**LEMBAR KEGIATAN PESERTA DIDIK**

**(LKPD)**

Nama Siswa :

Nomor Induk SIswa :

Kelas/Rombel :

Topik Materi :

Tujuan Pembelajaran

Setelah menyelesaikan kegiatan pembelajaran sesuai instruksi, peserta didik mampu: 1) mengetahui apakah Laravel Tinker?; 2) mendeskripsikan laravel tinker; 3) membuat aplikasi mikrokontroler dengan protokol komunikasi MQTT secara mandiri.

**Kegiatan 1: Memahami Konsep Laravel Tinker**

Deskripsi kegiatan:

Disediakan sumber belajar (*website*) tentang Laravel Tinker di *internet*, peserta didik membaca, memahami, merangkum konsep Laravel Tinker. Jika dirasa kurang, peserta didik boleh menggunakan sumber belajar yang berbeda.

Langkah Kerja:

1. Kunjungi *website* berikut: (*literasi*)
2. <https://dev.to/japhethjoepari/php-artisan-tinker-a-practical-guide-for-laravel-developers-19of>
3. <https://www.javatpoint.com/laravel-tinker#:~:text=The%20php%20artisan%20is%20a,%2C%20insert%20the%20data%2C%20etc.&text=The%20above%20screen%20shows%20that%20the%20tinker%20environment%20has%20been%20created>.
4. <https://tinkerwell.app/blog/the-ultimate-guide-to-php-artisan-tinker>
5. Merangkum hasil literasi:

Berdasarkan hasil literasi, tuliskan pemahaman konsep Laravel Tinker berikut:

1. Pengertian PHP Artisan Tinker

PHP Artisan Tinker adalah *Command Line Interface (CLI)* di dalam Laravel yang mampu berinteraksi dengan aplikasi dan data secara langsung.

1. Pengertian REPL (*Real-Eval-Print-Loop*)

REPL adalah lingkungan interaktif di mana Anda dapat menulis kode, mengevaluasi ekspresi, dan melihat hasilnya secara langsung tanpa harus menulis program lengkap. Tinker adalah salah satu contoh REPL di Laravel.

1. Penemu Laravel Tinker

Protokol MQTT ditemukan oleh Andy Stanford-Clark dari IBM dan Arlen Nipper dari Arcom (sekarang Cirrus Link)

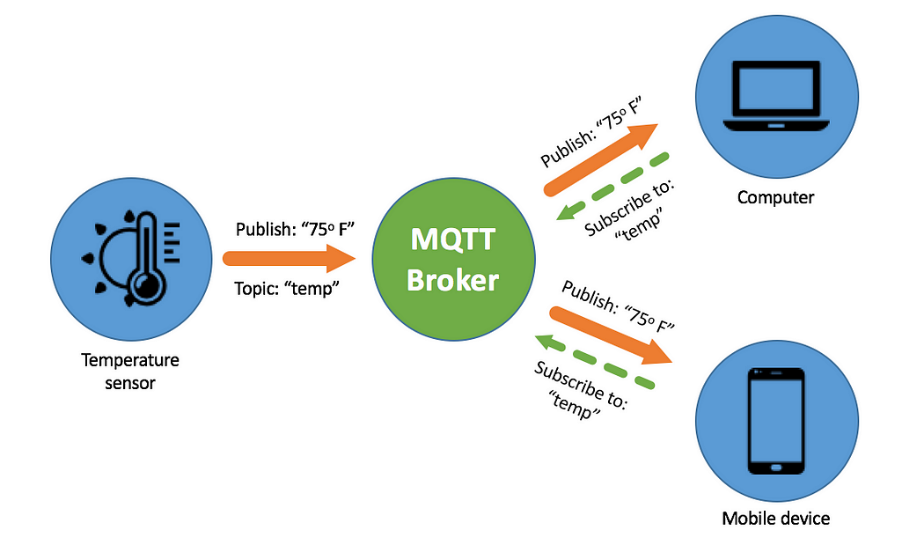
1. Tahun penemuan MQTT

Protokol MQTT dikembangkan pada tahun 1999

1. Komponen MQTT

MQTT terdiri dari 4 komponen, yaitu: *Broker, Publisher, Subscriber, Topic*

1. Mendeskripsikan hasil literasi (How)
2. Menampilkan gambar komunikasi protokol MQTT



Gambar 1. Ilustrasi protokol MQTT

Sumber: <https://medium.com/pujanggateknologi/berkenalan-dengan-teknologi-mqtt-7e63cab9d00d>

1. Deskripsi *Publisher*

Mengirimkan pesan ke *broker* dengan topic tertentu. Perangkat mikrokontroler yang mengirimkan pesan ke *broker* dengan topic tertentu disebut *publisher*, contoh perangkat: sensor suhu, sensor gempa, dan lain-lain.

