput tadi akan diberikan kepada K6 dan kemudian akan memulai proses uji coba. Laporan akan dibuat dalam bentuk HTML dan juga akan ditampilkan lewat terminal dimana program dijalankan.

## Pengaturan pengujian Automated Performance Testing

Selama penelitian, penulis telah melakukan beberapa testing dengan berbagai jenis skenario. Testing dilakukan di tiga buah halaman website Komiqus, yaitu halaman utama, halaman katalog dan halaman author atau kreator, sesauai dengan batasan masalah penelitian ini. Macam – macam skenario yang dijalankan juga bervariasi dengan tujuan mencari dan membuktikan hasil yang paling cocok. Skenario – skenario yang dijalankan adalah 1 VUs, 30 VUs, 50 VUs dan 100 VUs dengan waktu testing dimulai dari 5 menit, 10 menit, 15 menit, 20 menit dan 25 menit. Berikut adalah hasil pengujian dari setiap skenario di tiap halaman website yang diuji:

1. Halaman Utama

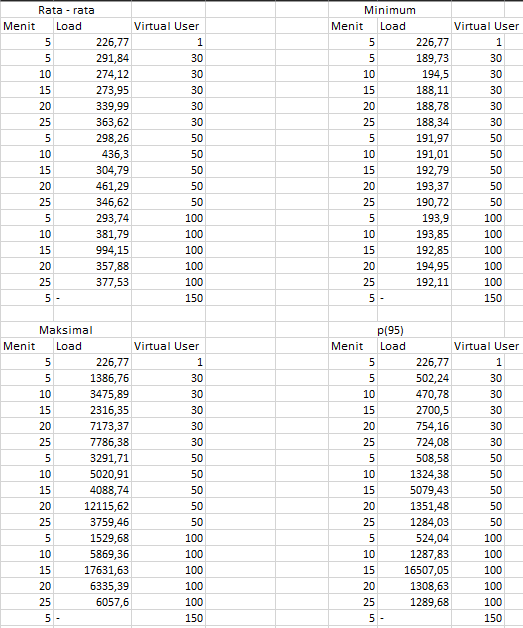
Gambar 4.12 Screenshot hasil pengujian halaman utama

Empat buah tabel diatas merupakan tabel hasil pengujian halaman utama, hasil yang diambil dari laporan JSON yang diambil adalah nilai hasil dari *http\_req\_duration* yang merupakan nilai metrik yang merepresentasikan total waktu yang dibutuhkan untuk setiap permintaan HTTP yang dibuat selama uji load testing. Metrik ini mencakup waktu yang diperlukan untuk membuat koneksi, mengirim permintaan, menerima respons, dan waktu pemrosesan lainya. http\_req\_duration memberikan wawasan tentang kinerja permintaan individual, memungkinkan mengidentifikasi permintaan spesifik yang mungkin lambat atau mengalami kemacetan. Ini membantu menilai daya tanggap situs web secara keseluruhan. Dari Tabel diatas terdapat 4 buah tabel: minimal, rata – rata, maksimal dan P(95), P(95) merupakan metrik nilai persentil 95% yang berarti dari 95% jumlah nilai data yang ada memiliki nilai dibawah P(95)

tersebut. Tabel minimal menampilkan nilai waktu terendah permintaan HTTP yang tercatat selama testing berjalan, tabel rata – rata menampilkan nilai waktu rata – rata, dan tabel maksimal menampilkan waktu tertinggi, dan terakhir P(95) atau persentil 95 menunjukan bahwa 95% dari permintaan HTTP yang dibuat memiliki nilai yang ditampilkan dan sisanya ada dibawahnya.

12668,20 mili detik pada pengujian load testing, dengan beban 30 VU dengan durasi testing 20 menit merupakan hasil tertinggi yang didapat. Waktu maksimal akan memiliki nilai yang tinggi sekali karena dari nilai http\_req\_duration yang diambil akan mencatat waktu walaupun pada saat proses permintaan HTTP terjadi *error*, *error* ini bisa membuat waktu durasi permintaan HTTP menjadi sangat panjang atau sangat pendek. Untuk load testing paling lama adalah 1452,04 mili detik dengan beban 50 VUs dan waktu durasi 10 menit.

1. Halaman Katalog



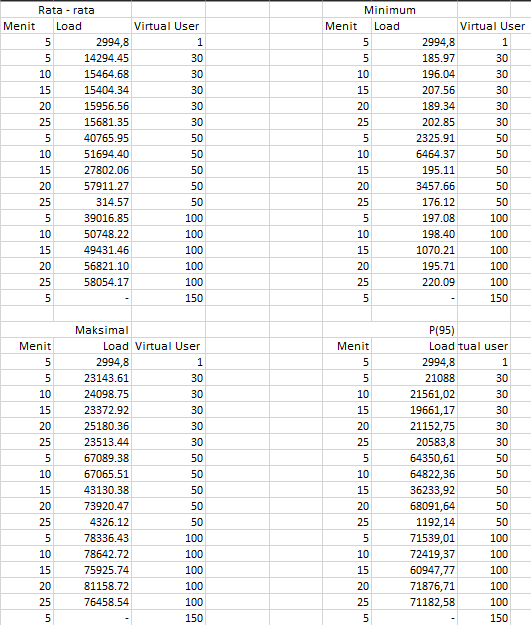
Gambar 4.13 Screenshot hasil pengujian halaman katalog

Gambar tabel – tabel diatas merupakan hasil pengujian dari halaman katalog. Sama seperti halaman utama skenario – skenario yang diuji

adalah 1 VUs, 30 VUs, 50 VUs dan 100 VUs dengan waktu testing

dimulai dari 5 menit, 10 menit, 15 menit, 20 menit dan 25 menit. Jika dilihat dari tabel P(95) waktu paling tinggi yang didapat pada saat load testing adalah 5904,62 mili detik dengan beban 100 VUs dengan durasi waktu 15 menit.

1. Halaman Profil Kreator



Gambar 4.14 Screenshot hasil pengujian halaman profil kreator

Gambar diatas merupakan hasil pengujian dari halaman kreator. Sama seperti halaman utama dan halaman katalog, skenario – skenario yang diuji adalah 1 VUs, 30 VUs, 50 VUs dan 100 VUs

dengan waktu testing dimulai dari 5 menit, 10 menit, 15 menit, 20 menit dan 25 menit. Jika dilihat dari tabel P(95) waktu paling tinggi yang didapat pada load testing waktu tertinggi adalah 72419,37 mili detik dengan beban 100 VUs dengan durasi waktu 10 menit, dan terakhir.

## Test Execution

* + 1. **Evaluasi hasil pengujian local environment**

Hasil pengujian local environment dijalankan pada laptop penulis yang memiliki spesifikasi RAM 8GB dengan 2GB dari 8GB tersebut dialokasi ke sistem, memori penyimpanan 1 TB HDD, dan prosesor AMD Athlon 300u. Kecepatan internet yang dimiliki oleh penulis adalah 2.86Mbps untuk download dan 5.65Mbps untuk upload. Penulis juga menggunakan sistem operasi windows 11 Home 1 bahasa dengan versi 10.0.22621 Build 22621. Pengujian dilakukan dengan menggunakan jumlah VUs dari 1 hingga 100 dengan waktu pengujian 5 menit, 10 menit, 15 menit,

20 menit dan 25 menit.

Berdasarkan data – data latih waktu durasi terbaik yang dipilih adalah 15 menit sebagai waktu acuan hasil yang akan dievaluasi dikarenakan durasi Pengujian yang cukup, durasi pengujian 15 menit memberikan periode yang cukup untuk mengamati performa aplikasi web dibawah tekanan yang diberikan. Hal ini memungkinkan waktu yang cukup untuk menghasilkan data yang berarti dan menangkap potensi masalah kinerja yang mungkin timbul selama penggunaan yang lama. 15 menit juga representasi yang seimbang, hasil tes selama 15 menit dapat dianggap sebagai representasi yang seimbang dari keseluruhan kinerja. Tidak terlalu singkat untuk melewatkan masalah potensial dan tidak terlalu lama sehingga sulit untuk menganalisis dan menginterpretasikan hasilnya. Durasi ini memberikan sampel interaksi pengguna yang cukup banyak, sehingga dapat mengevaluasi metrik kinerja secara efektif. Durasi pengujian 15 menit menghasilkan jumlah data yang dapat dikelola yang dapat dengan mudah dianalisis dan diinterpretasikan. Durasi pengujian yang lebih lama dapat menghasilkan volume data yang besar, sehingga lebih memakan waktu dan menantang untuk mengekstrak wawasan yang bermakna. Durasi 15 menit memberikan keseimbangan antara mengumpulkan data yang cukup dan memastikan analisis yang efisien.

Berikut adalah grafik nilai data tiap – tiap halaman yang diuji dengan durasi waktu 15 menit:

1. Halaman utama:

Halaman Utama

1600

1400

1200

1000

800

600

400

200

0

Respon waktu (mili detik)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 VUs | 30 VUs | 50 VUs | 100 Vus |
| Load Testing | 271,76 | 530 | 652,42 | 1368,39 |

Gambar 4.15 Hasil Pengujian Halaman utama

Pada grafik diatas bisa dilihat bahwa semakin tinggi VUs yang diberikan, semakin lama durasi waktu yang dibutuhkan hingga waktu halaman selesai dimuat, dengan waktu tertinggi terdapat pada *Load Testing* dengan durasi waktu 1368,74 mili detik dengan total beban 100 VUs.

1. Halaman Katalog

Halaman Katalog

7000

6000

5000

4000

3000

2000

1000

0

Respon waktu (mili detik)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 VUs | 30 VUs | 50 VUs | 100 VUs |
| Load Testing | 226,77 | 239,48 | 552,67 | 5904,62 |

Gambar 4.16 Hasil Pengujian Halaman Katalog

Pada grafik diatas bisa dilihat bahwa sama seperti test pada halaman utama, semakin tinggi VUs yang diberikan, semakin lama durasi waktu yang dibutuhkan hingga waktu halaman selesai

dimuat. Pada waktu load testing tertinggi pada 100 VUs dengan waktu hasil 5904,62 mili detik atau 5 detik

1. Halaman Profil Kreator

Halaman Profil Kreator

70000

60000

50000

40000

30000

20000

10000

0

Respon waktu (mili detik)

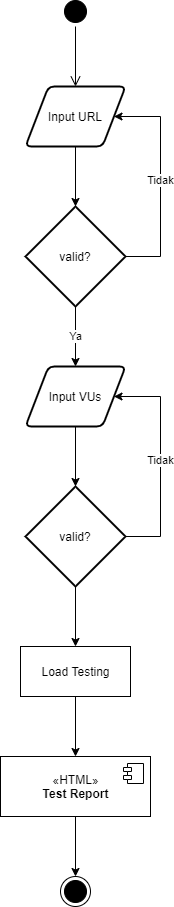
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1VUs | 30 VUs | 50 VUs | 100 VUs |
| Load Testing | 2994,8 | 20694,05 | 35743,31 | 66049,67 |

Gambar 4.17 Hasil Pengujian Halaman Author / Creator

Pada grafik diatas bisa dilihat bahwa sama seperti test pada halaman utama dan halaman katalog, semakin tinggi VUs yang diberikan, semakin lama durasi waktu yang dibutuhkan hingga waktu halaman selesai dimuat, waktu load testing tertinggi pada 100 VUs dengan waktu hasil 66049,67 mili detik atatu 66 detik.

