

**USULAN TUGAS AKHIR**

**1. IDENTITAS PENGUSUL**

**NAMA** : Cahya Putra Hikmawan  
**NRP** : 05111540000119  
**DOSEN WALI** : Wijayanti Nurul Khotimah, S.Kom, M.Sc.  
**DOSEN PEMBIMBING** : 1. Royyana Muslim I, S.Kom., M.Kom., Ph.D  
2. Bagus Jati Santoso, S.Kom., Ph.D.

**2. JUDUL TUGAS AKHIR**

“Implementasi *Headless Browser* untuk *Load Testing* Berbasis *Web Service* Menggunakan Infrastruktur *Docker*.”

**3. LATAR BELAKANG**

*Headless Testing* merupakan suatu teknik untuk melakukan pengujian aplikasi web khususnya *Document Object Model (DOM)* pada *browser* tanpa menampilkan *GUI browser*, pada umumnya fitur yang digunakan adalah *Headless Browser*. Teknik ini menyediakan akses kontrol yang sama seperti *browser* yang menggunakan *GUI*, namun yang membedakan adalah eksekusi dijalankan melalui *Command Line Interface (CLI)*.

*Headless Browser* banyak digunakan *developer* untuk melakukan suatu pengujian tampilan, contohnya yaitu *Headless Chrome*, *PhantomJS*, *Firefox Headless Mode*, *Selenium WebDriver*.

Saat ini aplikasi berbasis web sangat banyak dikembangkan, namun biasanya ada kegagalan load ketika diakses oleh banyak orang, *response time* yang lama ataupun yang lainnya, hal ini disebabkan karena tidak adanya *load testing* yang dilakukan sebelum peluncuran web tersebut. Oleh karena itu dibutuhkannya skenario

pengujian agar aplikasi yang diluncurkan tidak bermasalah saat digunakan penggunaanya dan tanpa harus melakukan pengujian manual satu-persatu. Untuk setiap pengujian diperlukannya *container* atau *virtual machine* sebagai *host* maupun *client* untuk menentukan jumlah *request* yang akan dijalankan untuk melakukan skenario pengujian. Pada Tugas Akhir ini diharapkan dapat memberikan solusi untuk melakukan pengujian aplikasi web menggunakan skenario yang diinginkan, infrastruktur yang digunakan akan menggunakan kontainer yaitu *docker*.

#### 4. RUMUSAN MASALAH

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana cara mengimplementasikan *Headless Chrome*?
2. Bagaimana *docker* bisa menjadi *load generator* untuk pengujian?
3. Bagaimana mengatur jumlah *worker* untuk pengujian?
4. Bagaimana cara menghasilkan laporan uji beban dalam bentuk *dashboard*?
5. Bagaimana cara mengimplementasikan untuk aplikasi di ITS?

#### 5. BATASAN MASALAH

Permasalahan yang dibahas dalam tugas akhir ini memiliki batasan antara lain:

1. *Headless browser* yang digunakan adalah *headless chrome*..
2. Kontainer yang digunakan adalah *docker*.
3. Aplikasi yang akan diuji berupa aplikasi web.
4. Uji coba aplikasi akan menggunakan API.

#### 6. TUJUAN PEMBUATAN TUGAS AKHIR

Tujuan pembuatan tugas akhir ini antara lain:

1. Membuat sistem manajemen pengujian aplikasi secara *headless* menggunakan *headless chrome*.
2. Membuat sistem agar *docker* bisa menjadi *load generator* untuk pengujian.
3. Mengimplementasikan pengujian menggunakan skenario yang sudah disiapkan sebelumnya.
4. Membuat sistem untuk menampilkan laporan uji beban dalam bentuk *dashboard*.
5. Mengimplementasikan sistem manajemen pengujian ini untuk aplikasi di ITS.

## 7. MANFAAT TUGAS AKHIR

Manfaat dari pembuatan tugas akhir ini adalah:

1. Mempelajari penggunaan *Headless Browser* untuk pengujian suatu aplikasi yaitu *Headless Chrome*.
2. Meminimalisir adanya *error loading* suatu aplikasi web ketika sudah diluncurkan.
3. Mengetahui performa *load* suatu aplikasi web.