1. Deskripsi *Broker*

Menerima pesan dari *publisher* dan meneruskannya ke *subscriber* sesuai topic.

Broker MQTT adalah sistem *backend* yang mengoordinasikan pesan antara *client* yang berbeda.

Tanggung jawab *broker* meliputi menerima dan memfilter pesan, mengidentifikasi *client* yang berlangganan setiap pesan, serta mengirimi pesan kepada *client*.

Broker juga bertanggung jawab untuk tugas-tugas lain seperti:

1. Mengotorisasi dan mengautentikasi klien MQTT
2. Mengirim pesan ke sistem lain untuk analisis lebih lanjut
3. Menangani pesan tak terjawab dan sesi klien
4. Deskripsi *Subscriber*

Menerima pesan dari *broker* berdasarkan topic yang mereka langganan

1. Deskripsi *Topic*

Setiap pesan dikirim *publisher* dan *subscriber* memiliki kategori yang dinamakan dengan topic

1. Cara kerja protokol MQTT

MQTT menggunakan model komunikasi *publish-subscribe*, di mana perangkat *publisher* mengirim pesan ke *broker* dengan topic tertentu. *Broker* menerima dan memfilter pesan pesan dari *publisher*, mengidentifikasi *subscriber* yang berlangganan setiap pesan sesuai topic, mengirimkan pesan ke *subscriber*.

1. Kelebihan protokol MQTT (Why)
2. Ringan dan Efisien

MQTT memiliki *overhead* yang rendah karena ukuran *header* pesan yang kecil dan penggunaan *bandwidth* yang minim. Hal ini membuatnya cocok untuk perangkat dengan sumber daya terbatas, seperti sensor atau perangkat baterai.

1. Koneksi yang Handal

MQTT dirancang untuk menjaga koneksi yang andal di lingkungan jaringan yang tidak stabil. Jika koneksi terputus, protokol ini memiliki mekanisme untuk menyimpan pesan-pesan yang belum terkirim dan mengirimkannya kembali setelah koneksi terhubung kembali.

1. Skalabilitas

MQTT dapat dengan mudah mengakomodasi jutaan perangkat terhubung secara bersamaan. Dengan arsitektur *publish-subscribe*, pesan dapat dikirim ke banyak *subscriber* secara bersamaan, sehingga memungkinkan skalabilitas tinggi dalam sistem IoT yang besar.

1. Desain yang Sederhana

MQTT memiliki desain yang sederhana dan mudah diimplementasikan. Hal ini membuatnya menjadi pilihan yang populer bagi pengembang perangkat IoT.

1. Dukungan untuk *Quality of Service* (QoS)

MQTT mendukung tiga level *Quality of Service*: QoS 0 (*fire and forget*), QoS 1 (setidaknya sekali), dan QoS 2 (pastikan persis sekali). Ini memungkinkan pengiriman pesan yang dapat diandalkan dan menyesuaikan kebutuhan aplikasi.

1. Penggunaan Port

MQTT menggunakan *port default* 1883 untuk komunikasi non-terenkripsi dan *port* 8883 untuk komunikasi dengan enkripsi TLS/SSL.

1. Keamanan

MQTT tidak memiliki lapisan keamanan bawaan, tetapi dapat di kombinasikan dengan protokol keamanan tambahan, seperti TLS/SSL, untuk mengamankan komunikasi. Keamanan di implementasikan pada tingkat transportasi.

**Kegiatan 2: Menerapkan konsep protokol MQTT | *Chatting***

Deskripsi kegiatan:

Peserta didik menearpkan konsep tentang protokol MQTT dengan mempraktikkan berkomunikasi *chatting* menggunakan protokol MQTT.