## Evaluasi hasil pengujian real-world environment

Real-world pengujian environment dilakukan dengan menggunakan komputer server milik prodi TI IBI Kesatuan, dengan spesifikasi prosesor Intel i7-11700F 2.50GHz dengan RAM 32GB dan sistem operasi Windows 10 Pro 64-bit. Dengan perangkat spesifikasi diatas test dilakukan satu kali lagi pada 3 halaman website komiqus dengan beban 300 VUs dan waktu durasi 15 menit dan pengujian dengan beban 1 hingga 100 VUs menggunakan *Grafana Lab* yang disediakan oleh K6, pengujian 300 VUs tidak bisa dilakukan pada *Grafana Lab* karena limitasi fitur, program yang sudah selesai dijadikan file .exe dijalankan untuk mengetahui hasil test pada real-world environment.



Gambar 4.18 Diagram Activity Program

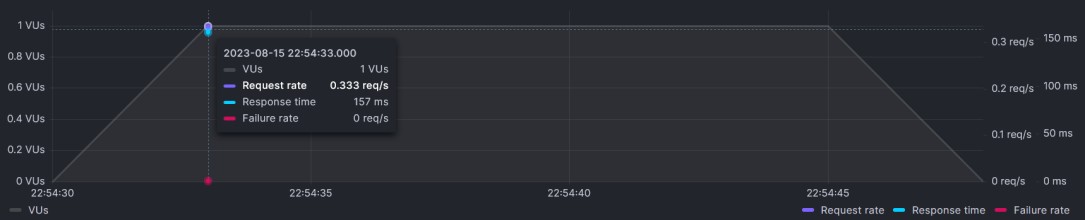
Dilihat dari diagram diatas, setelah program dibuka, tahap berikutnya adalah memasukan halaman *link URL* dari website yang akan diuji, *URL* ini harus dimulai dengan *https://* atau *http://*, apabila valid maka program akan melanjutkan ketahap berikutnya dan apabila tidak valid maka program akan mengulang kembali ketahap yang sama hingga *URL* yang dimasukkan valid atau program dihentikan. Tahap berikutnya adalah memasukkan jumlah *VUs (Virtual User)*, pada tahapan ini program hanya akan menerima angka yang melebihi dari sama dengan satu dan apabila dimasukkan nilai lain maka program akan mengulang dari tahap ini hingga nilai yang dimasukkan valid atau program dihentikan. Semakin besar jumlah *VUs* yang dimasukkan akan semakin besar juga *RAM* dan prosesor yang akan dipakai ketika proses test berjalan. Penulis telah mencoba untuk menjalankan pengujian dengan *VUs* sebanyak 300 yang berakhir dengan error karena sumber daya yang tidak mencukupi dalam hal ini adalah *RAM* dan kekuatan prosesor. Setelah *URL* dan *VU* berhasil dimasukkan program akan menjalankan *Load Test* dan membuat laporan test nya. Hasil laporan

test yang dibuat adalah dalam bentuk *HTML* yang bisa dibuka dengan menggunakan browser internet. Laporan test tersebut akan menampilkan data – data laporan. Setelah semua test selesai dijalankan maka program boleh dihentikan.

Dari program yang sudah jalankan *Load Test* akan dilakukan secara otomatis. Load Test adalah salah satu tipe dari *non-functional testing* dan juga bagian dari *performance testing* yang dilakukan untuk mengetahui kinerja sebuah aplikasi / sistem ketika diakses oleh pengguna dengan jumlah yang banyak dan juga bagaimana aplikasi mengatasi trafficnya (Siddhant Shrivastava, 2020).

## Test Closure

Berikut ini adalah bagan hasil dari pengujian pada real world environment menggunakan server TI IBI Kesatuan dan *Grafana Lab* K6:

1. Halaman Utama:

Gambar 4.19 Hasil Testing Real-world Halaman Utama Dengan 1 VU

Pada test ini Halaman Utama memerlukan waktu 157 mili detik untuk menyelesaikan permintaan ketika hanya memiliki beban 1 VU.



Gambar 4.20 Hasil Testing Real-world environment Halaman Utama Dengan 30

VUs

Pada gambar diatas menampilkan grafik hasil pengujian halaman utama dengan VUs sebanyak 30. Dengan waktu tertinggi adalah 17 detik, waktu terendah adalah 612 mili detik serta waktu rata – ratanya adalah 6 detik.



Gambar 4.21 Hasil Testing Real-world Environment Halaman Utama Dengan 50

VUs

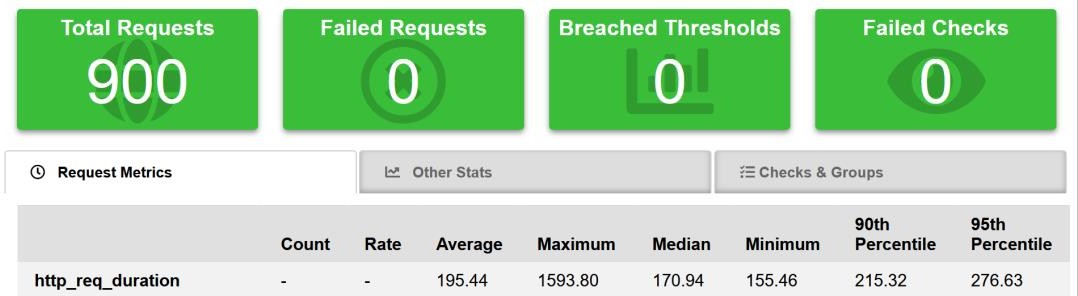
Pada test ini Halaman Utama memerlukan waktu paling tinggi selama 26 detik ketika berada dibawah tekanan 50 pengguna dan waktu terendah adalah 3 detik dan rata – rata adalah 11 detik.



Gambar 4.22 Hasil Testing Real-world Environment Halaman Utama dengan 100

VUs

Pada test ini Halaman Utama memerlukan waktu paling tinggi selama 55 detik ketika berada dibawah tekanan 100 pengguna dan waktu terendah adalah 5,24 detik dan rata – rata adalah 22 detik.



Gambar 4.23 Gambar Hasil Real-world Environment Halaman Utama Dengan

300 VUs.

Pada test ini Halaman Utama memerlukan waktu paling tinggi selama 276 mili detik ketika berada dibawah tekanan 300 pengguna dan waktu rata – ratanya adalah 195 mili detik dengan *bandwidth* yang digunakan adalah 80,75 MB.

1. Halaman Katalog :

Gambar 4.24 Hasil Testing Real-World Environment Halaman Katalog Dengan 1

VUs

Pada test ini Halaman katalog memerlukan waktu 393 mili detik untuk menyelesaikan permintaan ketika hanya memiliki beban 1 VU.



Gambar 4.25 Hasil Testing Real-world Environment Halaman Katalog Dengan

30 VUs

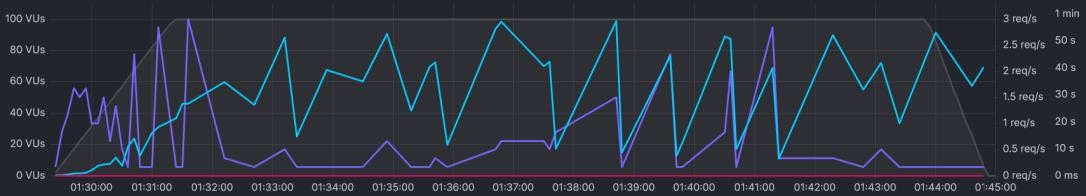
Pada gambar diatas menampilkan grafik hasil pengujian halaman katalog dengan VUs sebanyak 30. Dengan waktu tertinggi adalah 14 detik, waktu terendah adalah 503 mili detik serta waktu rata – ratanya adalah 3 detik.



Gambar 4.26 Hasil Testing Real-world Environment Halaman Katalog Dengan

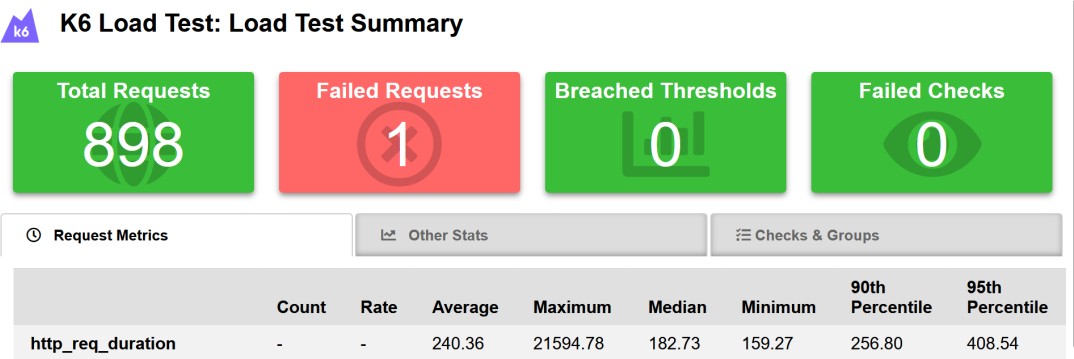
50 VUs

Pada gambar diatas menampilkan grafik hasil pengujian halaman katalog dengan VUs sebanyak 50. Dengan waktu tertinggi adalah 29 detik, waktu terendah adalah 279 mili detik serta waktu rata – ratanya adalah 9 detik.



Gambar 4.27 Hasil Testing Real-world Environment Halaman Katalog Dengan

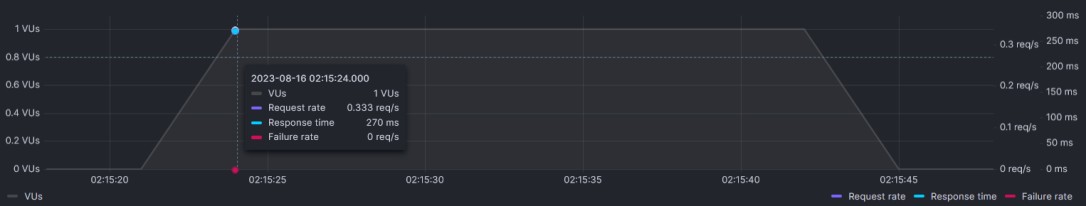
100 VUs

Pada gambar diatas menampilkan grafik hasil pengujian halaman katalog dengan VUs sebanyak 100. Dengan waktu tertinggi adalah 57 detik, waktu terendah adalah 6,36 mili detik serta waktu rata – ratanya adalah 21 detik.

