**PENERAPAN DAN ANALISIS HTTP/2 *SERVER-SENT EVENTS* DAN WEBSOCKET UNTUK *WEB APPLICATION* PADA SISTEM RUMAH PINTAR**

<<ABSTRACT>>

Salah satu hal yang sering menjadi kendala dalam penerapan rumah pintar adalah cara mereka agar bisa terhubung dengan perangkat apapun di dalam rumahnya melalui internet. Pengguna harus tahu metode pengiriman apa yang diperlukan untuk mengirimkan atau menerima data dari *web browser* ataupun antarmuka lainnya menuju mikrokontroler dengan jeda pengiriman secepat mungkin. Permasalahan tersebut dapat dipecahkan dengan memanfaatkan beberapa teknologi sekaligus, semisal dengan menggabungkan WebSocket atau Server-Sent Events (SSE) dengan MQTT serta Serveo. Di antara WebSocket, HTTP/1.1 SSE, HTTPS SSE, serta HTTP/2 SSE perlu dilakukan pemilihan metode pengiriman yang cocok untuk digunakan pada sistem kendali rumah pintar. Dengan alasan tersebut, pengujian *response time* dan presentase penggunaan CPU dilakukan kepada keempat metode pengiriman tersebut. Hasil yang didapatkan adalah WebSocket memiliki nilai *response time* terkecil sedangkan HTTP/2 SSE memiliki nilai presentase penggunaan CPU terkecil dalam pengujian pengiriman dua arah.

**BAB 1 - Pendahuluan**

<<Latar Belakang>>

Konsep yang paling penting dari *Internet of Things* adalah mengintegrasikan semua hal yang ada di dunia nyata ke dalam dunia *digital* (Kwan et al. 2016). Salah satu penerapan dari *Internet of Things* adalah pengendalian rumah pintar melalui *web browser*. Hal yang perlu diperhatikan dalam pembuatan sistem kendali rumah pintar adalah kecepatan transaksi informasi, yang mana sebaiknya memiliki nilai *response time* serendah mungkin (Saito and Menga 2015). Dalam proses pengiriman data menuju *web browser* dengan rentang waktu sependek mungkin, terdapat berbagai pilihan yang dapat diterapkan pada rumah pintar, seperti halnya *Polling*, *Long-Polling*, WebSocketdan *Server-Sent Events*. Dari beberapa pilihan tersebut, *Polling* dan *Long Polling* termasuk metode pengiriman data yang harus secara aktif meminta data ke *server*. Teknologi *Polling* dan *Long-Polling* kurang cocok digunakan dalam rumah pintar karena keduanya membutuhkan *bandwith* dan *response time* yang besar, berbeda halnya dengan *WebSocket* ataupun *Server-Sent Events* (Souders 2009). Selain dari kecepatan transaksi data, rumah pintar cenderung dibuat secara sederhana. Karena itu, *server* yang digunakan untuk rumah pintar seringkali memiliki spesifikasi CPU maupun memori yang lebih rendah.

<<Tinjauan Literatur singkat>>

- Ketika penggunaan *resource server* dinaikkan, *response time* WebSocket relatif tetap (Paridhika Khayal, dkk. 2017)

- Besar latensi aplikasi pada penggunaan HTTP/2 SSE serta WebSocket relatif sama serta WebSocket lebih rentan di-*block* oleh *proxy server* dibandingkan HTTP/2 (Kalamullah Ramli, dkk. 2018)

- Rata-rata delay pada protokol SSE lebih kecil dibandingkan dengan WebSocket begitu juga dengan presentase penggunaan CPU (Panser Brigade Muhammad, dkk. 2018)

- Penggunaan CPU, memori maupun bandwith pada SSE serta WebSocket lebih kecil dibandingkan dengan XHR *Polling* serta *Long-Polling* (Oliver Örnmyr, dkk. 2017)

- Performa SSE maupun Websocket juga dipengaruhi oleh *web browser* serta konfigurasi jaringan yang digunakan (Elliot Estep, 2013)

<<Rumusan - option>>

Rumusan permasalahan pada penelitian ini adalah bagaimana cara menerapkan HTTP/2 *Server-Sent Events* ataupun WebSocket pada sistem rumah pintar berbasis website serta untuk mengetahui hasil perbandingan *response time* serta presentase penggunaan CPU pada HTTP/1.1 SSE, HTTPS SSE, HTTP/2 SSE dan WebSocket.