Skenario

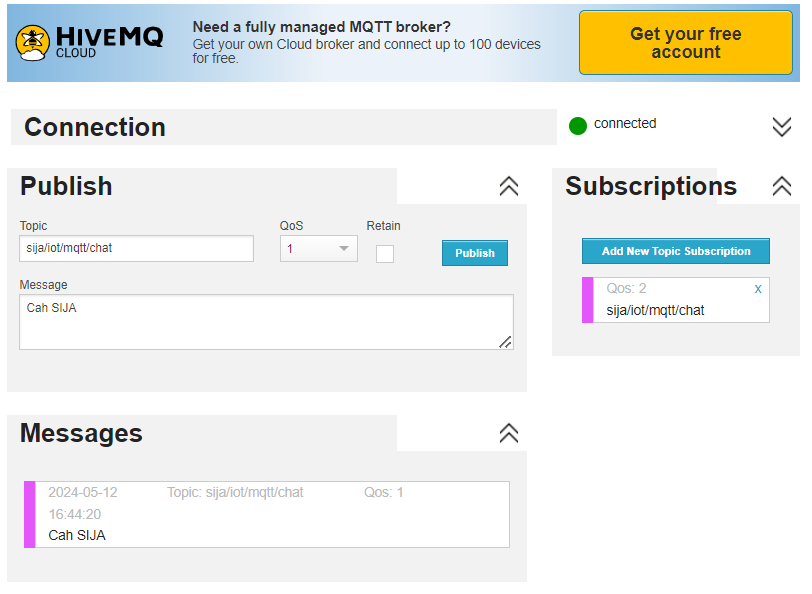
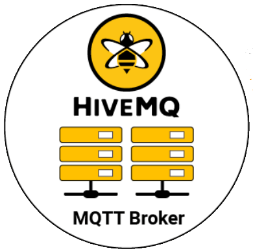
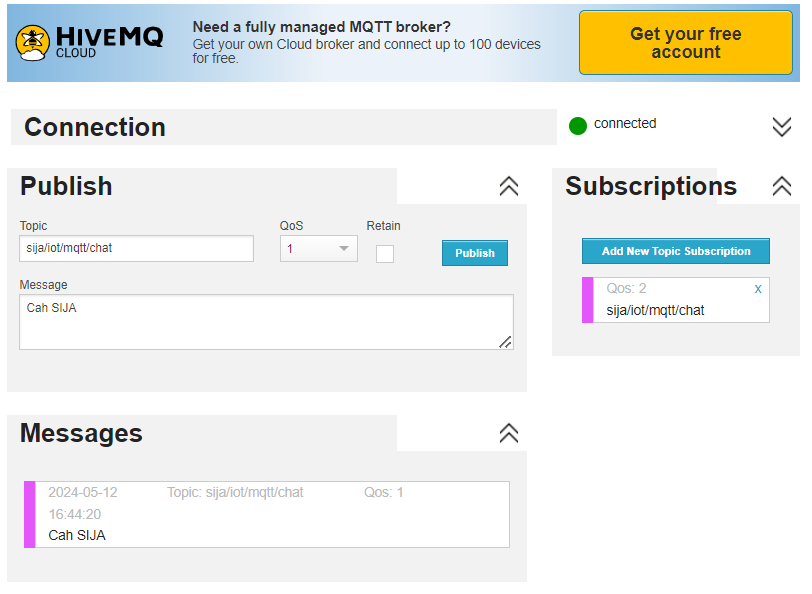
Broker : [broker.hivemq.com](http://broker.hivemq.com/)