Gambar 4.28 Hasil Testing Real-world Environment Halaman Katalog Dengan

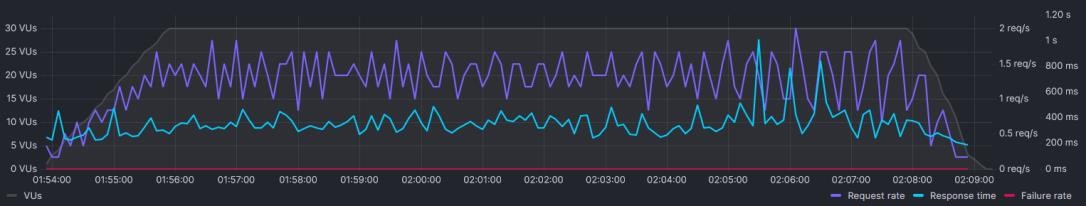
300 VUs

Pada test ini Halaman Katalog memerlukan waktu paling tinggi selama 21594 mili detik ketika berada dibawah tekanan 300 pengguna dan waktu rata – ratanya adalah 240,36 mili detik dengan *bandwidth* yang digunakan adalah 97,78 MB dimana pada pengujian ini terjadi satu kegagalan permintaan.

1. Halaman Profil Author

Gambar 4.29 Hasil Testing Real-world Environment Halaman Profil Kreator Dengan 1 VUs

Pada test ini halaman kreator memerlukan waktu 270 mili detik untuk menyelesaikan permintaan ketika hanya memiliki beban 1 VU.



Gambar 4.30 Hasil Testing Real-world Environtment Halaman Profil Kreator

Dengan 30 VUs

Pada gambar diatas menampilkan grafik hasil pengujian halaman profil kreator dengan VUs sebanyak 30. Dengan waktu tertinggi adalah 1 detik, waktu terendah adalah 239 mili detik serta waktu rata – ratanya adalah 262 mili detik.



Gambar 4.31 Hasil Testing Real-world Environtment Halaman Profil Kreator

Dengan 50 VUs

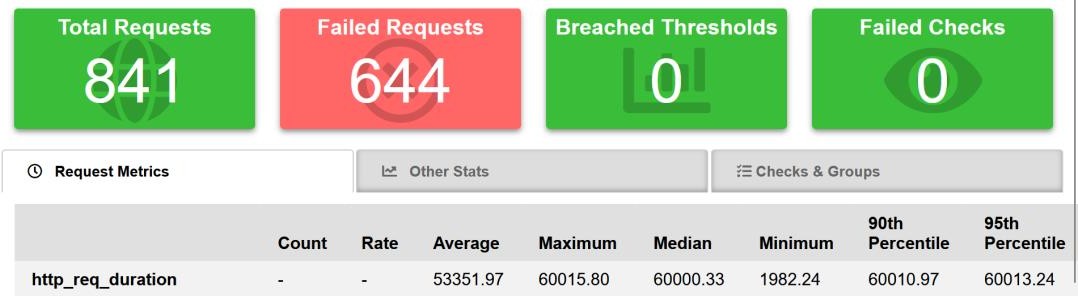
Pada gambar diatas menampilkan grafik hasil pengujian halaman profil kreator dengan VUs sebanyak 50. Dengan waktu tertinggi adalah 3 detik, waktu terendah adalah 632 mili detik serta waktu rata – ratanya adalah 639 mili detik.



Gambar 4.32 Hasil Real-world Environment Halaman Profil Kreator Dengan 100

VU

Pada gambar diatas menampilkan grafik hasil pengujian halaman profil kreator dengan VUs sebanyak 100. Dengan waktu tertinggi adalah 3 detik, waktu terendah adalah 1,94 detik serta waktu rata – ratanya adalah 2 detik.



Gambar 4.33 Hasil Testing Real-world Environment Halaman Profil Kreator

Dengan 300 VUs

Pada test ini Halaman Kreator memerlukan waktu paling tinggi selama 1 menit ketika berada dibawah tekanan 300 pengguna dan waktu rata

– ratanya adalah 55 detik dan waktu terendah adalah 53 detik dengan *bandwidth* yang digunakan adalah 39,88 MB dimana pada pengujian ini terjadi 644 kegagalan permintaan.

## Rangkuman

Dari pembahasan diatas bisa ditemukan bahwa Komiqus memiliki waktu yang bervariasi ketika berada diberbagai beban VU dan durasi yang berbeda-beda, banyak faktor yang mempengaruhi hasil tersebut yaitu spesifikasi perangkat yang menjalankan pengujian, kecepatan internet. Dengan waktu paling lama adalah 60000 mili detik ketika dibawah beban 300 VUs selama 15 menit pada saat *load testing*.

# BAB V KESIMPULAN DAN SARAN

## Kesimpulan

Setelah melakukan analisa masalah, perancangan pengujian, Test *Case development dan Environment Setup*, pengujian manual, pengujian otomatis dan laporan pengujian. metode STLC terhadap *Performance Test* untuk website Komiqus, pengujian ini berfokus pada kinerja aplikasi dan hasil pengujian dapat disimpulkan sebagai berikut:

* + 1. Berdasarkan pengujian *Performance Test* halaman utama dimana pengujian menggunakan user sebanyak 1 VU diketahui nilai estimasi matrix LCP (Largest Contentful Paint) dengan alat ukur GMetrix sebesar 2.3s, dimana nilai yang diperoleh ini bisa dikatakan cukup baik.
    2. Berdasarkan pengujian *Performance Test* halaman katalog dimana pengujian menggunakan user sebanyak 1 VU diketahui nilai estimasi matrix LCP sebesar 4.0, dimana nilai yang diperoleh ini bisa dikatakan buruk.
    3. Berdasarkan pengujian *Performance Test* halaman katalog dimana pengujian menggunakan user sebanyak 1 VU diketahui nilai estimasi matrix LCP sebesar 2.7, dimana nilai yang diperoleh ini bisa dikatakan cukup buruk.
    4. Berdasarkan pengujian *Load Test* halaman utama dengan pengujian user sebanyak 1 VUs mendapatkan hasil 157 mili detik, 30 VUs mendapatkan hasil 13 detik, 50 VUs dengan hasil 26 detik, 100 VUs dengan hasil 55 detik dan 300 VUs dengan hasil 408 mili detik.
    5. Berdasarkan pengujian *Load Test* halaman katalog dengan pengujian user sebanyak 1 VUs adalah 281 mili detik, 30 VUs mendapatkan hasil 13 detik dan pengujian user sebanyak 50 VUs mendapatkan hasil 26 detik, 100 VUs mendapatkan hasil 55 detik dan 300 VUs memiliki hasil sebesar kisaran 1 menit.
    6. Berdasarkan pengujian *Load Test* halaman profil kreator dengan pengujian user sebanyak 1 VUs adalah 270 mili detik, 30 VUs mendapatkan hasil 405 mili detik dan pengujian user sebanyak 50 VUs mendapatkan hasil 903 mili detik, 100 VUs mendapatkan hasil 3 detik dan 300 VUs memiliki hasil sebesar kisaran 1 menit.
    7. Untuk menjaga kestabilan website Komiqus ketika terjadinya naik turunya jumlah pengunjung, Komiqus bisa mempersiapkan Nodes tambahan untuk menjaga kestabilan website Komiqus.

Bagi penulis, penelitian ini menambah pengetahuan dalam membangun aplikasi menggunakan bahasa Go dimana bahasa pemrograman tersebut mampu dalam berinteraksi dengan aplikasi lain melalui terminal dan juga menambah wawasan dalam hal pengujian *non- functional* seperti cara melakukan pengujian *Load Testing* menggunakan *tools* K6. Penulis juga mendapat pengalaman menggunakan K6 sebagai aplikasi tools pengujian aplikasi yang mudah untuk berinteraksi dengan bahasa pemrograman Go. Aplikasi yang telah dibuat bisa digunakan dengan mengunduh seluruh *file* dan *folder* yang ada pada halaman *Github* penulis, kode aplikasi juga bisa diakses melalui halaman *Github* penulis melalui link berikut [https://github.com/dennydolok/Automatic-Performance-Testing.](https://github.com/dennydolok/Automatic-Performance-Testing)

## Saran

Berdasarkan kesimpulan penelitian diatas, penulis memiliki beberapa saran yang dapat disampaikan dari sisi operasional Komiqus dan dari sisi edukasi, yaitu sebagai berikut. Bagi pihak Komiqus sebaiknya mempersiapkan prosedur atau tindakan mitigasi sehingga di waktu Komiqus mengalami kedatangan pengunjung yang tinggi, performa website Komiqus terjaga dan tidak mengalami gangguan yang bisa mempengaruhi pengalaman pengunjung website Komiqus, banyak hal yang bisa dilakukan diantaranya adalah:

* + 1. Melakukan optimasi aset atau gambar yang digunakan pada halaman yang ditampilkan untuk mengurangi beban serta waktu yang dibutuhkan dalam memuat sebuah halaman.
    2. Mempersiapkan *Nodes Server* cadangan untuk mengurangi beban pada *Nodes Server* yang sudah ada.
    3. Menggunakan *Load Balancer* untuk mendistribusikan pekerjaan antar *Nodes Server* sehingga walaupun waktu memuat halaman berjalan dengan lambat, tidak terjadi kegagalan memuat halaman karena *Nodes Server* yang terlalu sibuk.
    4. Melakukan pemeriksaan dan optimasi pada halaman profil kreator karena banyaknya jumlah permintaan yang gagal dibandingkan dengan halaman yang lain.

Bagi yang ingin melanjutkan pengembangan aplikasi yang telah dibuat oleh penulis sebaiknya

1. Menyediakan UI yang ramah sehingga lebih mudah untuk digunakan.
2. Membuat aplikasi sebagai package installer sesuai dengan sistem operasi
3. Menambahkan model pengujian lain seperti Soak Testing, Endurance Testing, Stress Testing, Volume Testing dan Scalability Testing.
4. Mengoptimasi kode untuk menjalankan model pengujian sehingga dapat berjalan secara asynchronous.
5. Melakukan pencatatan hasil rekaman pengujian dari sisi disk I/O terhadap hardware yang di uji pada saat pengujian Load Testing sedang berjalan.
6. Mencoba *tools testing* lain selain K6 sebagai alternatif*.*

Bagi para *developers* pengembang K6 sebaiknya menambahkan fitur – fitur yang ada pada *Grafana Lab* kedalam K6 versi *desktop* karena terdapat fitur – fitur yang hanya bisa ditemukan pada versi *cloud* namun tidak tersedia pada versi *desktop*.