## 8. TINJAUAN PUSTAKA

Dalam mengerjakan tugas akhir ini, terdapat beberapa tinjauan pustaka yang digunakan yaitu:

### 1. Headless Browser

*Headless Browser* [1] mengambil segala hak akses yang ada pada *Web Browser* normal, namun menghilangkan antarmuka pengguna. Hanya menyisakan mesin dan lingkungan *javascript*. Hal ini mengakibatkan penggunaan *Headless Browser* dilakukan menggunakan baris perintah atau *Command Line Interface (CLI)*, beberapa *headless browser* yaitu, *Headless Chrome*, *Selenium Webdriver*, *Firefox Headless Mode*.

### 2. Puppeteer

*Puppeteer* [2] adalah sebuah *Node Library* yang menyediakan *API* untuk mengontrol *Chrome* atau *Chromium* pada protokol pengembangan yang disediakan oleh *Google* untuk mengakses segala yang ada pada *Chrome* atau *Chromium*. Biasanya *puppeteer* berjalan secara *headless* tapi bisa juga dikonfigurasi menampilkan tampilan penuh untuk *Chrome* dan *Chromium*.

### 3. Docker

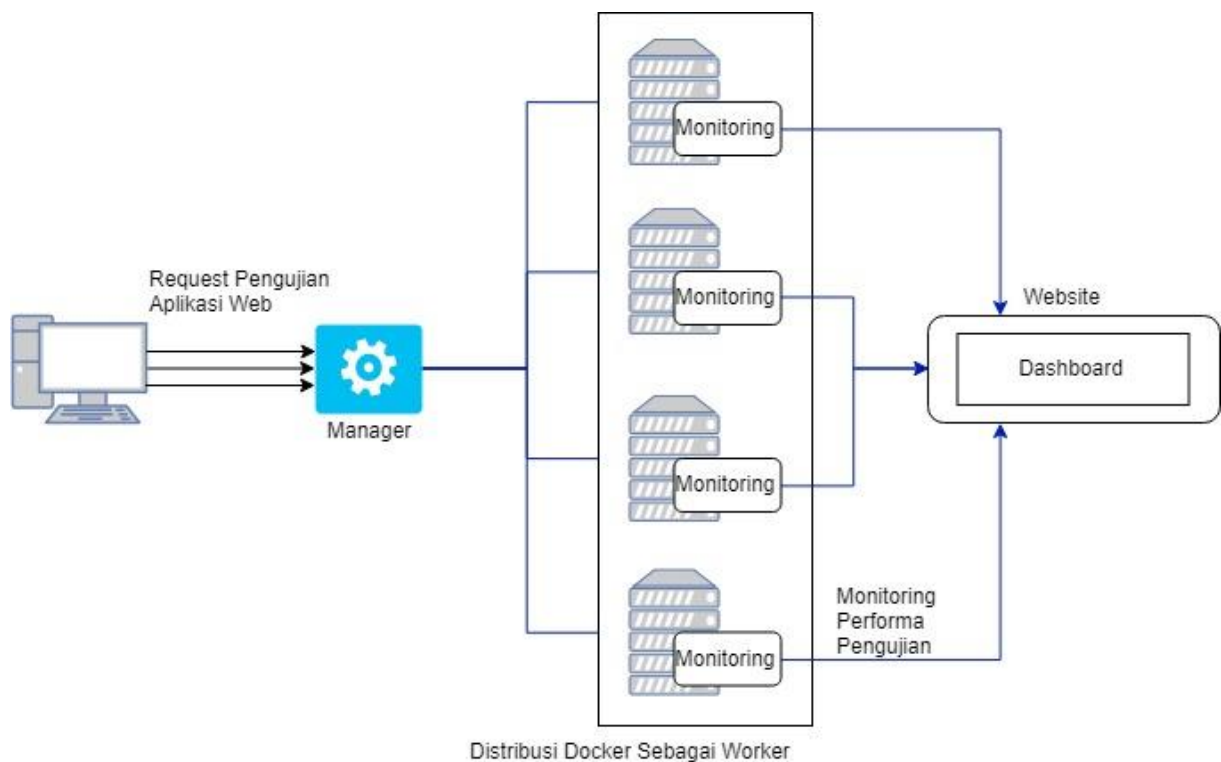
*Docker* [3] adalah sebuah *platform* terbuka untuk membangun, membungkus, dan menjalankan aplikasi. *Docker* memungkinkan untuk memisahkan aplikasi dari infrastruktur supaya software dapat di jalankan dengan lebih cepat. Dengan *Docker*, dimungkinkan untuk mengelola infrastruktur yang ada dengan cara yang sama untuk mengelola aplikasi. Dengan mengambil keuntungan dari metodologi *docker* untuk membungkus, melakukan tes dan melakukan *deploy* kode dengan cepat, *delay* antara penulisan kode dan menjalankannya pada saat produksi dan dikurangi dengan signifikan.