Port : 1883

*Publisher*/*Subscriber* *tool* : Free MQTT *Browser* *Client* by HiveMQ

Topic : sija/iot/mqtt/chat

Desain/Ilustrasi

Gambar 2. Ilustrasi *chatting* menggunakan protokol MQTT

Sumber Belajar

<https://www.hivemq.com/mqtt/public-mqtt-broker/>

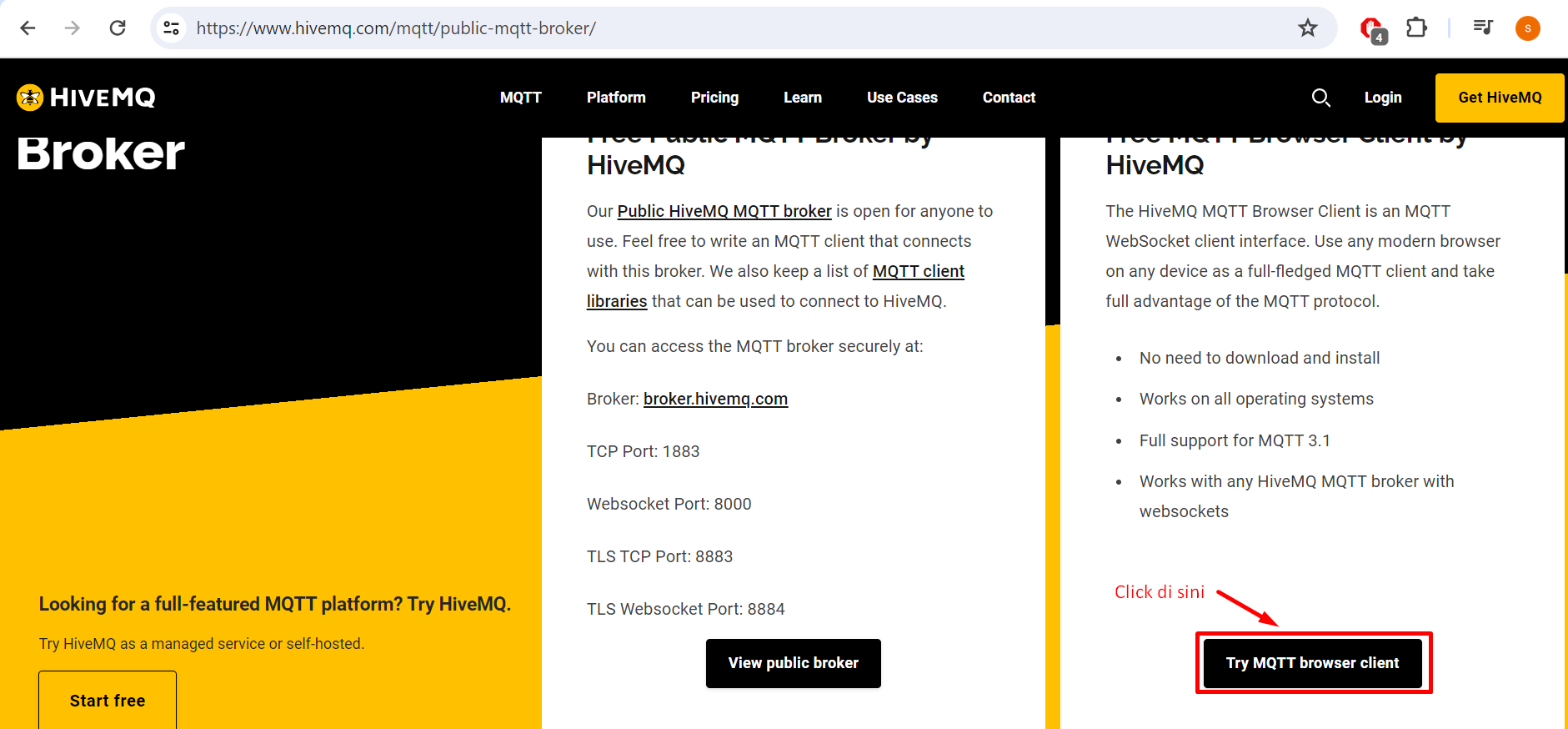
Langkah Kerja:

1. Membuka *website* MQTT *Client*

Aksi :