# DAFTAR PUSTAKA

Andriansyah, D. 2019. *Computer Based Information System Journal Performance Dan Stress Testing Dalam Mengoptimasi Website. Cbis Journal*, 07(01). http://ejournal.upbatam.ac.id/index.php/c[bisht](http://ejournal.upbatam.ac.id/i)tp:[//ejournal.upbatam.ac.id/i](http://ejournal.upbatam.ac.id/i) ndex.php/cbis.

Brawijaya, H., Widodo, S., Mandiri Jakarta, N., Damai No, J., & Jati Barat Pasar,

W. 2018. *P2M STMIK BINA INSANI Pengujian Sistem Informasi Penjualan dan Persediaan Barang Dengan Web Application Load Stress and Performance Testing (WAPT)*. INFORMATION SYSTEM FOR EDUCATORS AND PROFESSIONALS, 2(2), 125–134.

Dimaz, Rizky. 2019. Mengenal STLC — Software Testing Life Cycle. [Internet]. [Dikases pada 30 November 2022 11.00]. Tersedia pada : https://medium.com/dot-intern/mengenal-stlc-software-testing-life-cycle- d1bc5a938b72.

Freeman, A. 2022. *Pro Go The Complete Guide to Programming Reliable and Efficient Software Using Golang*. Apress. https://doi.org/10.1007/978-1- 4842-7355-5.

Hadi, H. N., Aditya, A., Purwiantono, F. E., Widyayuningtias, S., & Listio, P. 2022. *Pengujian Performa Pada Website Lomba Nasional Kreativitas Mahasiswa (Vol. 22, Issue 01)*.

Journal, I., & Shrivastava, S. 2020. *IRJET-Comprehensive Review of Load Testing Tools Comprehensive Review of Load Testing Tools. International Research Journal of Engineering and Technology*. [www.irjet.net.](http://www.irjet.net/)

Jyoti, B., & Sharma, A. K. 2016. *A perspective view of different techniques used in different phases of load testing. Indian Journal of Science and Technology*, 9(41). https://doi.org/10.17485/ijst/2016/v9i41/100227.

Musthafawi, A. Z., Mas’adah, A., Sukmadiningtyas, & Ramdani, F. 2020.

*Performance testing on the shopee website in the pandemic period of COVID-19. ACM International Conference Proceeding Series*, 195–199. https://doi.org/10.1145/3427423.3427457.

Nofia Delta, E. 2016. *Performance Test Dan Stress Website Menggunakan Open Source Tools*. In Jurnal Manajemen Informatika (Vol. 6). [https://unesa.ac.id/.](https://unesa.ac.id/)

Paul, Javin. 2020. Is Golang worth learning in 2022? Why should you learn Go Programming Language?. [Internet]. [Diakses pada tanggal 30 November 2022 10.30]. Tersedia pada : https://medium.com/javarevisited/what-is-go- or-golang-programming-language-why-learn-go-in-2020-1cbf0afc71db.

Pike, Rob. 2014. Go at Google: Language Design in the Service of Software Engineering. [Internet]. [Diakses pada tanggal 30 November 2022 10.00]. Tersedia pada : https://go.dev/talks/2012/splash.article.

Putri Agustika, C., & Saputra, W. S. 2021. *Pengujian Aplikasi Greenwallet Dengan Metode Load Testing Dan Apache Jmeter*. In Jurnal Informatika dan Sistem Informasi (JIFoSI) (Vol. 2, Issue 2).

Reza Maulana, M., Budi Susanto, E., Wahyu Binabar, S., & Widya Pratama Pekalongan, S. 2021. *Analisis Kinerja Website Pemerintah Kota Pekalongan (Vol. 20)*. https://pekalongankota.go.id/.

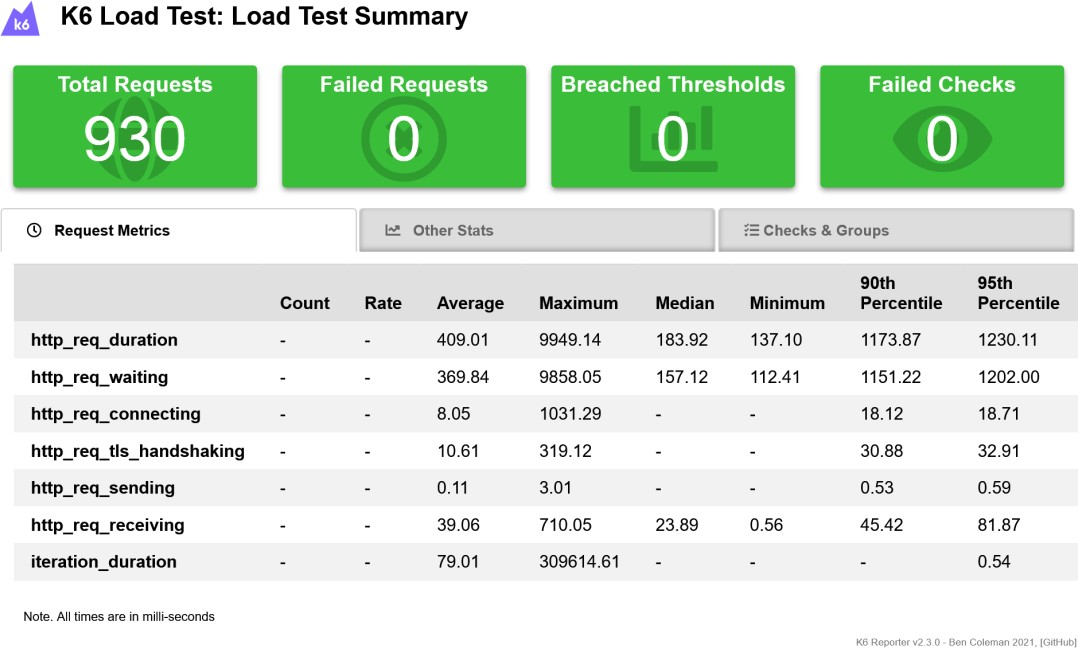
Sarojadevi, H. 2011. *Performance Testing: Methodologies and Tools*. 1(5). [www.iiste.org.](http://www.iiste.org/)

Heitzman, Adam. 2020. How Quickly Should a Page Load for Optimal Experience? [Internet]. Dikases pada 09 Juni 2023 11.00. Tersedia Pada : [https://www.searchenginejournal.com/how-quickly-should-page-](https://www.searchenginejournal.com/how-quickly-should-page-load/375799/#close) [load/375799/#close](https://www.searchenginejournal.com/how-quickly-should-page-load/375799/#close).

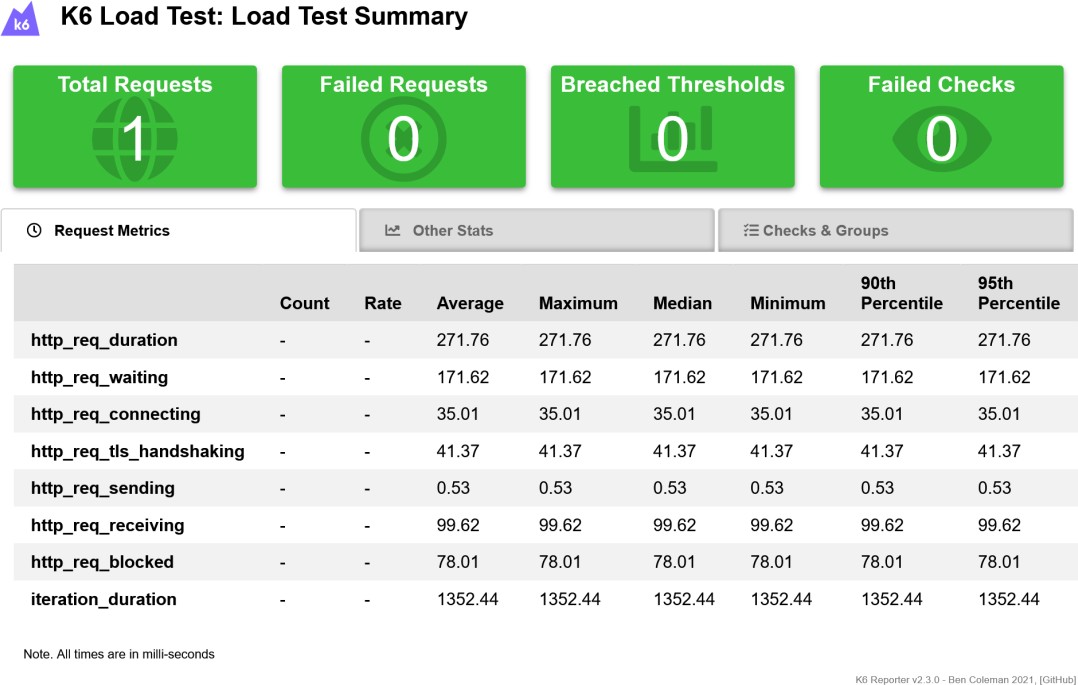
Diprima, G. (n.d.). *Comparison of tools and metrics for non-functional testing of web pages*. In *2022*. Turku University of Applied Sciences Information and Communications Technology.