## 9. RINGKASAN ISI TUGAS AKHIR

Tugas akhir ini disusun untuk mengani masalah pengujian aplikasi *website* yang menggunakan teknik secara *Headless* dengan cara membuat skenario yang akan dihubungkan oleh *manager* kepada *worker* yang akan disediakan yaitu *docker*.

Proses pengujian akan diwalai adanya *request* dari *client* kemudian *manager* akan menerjemahkan apakah *client* membuat *request* pengujian secara manual atau menggunakan skenario, *request* akan dikirimkan ke *docker* untuk proses pengujian tampilan aplikasi menggunakan *Headless Browser*. Pengujian pada *docker* akan mengolah dokumen yang didapatkan dari *website* untuk dicari bagian yang masih mengalami kegagalan saat dimuat, hasil *output* bisa berupa screenshot untuk melihat hasil tampilan dan untuk *javascript* bisa dilihat dari hasil *DOM*, *output* lain juga bisa berupa *response time* dan persentase *load error*.

Skema model dari sistem ini dapat dilihat seperti bagan dibawah ini:



Gambar 1. Arsitektur

Setiap *request* akan direpresentasikan sebagai *worker* pada *docker*. Dan akan dilakukan *monitoring* untuk menghasilkan *log* dan *cache* seperti *web browser* pada umumnya. Setelah pengujian selesai maka akan menghasilkan laporan *dashboard* untuk mengetahui hasil pengujian *website* tersebut.

## 10.METODOLOGI

### a. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan ini terdiri atas hal yang menjadi latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat dari hasil pembuatan tugas akhir. Selain itu dijabarkan pula tinjauan pustaka yang digunakan sebagai referensi pendukung pembuatan tugas akhir. Sub bab metodologi berisi penjelasan mengenai tahapan penyusunan tugas akhir mulai dari penyusunan proposal hingga penyusunan buku tugas akhir. Terdapat pula sub bab jadwal kegiatan yang menjelaskan jadwal pengerjaan tugas akhir.

### b. Studi literatur

Pada studi literatur ini, akan dipelajari beberapa sejumlah referensi yang akan diperlukan untuk membangun aplikasi yaitu mengenai *Headless Browser* dan *Docker*.

### c. Analisis dan desain perangkat lunak

Aktor dari aplikasi ini adalah pengguna yang akan melakukan *request* untuk pengujian aplikasi pada *website*.

### d. Implementasi perangkat lunak

Aplikasi ini akan dibangun menggunakan *docker* sebagai *container* penyedia layanan yang akan dipasang di server dan bahasa pemrograman Python.

### e. Pengujian dan evaluasi

Pengujian dalam aplikasi ini akan dilakukan dalam beberapa cara antara lain:

#### 1. Pengujian *Request* dari Pengguna

Pengujian ini berfokus pada penerjemahan *request* dari pengguna agar bisa terhubung dengan *Headless Browser* untuk melakukan pengujian aplikasi web.

#### 2. Pengujian Dengan Skenario

Pengujian ini akan berfokus pada bagaimana pengujian aplikasi web menggunakan skenario yang sudah disiapkan sebelumnya.

### f. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari

implementasi aplikasi perangkat lunak yang telah dibuat. Sistematika penulisan buku tugas akhir secara garis besar antara lain:

1. Pendahuluan
  - a. Latar Belakang
  - b. Rumusan Masalah
  - c. Batasan Tugas Akhir
  - d. Tujuan
  - e. Metodologi
  - f. Sistematika Penulisan
2. Tinjauan Pustaka
3. Desain dan Implementasi
4. Pengujian dan Evaluasi
5. Kesimpulan dan Saran
6. Daftar Pustaka

## 11. JADWAL KEGIATAN

Tahapan	2018			2019															
	Desember			Januari			Februari			Maret			April			Mei			
Penyusunan Proposal																			
Studi Literatur																			
Perancangan Sistem																			
Implementasi																			
Pengujian dan Evaluasi																			
Penyusunan Buku																			

## 12. DAFTAR PUSTAKA

- [1] R. Roemer, Backbone.js Testing, Birmingham B3 2PB, UK: Packt Publishing Ltd, 2013.
- [2] "Puppeteer," [Online]. Available: <https://pptr.dev/>. [Accessed 1 12 2018].
- [3] "WHAT IS DOCKER?," [Online]. Available: <https://www.docker.com/>. [Accessed 1 12 2018].