Kunjungi *website*: <https://www.hivemq.com/mqtt/public-mqtt-broker/>

Hasil



Gambar 3. *Browser* MQTT *Client*

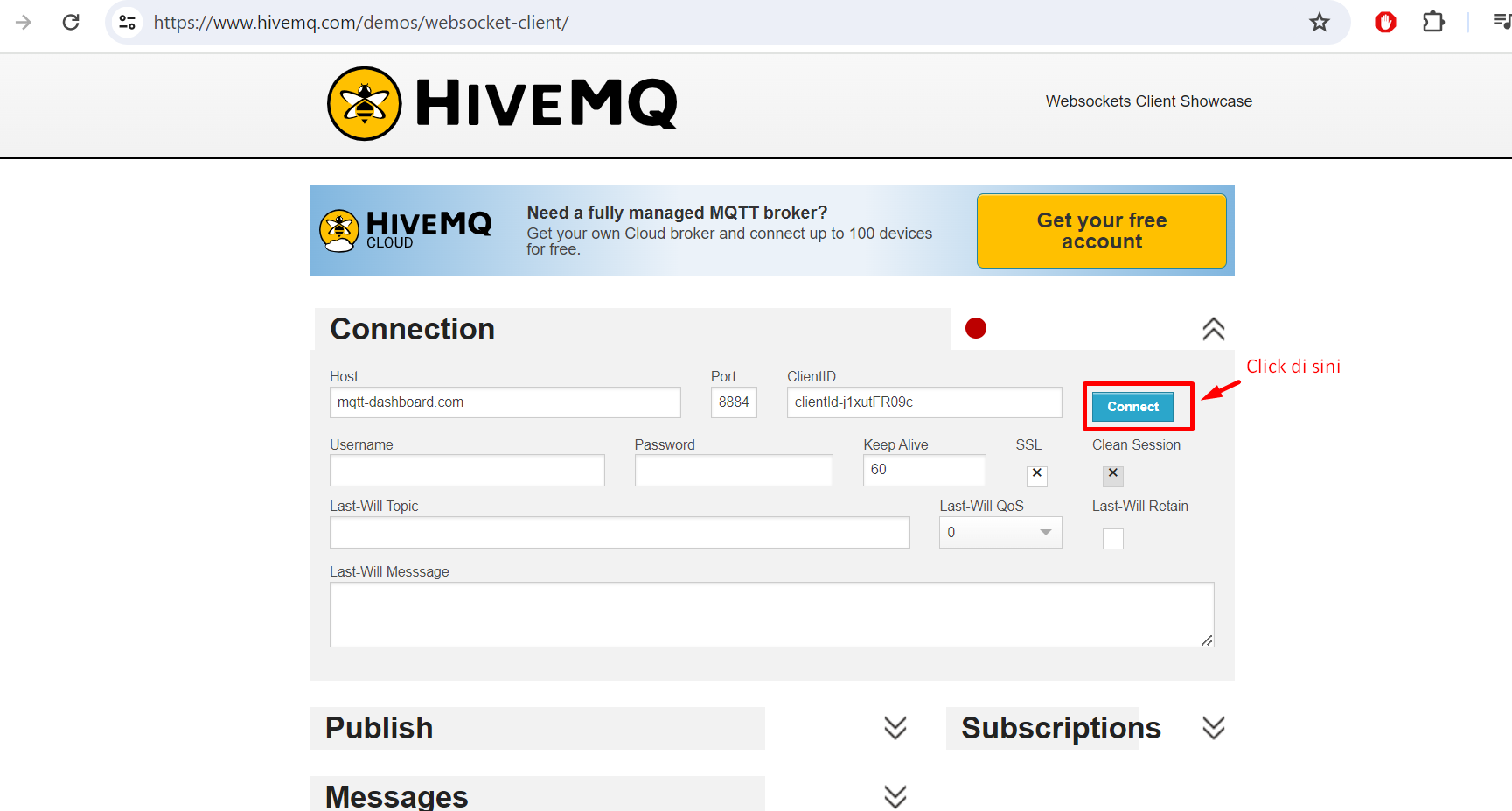
1. Mengkoneksikan *Client* ke MQTT *Broker*

Aksi :

