

# ASISTEN KELAS INTERAKTIF MOODLE BERBASISKAN INTERNET OF THINGS DAN TELEGRAM BOT

Alauddin Maulana Hirzan<sup>1\*</sup>, Surono<sup>2</sup>, Galih Sertiarso<sup>3</sup>

<sup>1,2,3</sup>Universitas Semarang; Jl. Soekarno Hatta, Tlogosari Kulon; 024 6702757

Received: xxxx-xx-xx

Accepted: xx-xx-xx

## Keywords:

Class Assistant,  
Internet of Things,  
Moodle,  
Telegram Bot

## Correspondent Email:

maulanahirzan@usm.ac.id

**Abstrak.** Pendidikan adalah aspek kehidupan yang juga menjadi hak asasi dari tiap manusia. Oleh karena itu, pendidikan tidak boleh putus dalam keadaan apapun. Semenjak terjadinya pandemi di seluruh dunia, popularitas platform *Moodle* sebagai tempat pembelajaran daring meningkat. Namun hal ini tidak diimbangi dengan sumber daya manusianya. Banyak sekali para pengajar senior yang tidak familiar dengan platform tersebut. Oleh karena itu, peneliti mengusulkan untuk mendesain sebuah asisten kelas interaktif berbasis *Internet of Things* dan *Telegram Bot* untuk membantu para pengajar untuk mengakses *Moodle* hanya melalui aplikasi berpesan instan. Dari hasil pengujian yang dilakukan, model tersebut telah sukses melayani 117.628 permintaan. Namun juga disertai permintaan dalam antrian mencapai 75.526 permintaan dan permintaan gagal mencapai 3.175 permintaan. Berdasarkan kecepatan waktu respons nya, model dapat melayani 1000 pengguna dengan rata-rata waktu 66,86 detik. Bisa disimpulkan bahwa model yang diusulkan berhasil melayani banyak pengguna meskipun memiliki kegagalan yang tidak signifikan.

**Abstract.** Education is an aspect of life that is also the birthright of every human being. Therefore, education should not be interrupted under any circumstances. Since the worldwide pandemic, the popularity of the Moodle platform for online learning has increased. However, this is not matched by its human resources. Many senior teachers are not familiar with the platform. Therefore, the researcher proposes to design an interactive classroom assistant based on the Internet of Things and Telegram Bot to help teachers access Moodle only through instant messaging applications. From the test results, the model has successfully served 117,628 requests. But also accompanied by requests in the queue reaching 75,526 requests and failed requests reaching 3,175 requests. Based on its response time speed, the model can serve 1000 users with an average time of 66.86 seconds. It can be concluded that the proposed model successfully serves many users despite having insignificant failures.

## 1. PENDAHULUAN

*Moodle* merupakan sebuah perangkat lunak yang difokuskan dalam dunia pendidikan di berbagai macam tingkat. Implementasi teknologi merupakan sebuah bentuk dari majunya dunia pendidikan dalam mengikuti perkembangan teknologi masa kini. Banyak

sekali fitur yang dihadirkan oleh perangkat lunak ini untuk membantu para pengajar (guru maupun dosen) untuk menata materi maupun penilaia kelasnya[1], [2]. Para guru maupun dosen dapat membuat sebuah kelas interaktif menggunakan *plugin* yang ada di *Moodle*, sehingga kelas yang diajarkan lebih terlihat

menyenangkan dan mudah dipahami[3]. Selain itu karena sifatnya yang daring, pengaksesannya dapat dilakukan di mana saja dan perangkat apapun. Sehingga peserta didik tidak harus berada di sekolah ataupun kampus untuk mempelajari materi yang diberikan[4]. Karena kemudahan yang dihadirkan inilah, *Moodle* banyak diimplementasikan di banyak tingkat pendidikan di semua sektor yang ada. Sebagai contoh *Moodle* dapat diimplementasikan untuk kelas perawat menurut artikel[5]. Dunia medis juga dapat menggunakan *Moodle* menurut artikel[6]. Bahkan untuk Sekolah Dasar pun dapat diimplementasikan[7].