Fydorenchyk, Tetiana. 2017. Jelastic Complex Organism in Simple Words [Internet]. Dikases pada 21 Agustus 2023 16.00. Tersedia Pada : [https://www.virtuozzo.com/company/blog/jelastic-complex-organism-in-simple-](https://www.virtuozzo.com/company/blog/jelastic-complex-organism-in-simple-words/) [words/.](https://www.virtuozzo.com/company/blog/jelastic-complex-organism-in-simple-words/)

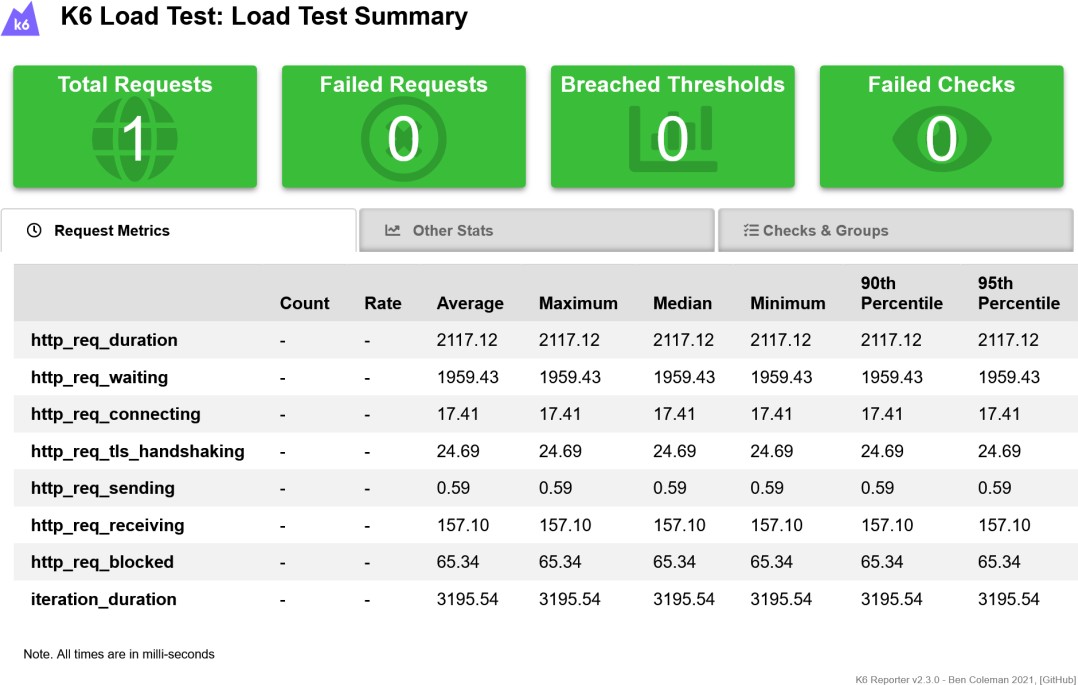
## LAMPIRAN



Gambar Hasil Load testing halaman Utama



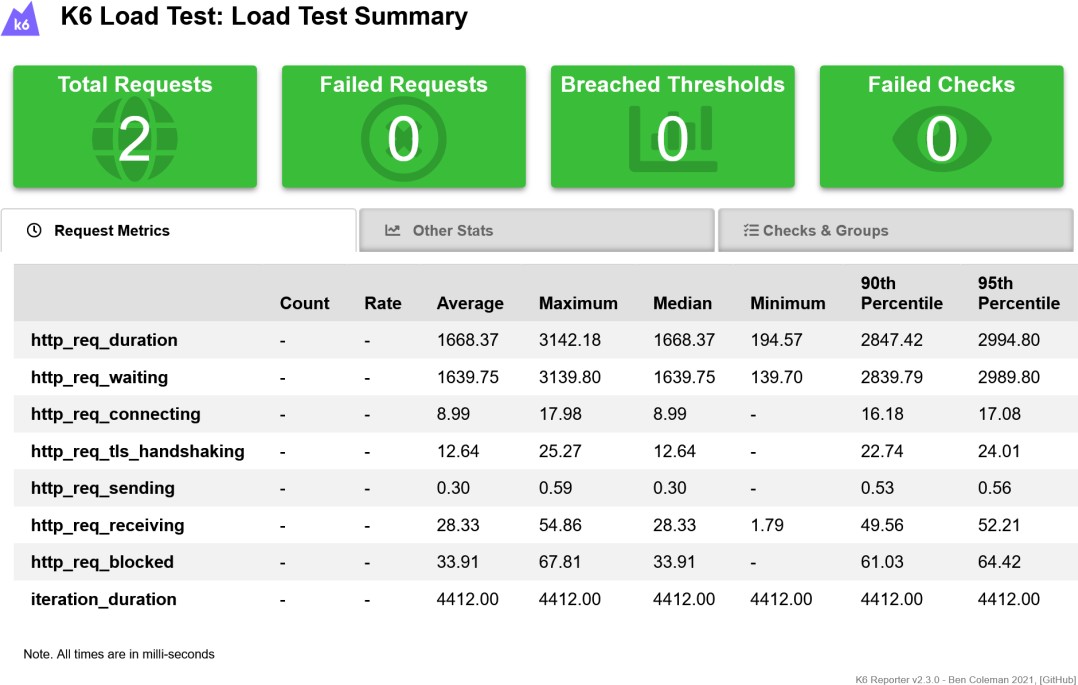
Gambar Hasil load testing halaman utama



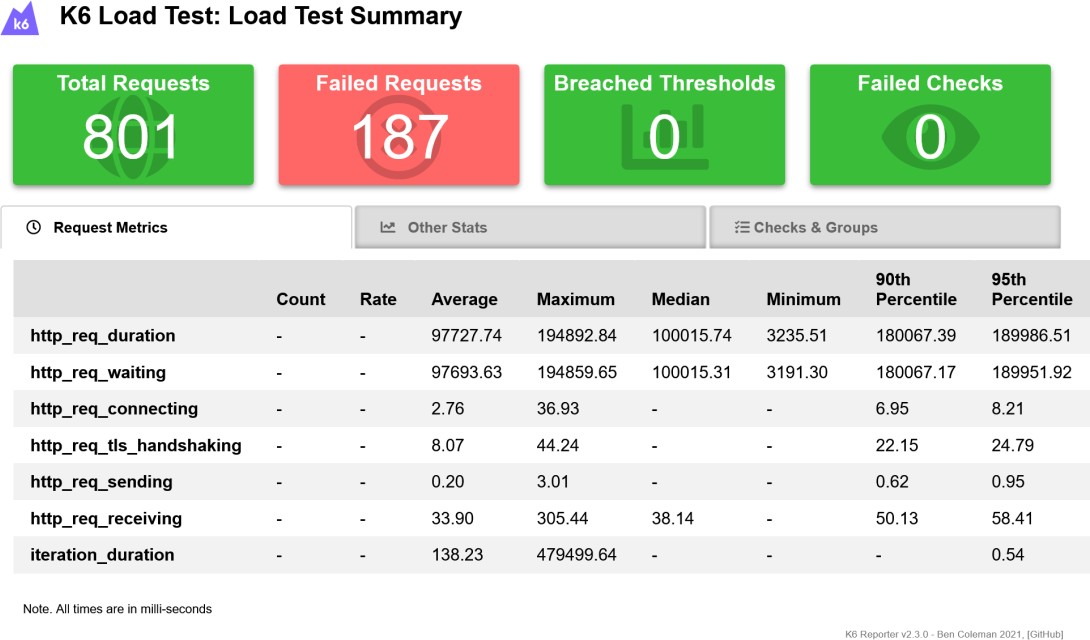
Gambar hasil load testing halaman katalog



Gambar hasil load testing halaman katalog



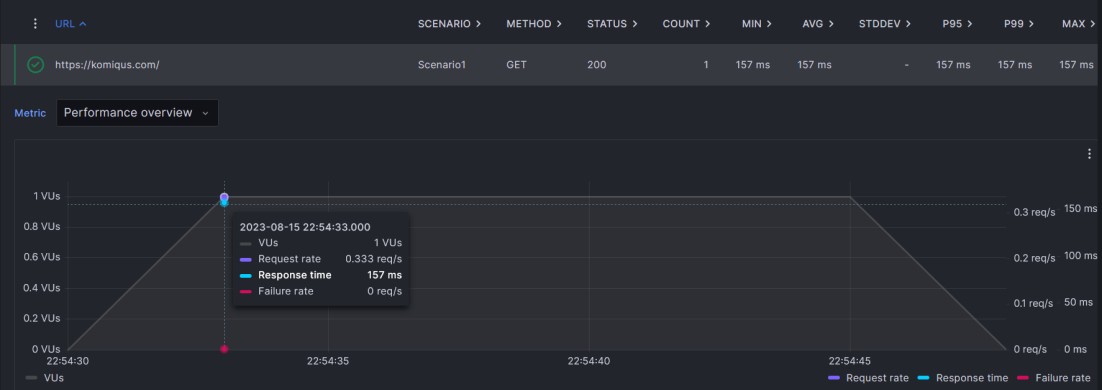
Gambar hasil load testing halaman author