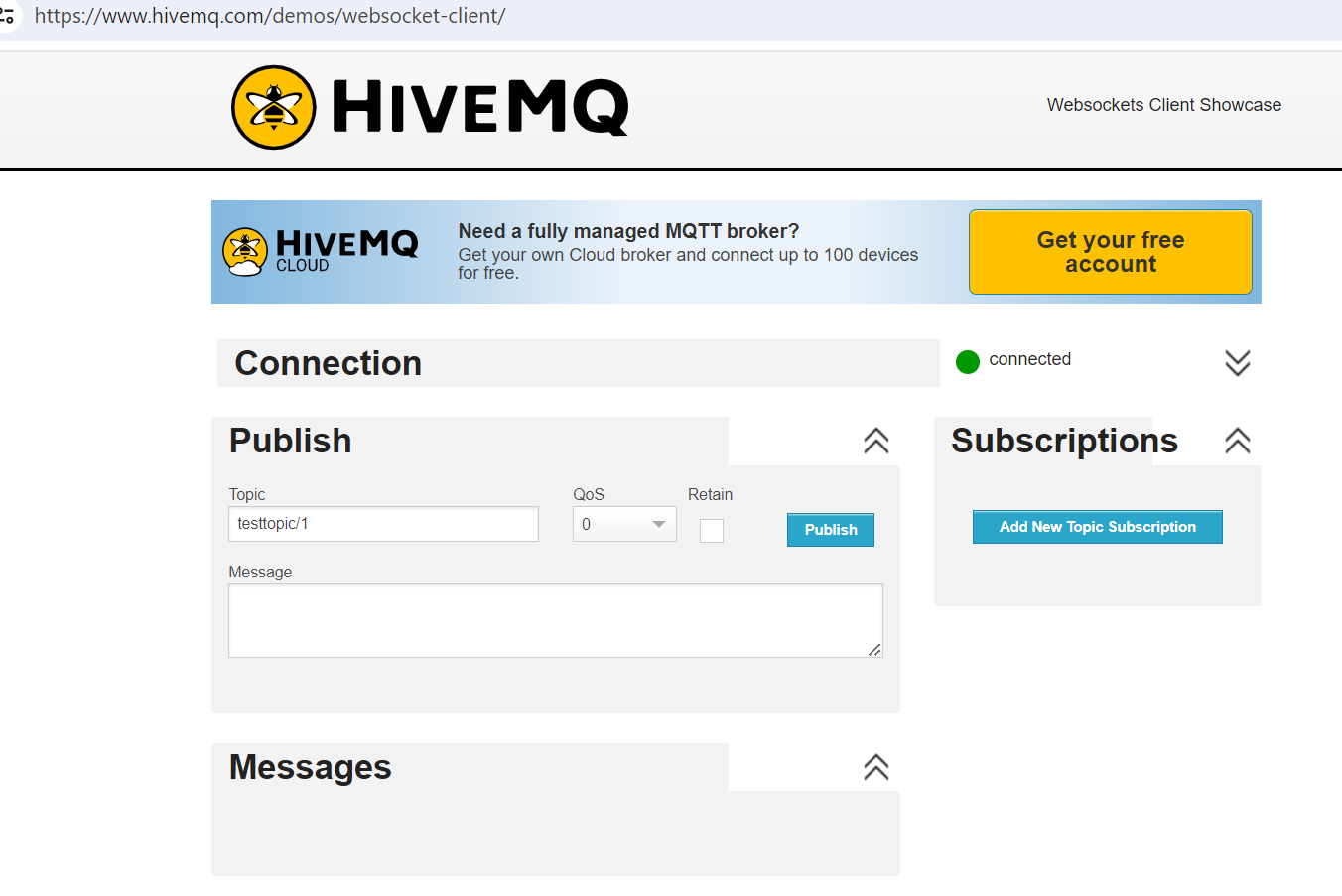
Isikan data: MQTT *Broker*, MQTT Port, *Username*, *Password*

(MQTT *Broker* HIVEMQ versi gratis, gunakan data *default*, langsung *click* tombol **Connect**)

Hasil



Gambar 4. Konfigurasi data di MQTT *Client*



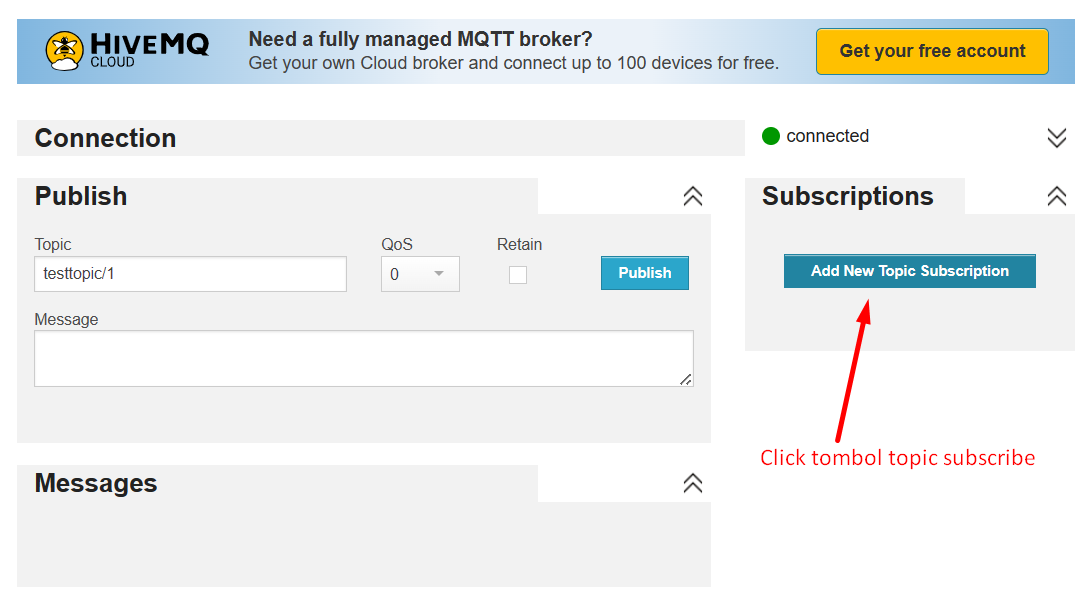
Gambar 5. Koneksi MQTT *Client* ke MQTT *Broker* berhasil