<<Alasan - option>>

Baik *WebSocket* maupun *Server-Sent Events* masih memerlukan penelitian lebih jauh untuk mencari teknologi mana yang memiliki nilai *response time* paling rendah serta menggunakan *resource* di *server* terendah pula ketika diterapkan di rumah pintar, dengan alasan itulah penelitian ini dilakukan.

**BAB 2 - Metodologi Penelitian**

<<Peralatan dan Bahan>>

**Peralatan**

Berikut merupakan daftar peralatan yang akan digunakan selama proses penelitian berlangsung :

1. Komputer sebagai client dengan spesifikasi:

CPU : Intel Celeron 1.50GHz x 4

Hardisk : 750 GB

RAM : 8 GB

OS : Linux Mint 18.2 Cinnamon 64-bit

2. Raspberry Pi 3 Model B sebagai broker sekaligus web server dan server API sebanyak 1 unit dengan spesifikasi:

CPU : 1.2 GHz quad-core ARM

Memory : 1 GB LPDDR2-900 SDRAM

USB Port : 4

Network : 10/100 Mbps Ethernet, 802.11 n Wireless LAN

3. NodeMCU 1 Unit dengan spesifikasi:

MCU : Xtensa Dual-Core 32-bit L106

Wifi : 802.11 b/g/n HT40

Typical Frequency : 160 MHz

Tipe ESP : ESP32

4. NodeMCU 1 Unit dengan spesifikasi:

MCU : Xtensa Single-Core 32-bit L106

Wifi : 802.11 b/g/n HT20

Typical Frequency : 80 MHz

Tipe ESP : ESP8266

5. Access Point sebanyak – 1 unit

6. Sensor DHT11 – 2 unit

7. LED Putih – 13 unit

8. Servo Motor SG90 – 2 unit

9. Motor DC 3 volt – 1 unit

10. Cooling fan DC 5 volt – 1 unit

11. Kabel Jumper

**Bahan**

Berikut merupakan daftar perangkat lunak yang akan digunakan selama penelitian ini berlangsung :