Gambar hasil load testing halaman author



Gambar screenshot kode program



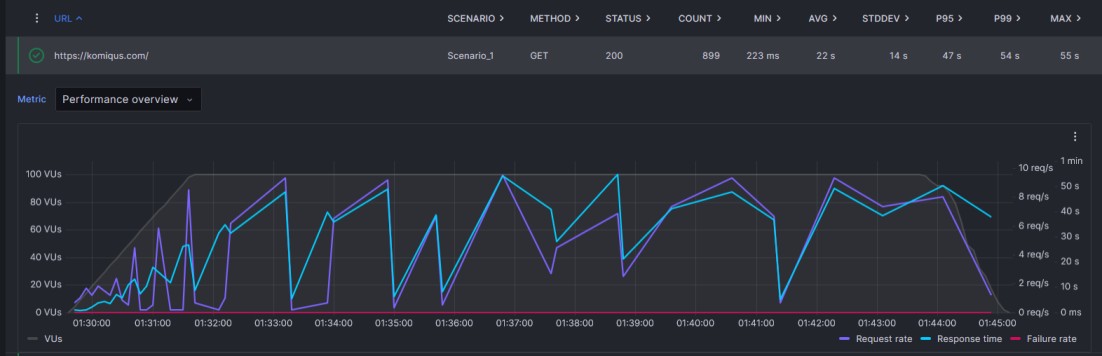
Hasil Pengujian Halaman Utama 1 VUs



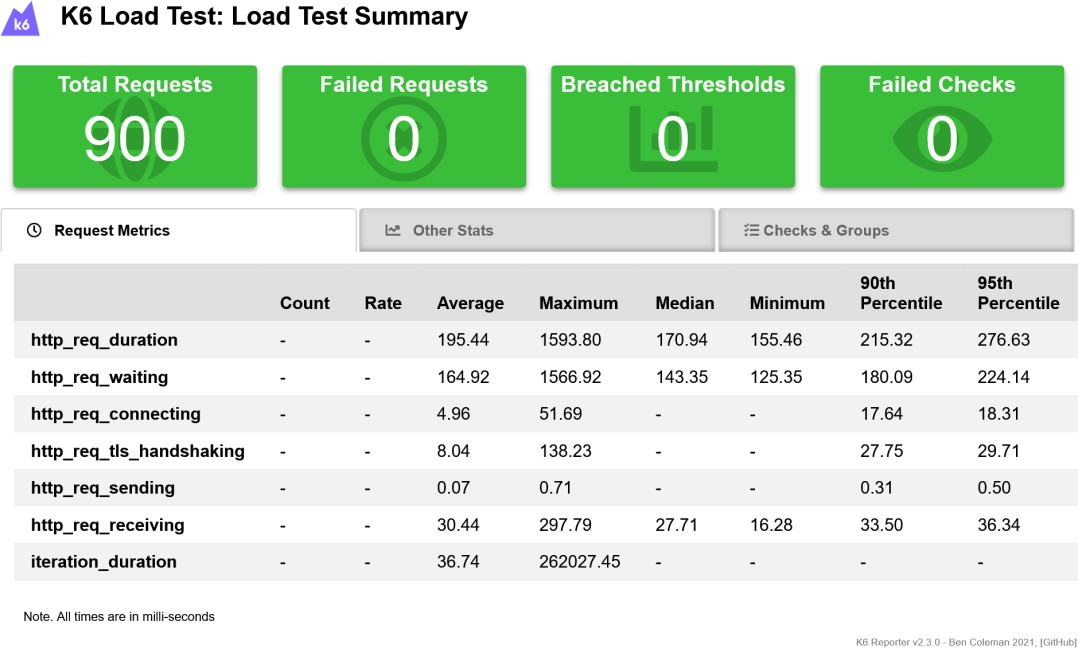
Hasil Pengujian Halaman Utama 30 VUs



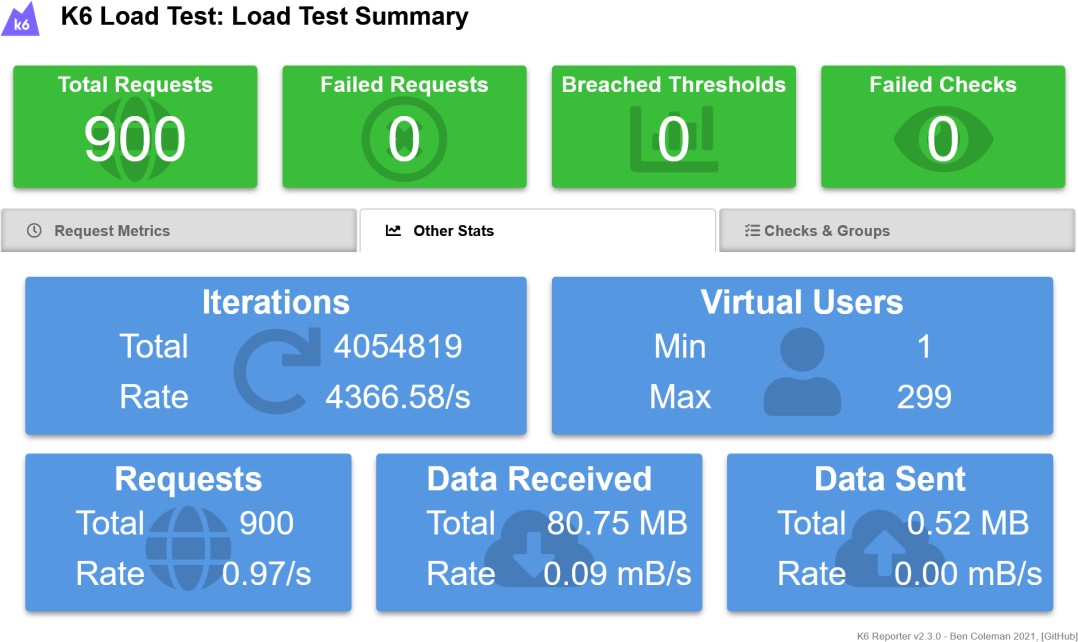
Hasil Pengujian Halaman Utama 50 VUs



Hasil Pengujian Halaman Utama 100 VUs



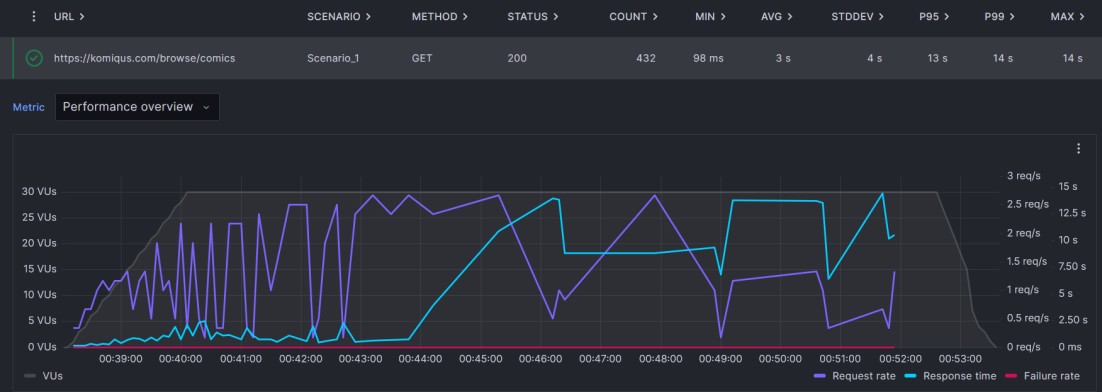
Hasil Pengujian Halaman Utama 300 VUs



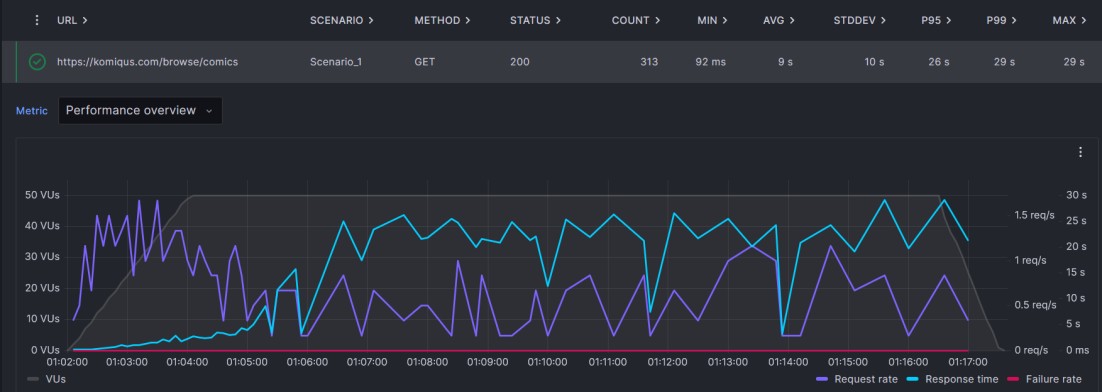
Hasil Pengujian Halaman Utama 300 VUs



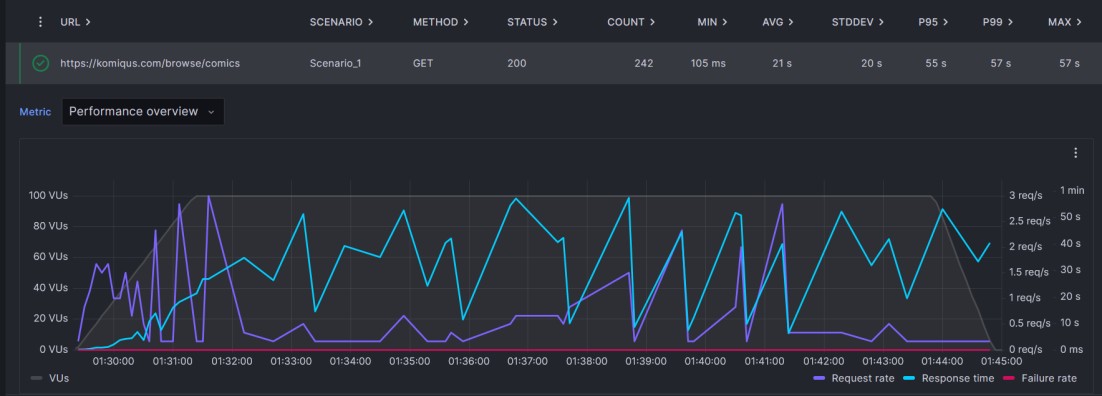
Hasil Pengujian Halaman Katalog 1 VUs