1. Mengkonfigurasi MQTT *Client* sebagai *Subscriber*

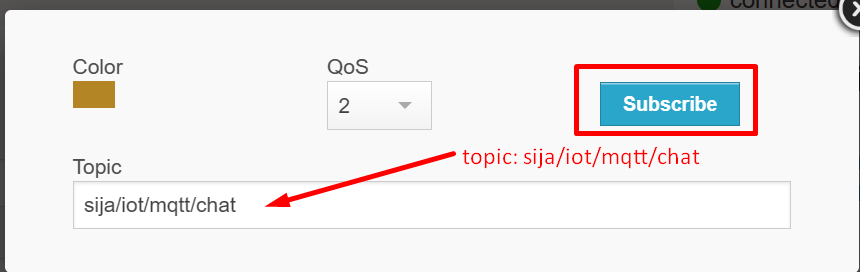
Aksi :

Buat topic baru dengan cara *click* tombol **Add New Topic Subscription**. Isikan topic: sija/iot/mqtt/chat

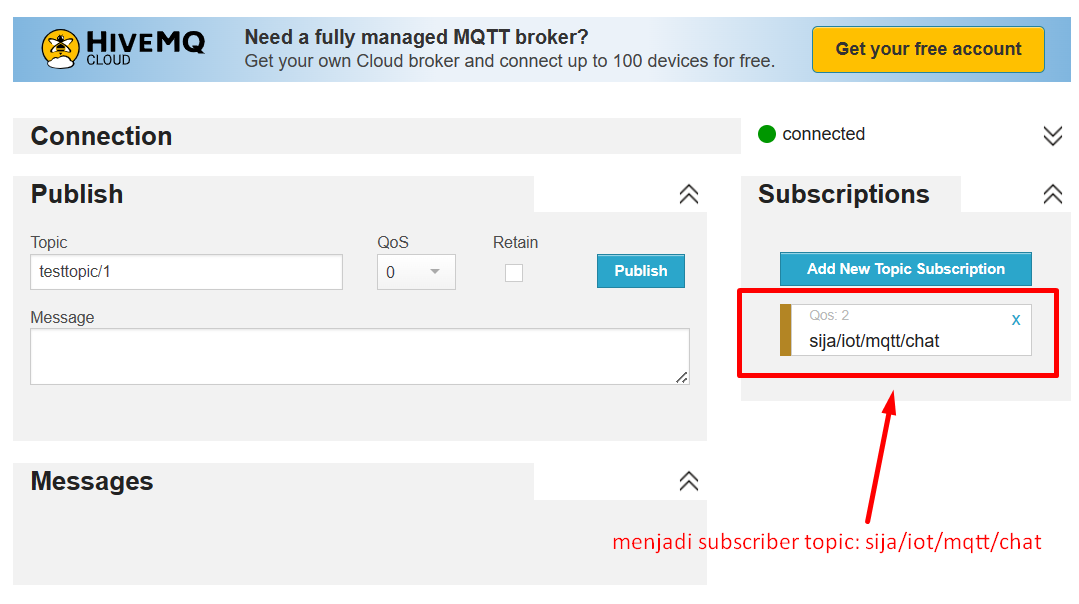
Hasil



Gambar 6. Tombol membuat topic *subscribe* baru



Gambar 7. Membuat Topic *Subscribe* baru



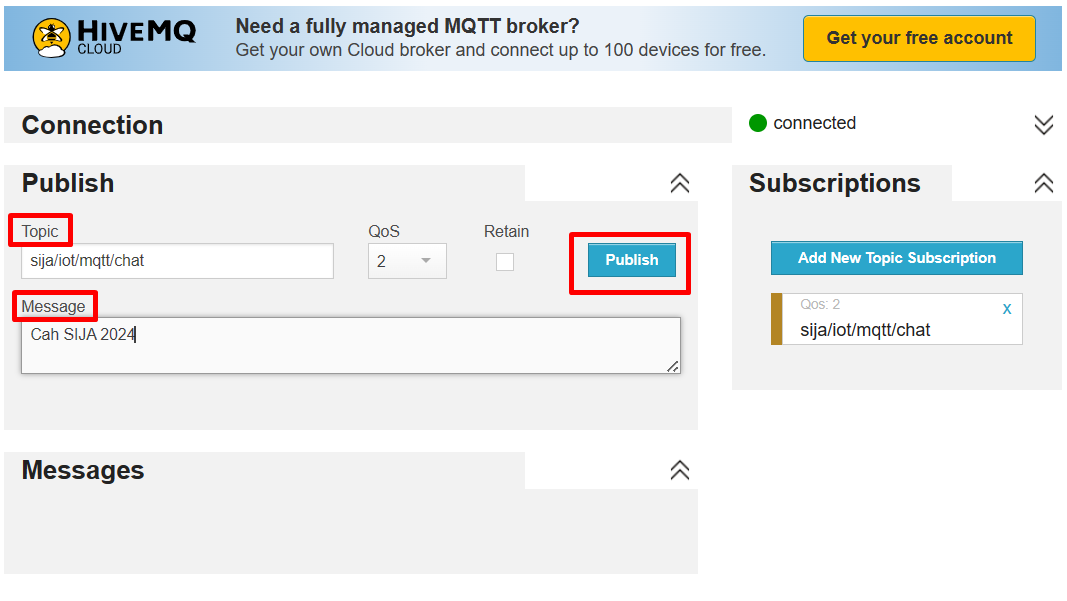
Gambar 8. Menjadi *Subscriber* topic: *sija/iot/mqtt/chat*

1. Mengkonfigurasi MQTT *Client* sebagai *Publisher*

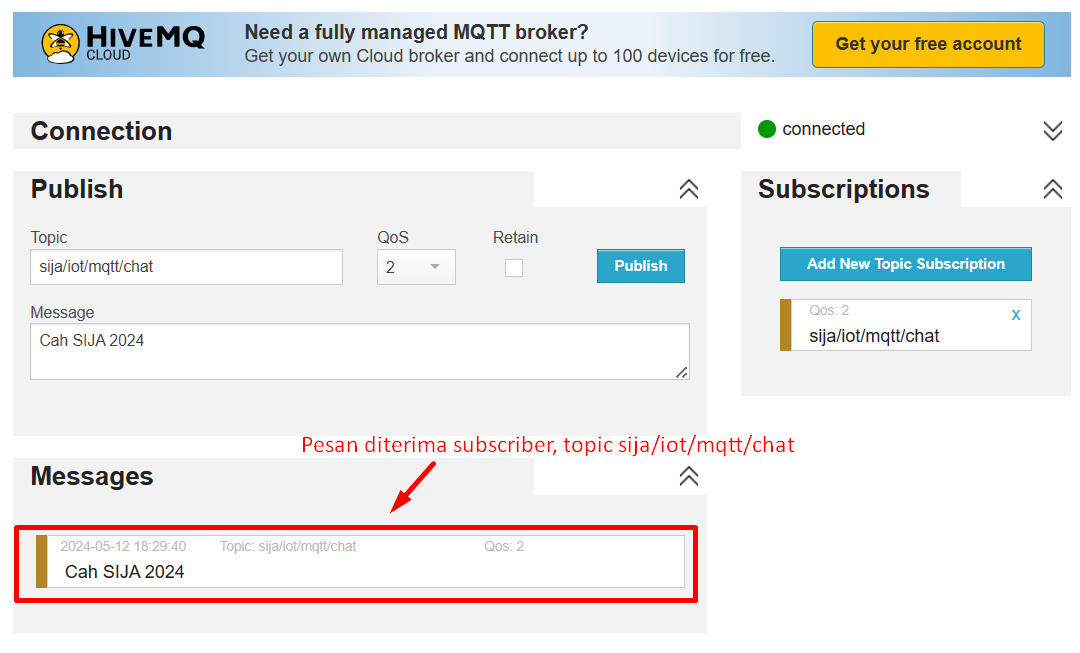
Aksi :

Isikan data topic pada *textbox* topic: (contoh) *sija/iot/mqtt/chat*, mengisi pesan yang akan dikirim (contoh) *Cah SIJA 2024* pada *textarea*, *click* tombol **Publish**!

Hasil



Gambar 9. Tombol membuat topic *subscribe* baru



Gambar 10. Tombol membuat topic *subscribe* baru

1. Selesai

# Bibliography

Joepari, J. (2024, Juni 18). *PHP Artisan Tinker: A Practical Guide for Laravel Developers*. Retrieved from https://dev.to: https://dev.to/japhethjoepari/php-artisan-tinker-a-practical-guide-for-laravel-developers-19of