Namun sebuah teknologi pastinya memiliki kelebihan serta kekurangan. Terdapat beberapa kelemahan yang dapat ditemui di dalam sistem *Moodle* seperti: antarmuka yang tidak mudah bagi orang lain untuk kuasai khususnya pengajar yang cukup berumur. Hal ini dapat dijumpai ketika seluruh dunia mengalami pandemi dan melakukan migrasi besar-besaran ke platform daring agar pendidikan tetap terlaksana[8]–[10]. Para pengguna pemula yang datang baik dari kalangan peserta didik maupun para pengajar dari berbagai jurusan harus berhadapan dengan ekosistem aplikasi *Moodle* ini. Sehingga banyak yang mengalami kebingungan apa yang harus dilakukan dengan platform tersebut. Hal ini terjadi karena kurangnya literasi digital yang dimiliki oleh pengguna baru tersebut. Karena masalah tersebut menjadi penurunan motivasi bagi para pengajar untuk melakukan migrasi ke platform digital[11].

Untuk menangani masalah tersebut, para peneliti telah mendesain berbagai macam aplikasi tambahan pendukung yang berguna untuk membantu para pengguna. Salah satu cara yang dapat digunakan untuk menyelesaikan masalah ini yaitu melalui teknologi *Internet of Things*. Teknologi *Internet of Things* mengizinkan komunikasi baik dari tingkat perangkat ke perangkat (*Machine-to-Machine communication*) maupun ke perangkat lunak[12], [13]. Pada tahun 2019, peneliti mengusulkan sebuah contoh implementasi *Internet of Things* untuk meningkatkan kemudahan dalam melakukan eksperimen lab[14]. Namun perkembangan ini tidak berhenti begitu saja. Dengan peneliti yang masih sama, kemudian alat dikembangkan

dengan menambahkan fitur penilaian otomatis[15] selama proses lab berlangsung. Di tahun berikutnya di tahun 2020, penelitian dilanjutkan dengan menghubungkan antarmuka dari aplikasi lain seperti LINE ke platform *Moodle* sehingga memungkinkan untuk mengakses segala materi yang ada di platform tersebut[16].

Namun terdapat beberapa kelemahan yang dimiliki oleh penelitian-penelitian sebelumnya seperti: penelitian yang dilakukan di tahun 2019 hanya menjangkau sesi praktik lab dan proses penilaian saja. Sedangkan, untuk penelitian tahun 2020 memiliki masalah dengan platform aplikasi yang digunakan. Aplikasi *LINE* sempat populer di kalangan anak muda di Indonesia, namun tidak bertahan lama karena tren *WhatsApp*. Berdasarkan artikel[17], penggunaan *LINE* hanya mencapai 9% jika dibandingkan *WhatsApp* yang mencapai 55%.

Untuk menyelesaikan masalah tersebut, penelitian ini memiliki tujuan untuk mendesain sebuah mode purwarupa berbasis *Internet of Things* yang dilengkapi dengan teknologi *Telegram Bot* sebagai asisten kelas interaktif yang dapat membantu dalam melaksanakan pembelajaran. Penggunaan *Telegram Bot* sendiri digunakan karena popularitas saat ini yang mendekati *WhatsApp* dan memiliki fitur yang unik jika dibandingkan dengan *WhatsApp*.

## 2. TINJAUAN PUSTAKA

Dalam artikel ini, peneliti menggunakan teori-teori ataupun konsep sebagai pondasi dalam pembuatan penelitian maupun artikel publikasi ini. Berikut ini adalah teori-teori yang digunakan dalam penelitian ini:

### 2.1. Moodle

Sebuah platform pembelajaran digital sumber terbuka yang dapat dipasang ke peladen pribadi secara gratis untuk membantu proses belajar mengajar semua tingkat sekolah hingga perguruan tinggi. *Moodle* menyediakan fitur-fitur yang sangat membantu proses pembelajaran yang sangat interaktif, sehingga siswa maupun mahasiswa tidak merasa bosan dengan pembelajaran luring[2], [4], [18].

### 2.2. Internet of Things

Teknologi *Internet of Things* merupakan sebuah teknologi komunikasi yang mengizinkan perangkat-perangkat modern

seperti peralatan elektronika rumah tangga hingga kendaraan bermotor untuk ditanami kemampuan memproses dan berkomunikasi melalui jaringan Internet. Sehingga dapat meningkatkan kehidupan manusia menjadi lebih layak[12], [13], [19].