1. Mosquitto, sebagai broker untuk protokol MQTT

2. Node.js, sebagai web server yang khusus Javascript

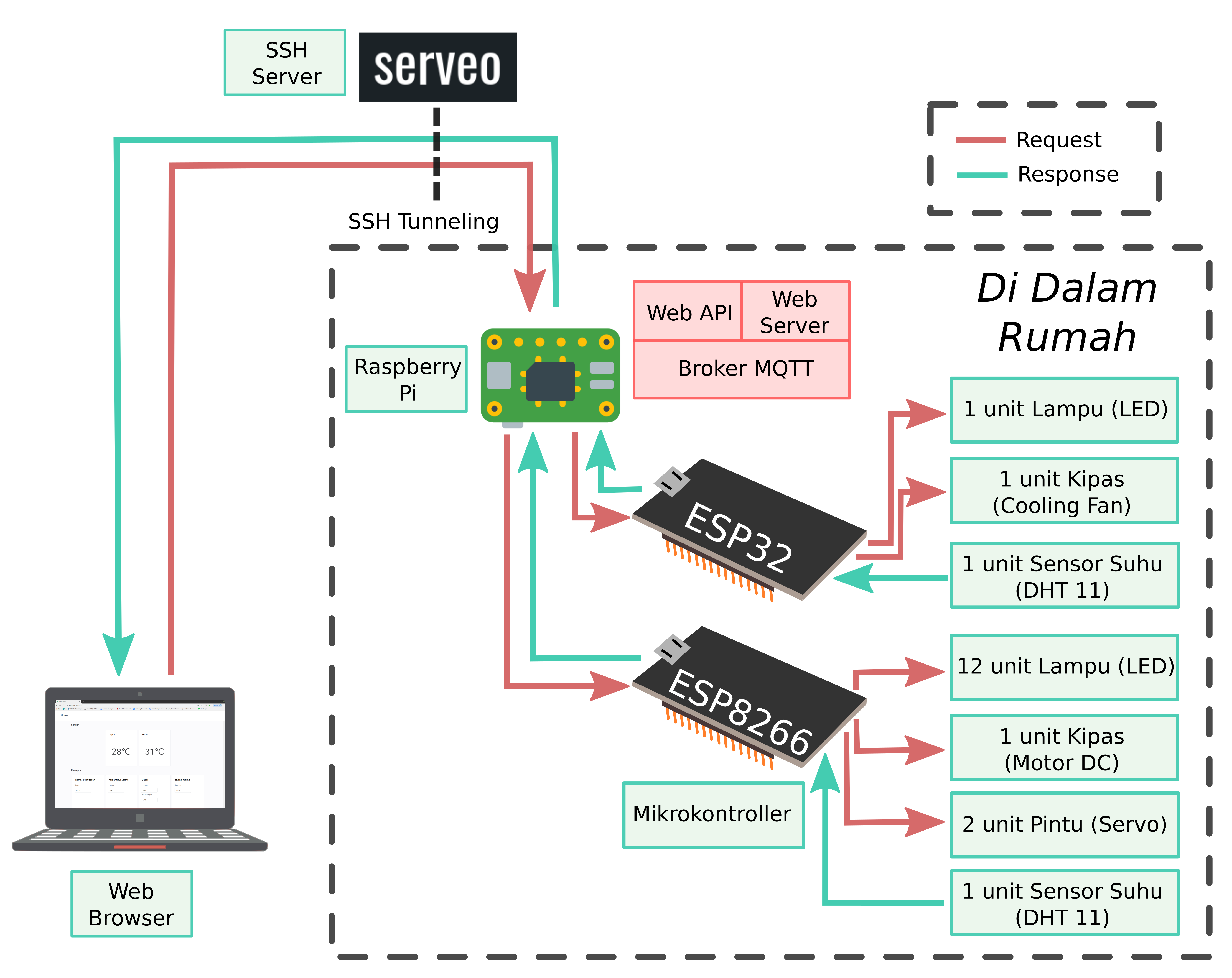
3. Vue.js, sebagai javascript framework untuk pembuatan web application