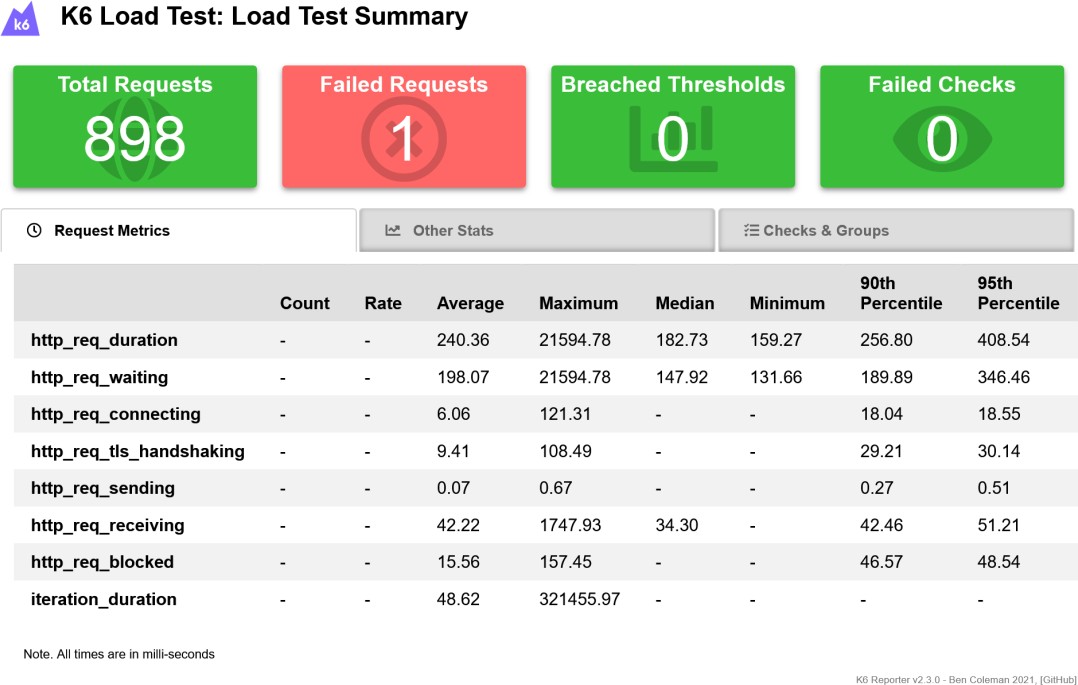
Hasil Pengujian Halaman Katalog 30 VUs



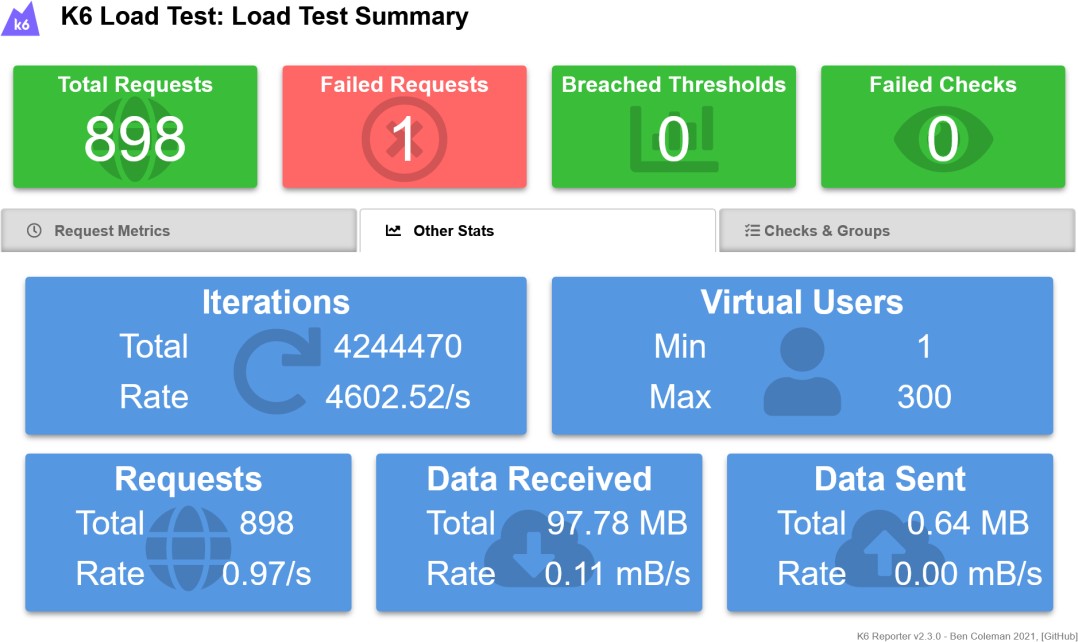
Hasil Pengujian Halaman Katalog 50 VUs



Hasil Pengujian Halaman Katalog 100 VUs



Hasil Pengujian Halaman Katalog 300 VUs



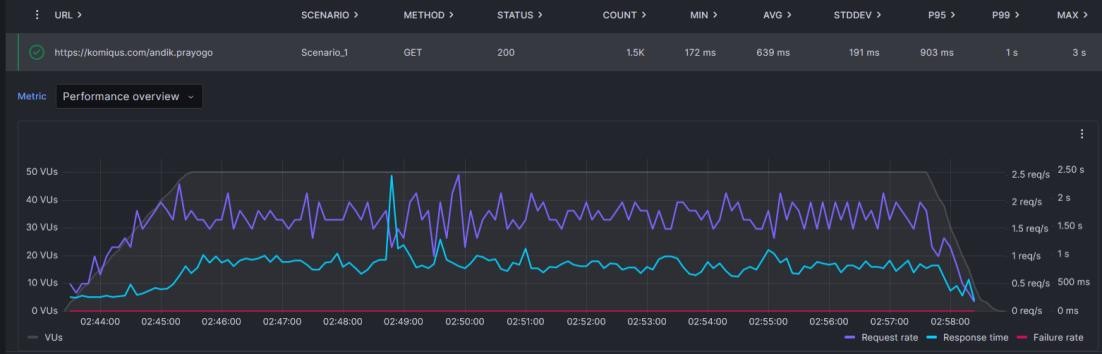
Hasil Pengujian Halaman Katalog 300 VUs



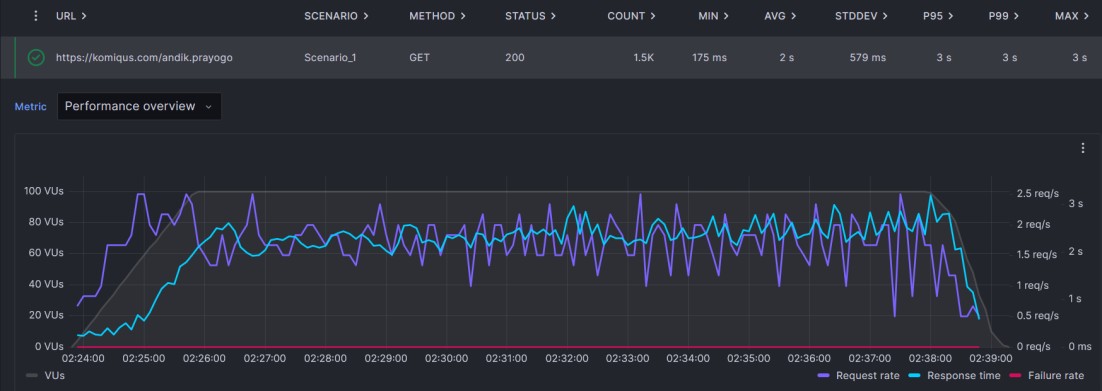
Hasil Pengujian Halaman Profil Kreator 1 VUs



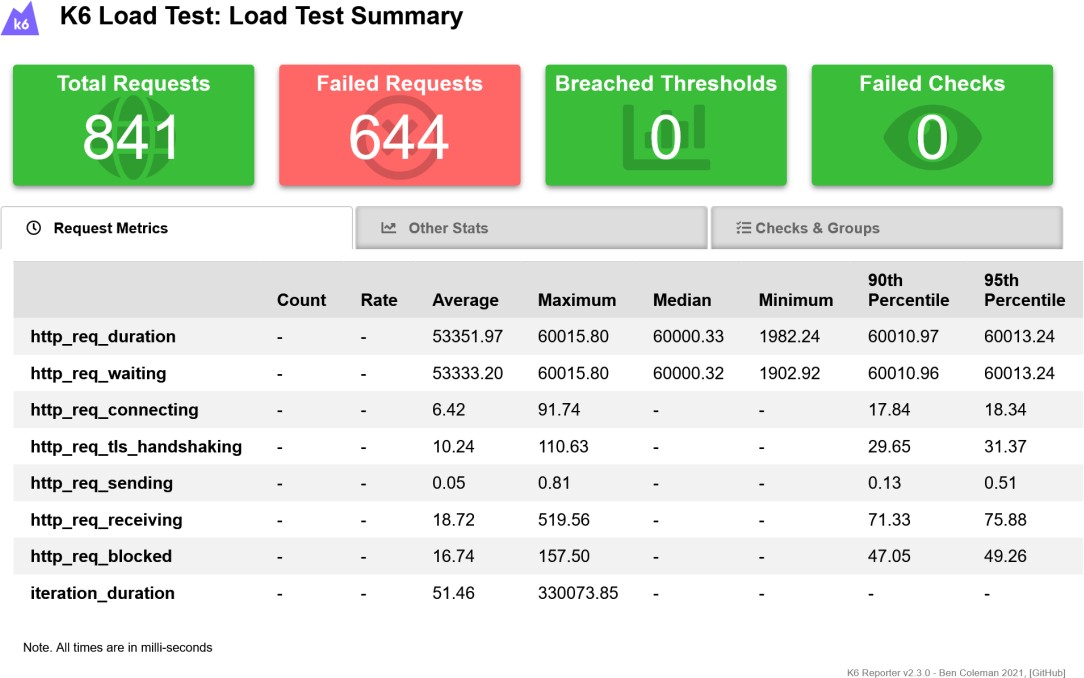
Hasil Pengujian Halaman Profil Kreator 30 VUs



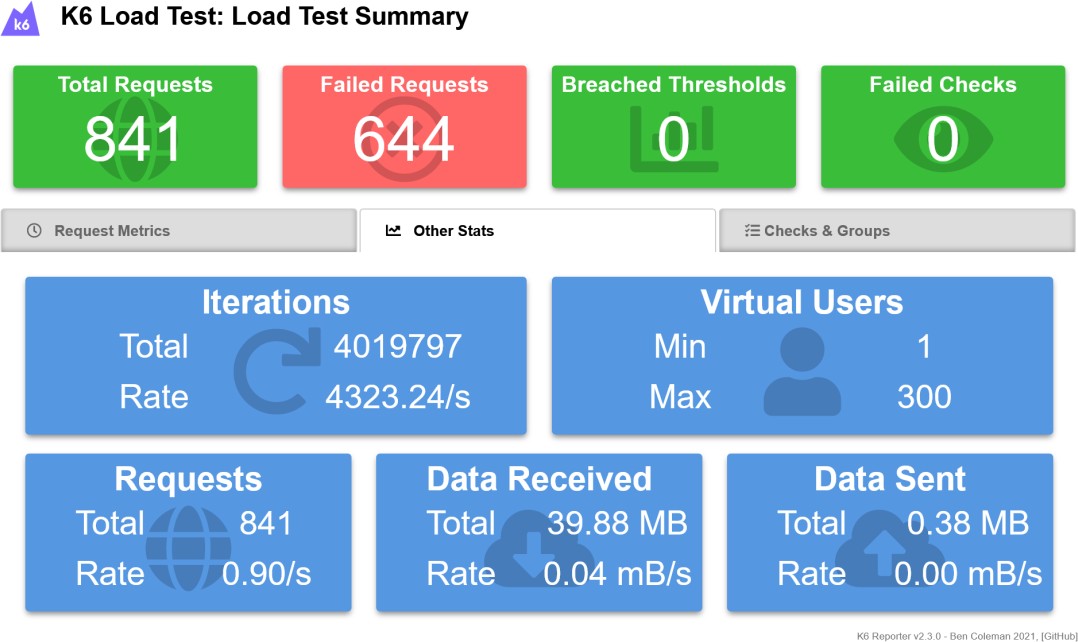
Hasil Pengujian Halaman Profil Kreator 50 VUs