4. OpenSSL, untuk pembuatan public key dan private key pada HTTPS serta HTTP/2

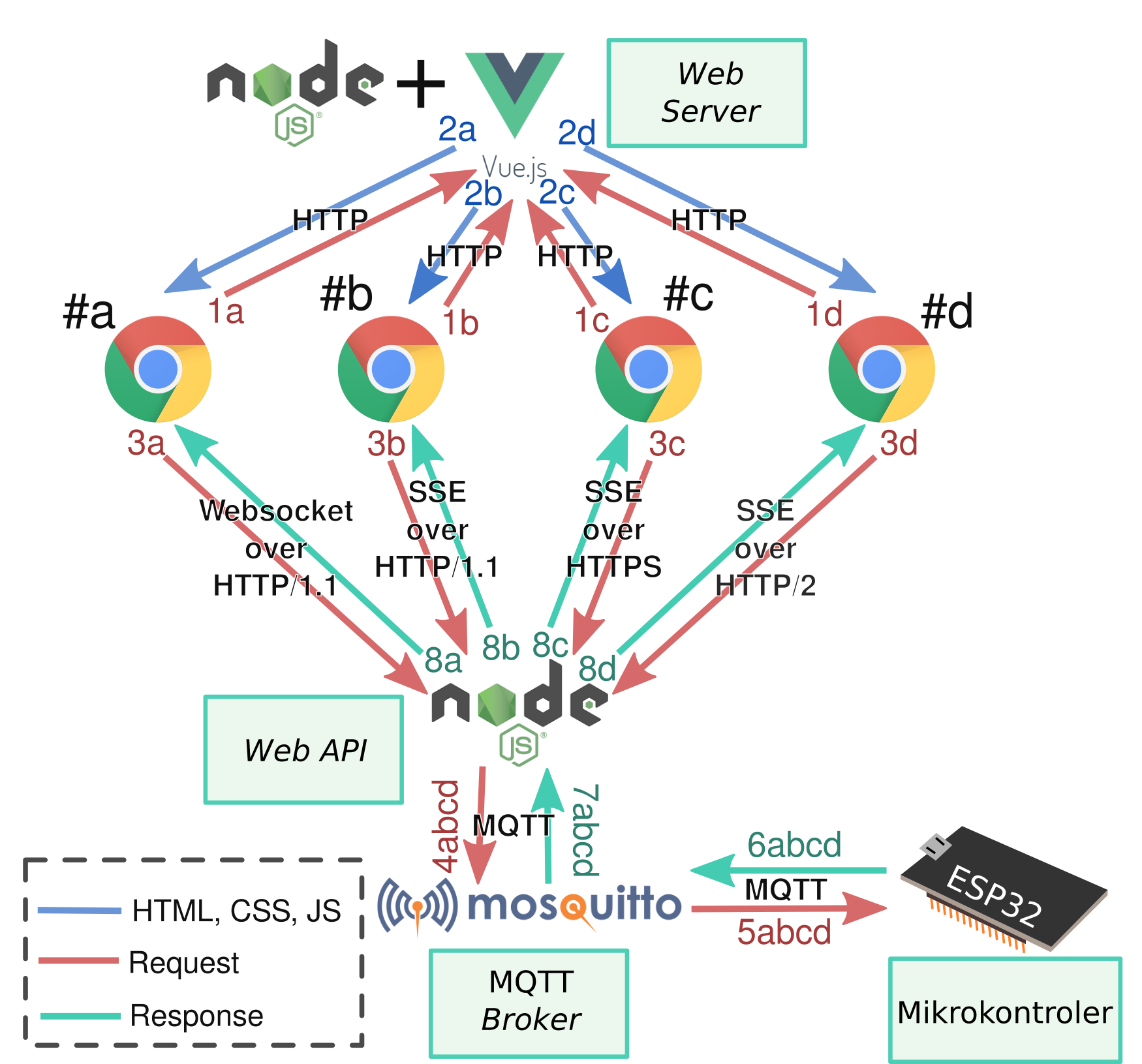
5. Serveo, untuk menjadikan local server mampu diakses secara publik

6. Google Chrome, untuk mengakses website

<<Topologi data>>



<<Topologi Pengujian ( Data )>>



<<Hasil dan Pembahasan>>

**Response Time**

Pengambilan data *response time* untuk keempat metode dilakukan dengan cara menghitung selisih waktu dari dua puluh perintah yang dikirimkan dari *web browser* sampai mendapatkan balasan dari mikrokontroler. Keduapuluh perintah dikirimkan dengan jeda waktu pengiriman yang bervariasi serta memiliki ukuran paket yang sama. Perhitungan waktu dilakukan dengan menandai waktu awal pengiriman perintah serta waktu akhir ketika pesan balasan diterima dari sisi *web browser*. Proses pengambilan *response time* dilakukan dalam dua skenario yang berbeda, yakni di dalam jaringan lokal yang sama serta melalui jaringan internet.

**Penggunaan CPU**

Pengambilan datapresentasepenggunaanCPU dilakukan dengan menggunakan bantuan aplikasi ‘top’. Skenario pengambilan data presentase penggunaan CPU dilakukan dengan cara mengirimkan perintah setiap detiknya dari beberapa *web browser* menuju mikrokontroler secara bersamaan. Setiap *web browser* diibaratkan sebagai satu pengguna. Setiap perintah yang dikirimkan dari *web browser* akan dibalas oleh mikrokontroler. Perintah yang dikirimkan dari *web browser* memiliki ukuran paket yang sama, begitu juga dengan perintah yang dikirimkan dari mikrokontroler.

Dua w*eb browser* yang sama ketika dibuka pada waktu yang bersamaan akan menggunakan *session*, *history*, maupun *cache* yang sama. Hal ini tidak mengganggu proses pengambilan data WebSocket maupun HTTP/2 SSE yang memiliki karakteristik paralel, namun tidak dengan HTTP/1.1 SSE maupun HTTPS SSE yang hanya mampu membuka enam TCP *Connection.* Untuk mengatasi hal tersebut dibuat *session* yang berbeda untuk setiap membuka *web browser* dengan menjalankan *script* tertentu di terminal Linux.