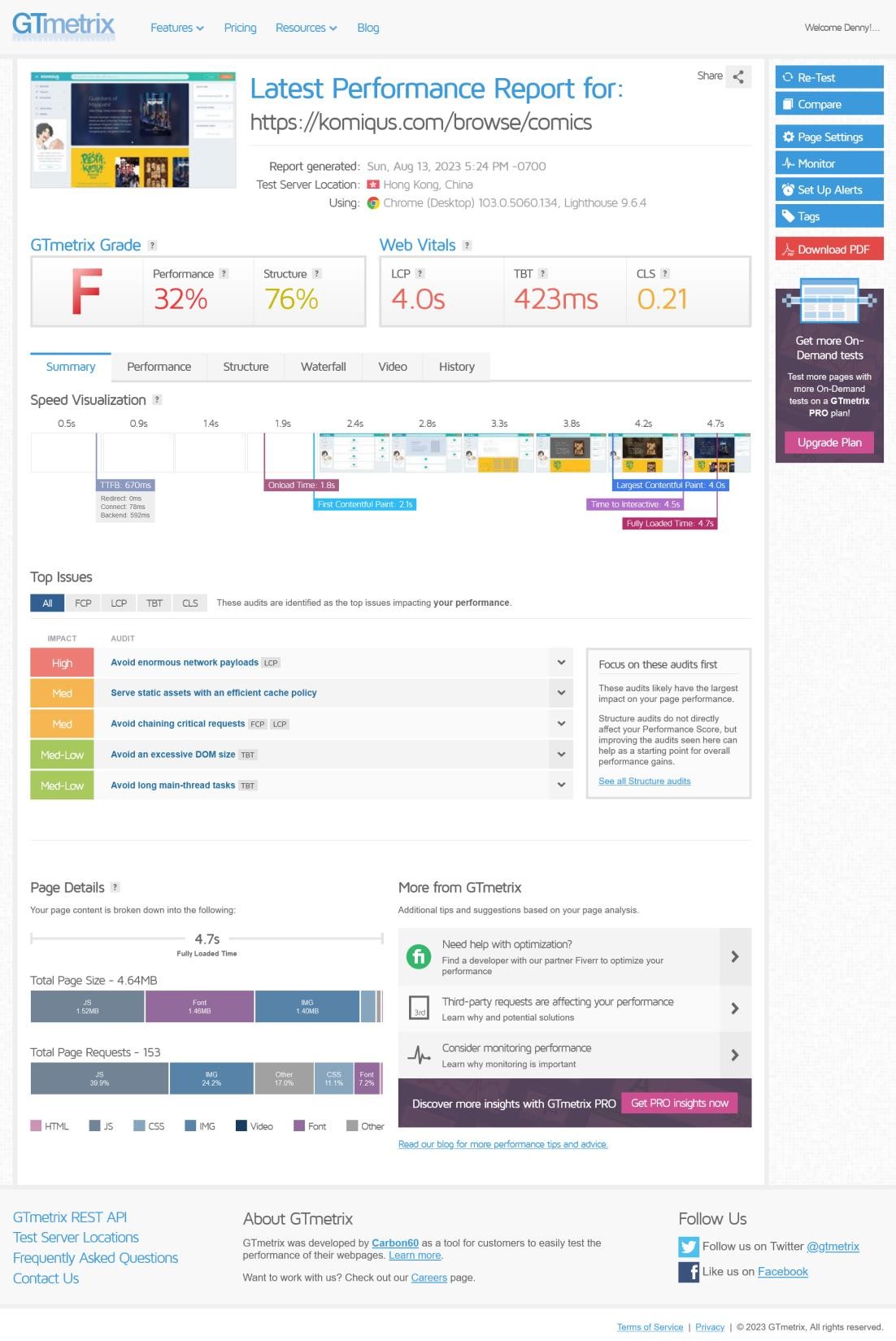
Hasil Pengujian Halaman Profil Kreator 100 VUs



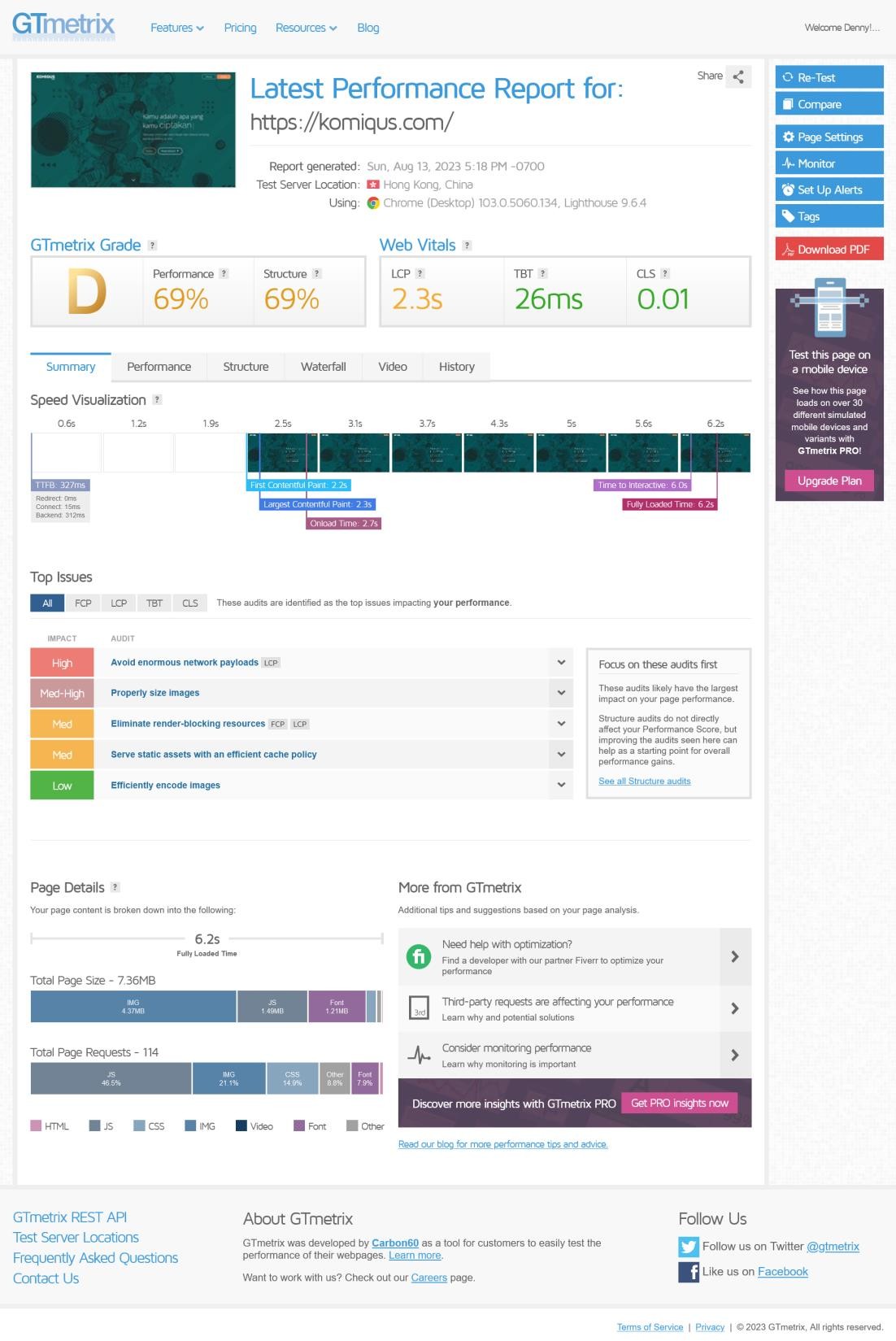
Hasil Pengujian Halaman Profil Kreator 300 VUs



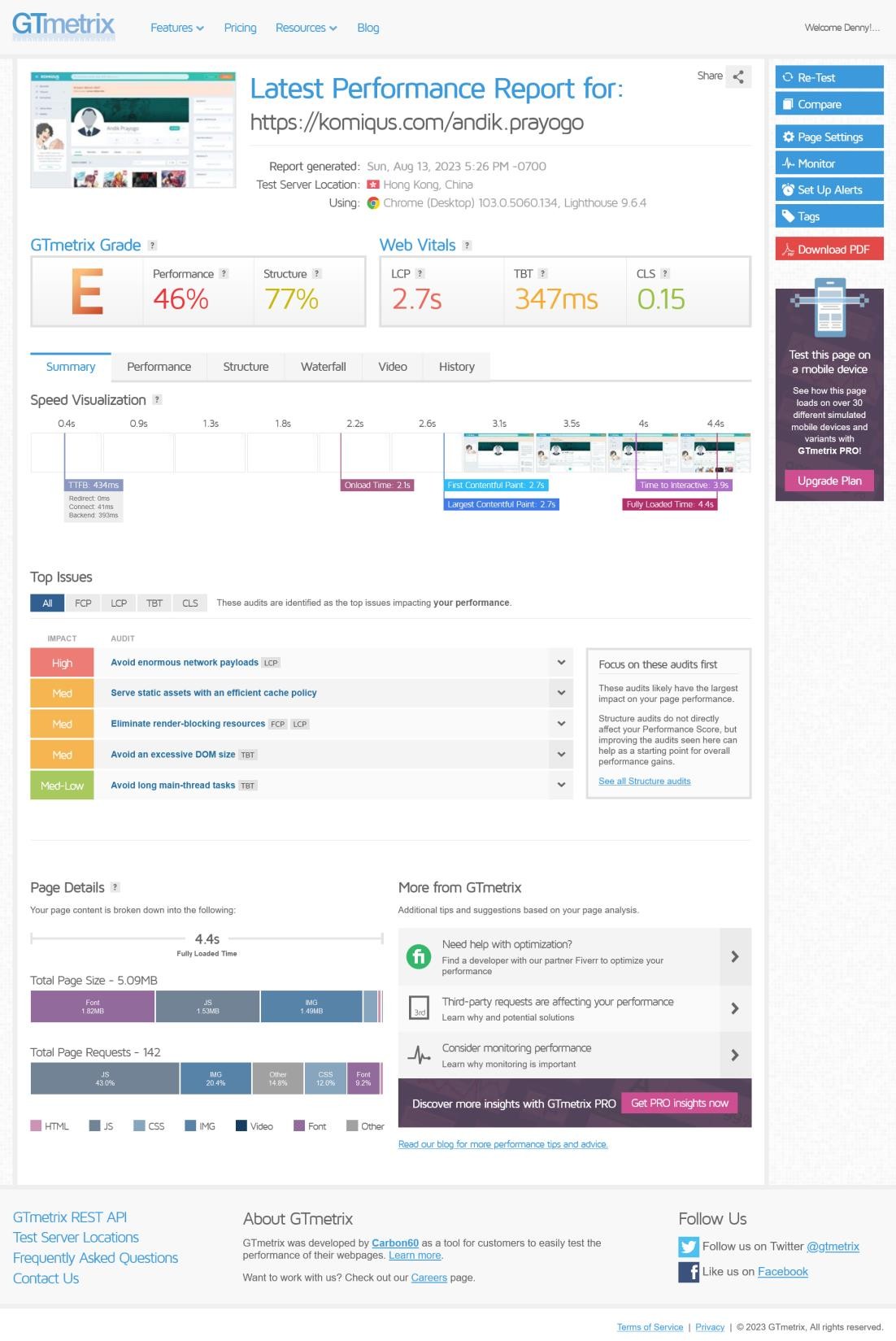
Hasil Pengujian Halaman Profil Kreator 300 VUs



Hasil Pengujian GMetrix Halaman Katalog



Hasil Pengujian GMetrix Halaman Utama



Hasil Pengujian GMetrix Halaman Profil Kreator