### 2.3. Telegram Bot

Sebuah fitur layanan yang dihadirkan oleh aplikasi berpesan instan *Telegram* yang menawarkan sebuah robot yang dapat diprogram dan digunakan oleh pengguna untuk berbagai macam kebutuhan. Mulai dari hal sederhana seperti pengingat jadwal hingga hal kompleks seperti pencarian buku maupun artikel pembelajaran[20].

### 2.4. Web Service

Sebuah teknologi yang dapat menghadirkan komunikasi dua arah di antara perangkat homogen maupun heterogen seperti perangkat *Internet of Things* ke basis data dari melalui permintaan *Representational State Transfer (ReST)* maupun melalui *Simple Object Access Protocol (SOAP)*. Tergantung dari keadaan pita lebar yang ada, teknologi ReST sangat direkomendasikan untuk koneksi ke Internet dengan pita lebar yang rendah[21], [22].

### 2.5. Javascript Object Notation

Teknik penyimpanan data ini juga sering disebut dengan istilah *JSON* memiliki kelebihan unik berupa tingkat kebacaaan yang tinggi untuk manusia maupun mesin. *JSON* menghadirkan format *dictionary* atau juga lebih dikenal sebagai *key-value* dalam penyimpanan datanya sehingga lebih mudah untuk diproses melalui pemrograman. Format ini sangat sering digunakan oleh *Web Service* sebagai respon dari suatu permintaan[23], [24].

## 3. METODE PENELITIAN

Dalam melakukan penelitian ini, peneliti menggunakan metode eksperimen dalam mendesain purwarupa asisten kelas interaktif *Moodle*.

### 3.1. Tahap Pengumpulan Data

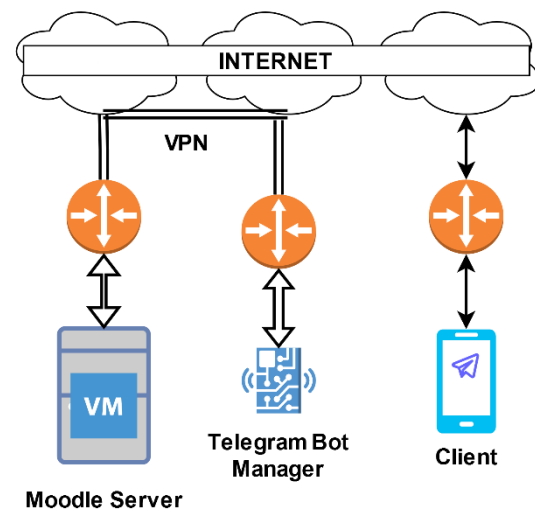
Peneliti mengumpulkan data-data dengan menggunakan observasi untuk mengetahui bagaimana cara kerja *Moodle* sendiri. Selain itu, peneliti juga melihat dokumentasi-dokumentasi yang disediakan secara bebas oleh *Moodle*

sendiri khususnya untuk bagian akses *Moodle* melalui antarmuka *web service*.

### 3.2. Tahap Desain Purwarupa

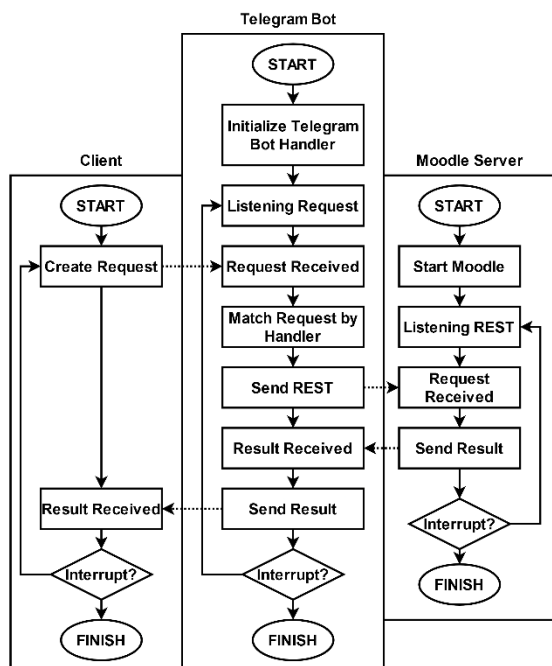
Setelah mendapatkan data-data berkaitan dengan *Moodle*, peneliti melanjutkan proses dengan mendesain dari purwarupa asisten kelas interaktif dengan menggunakan perangkat *Internet of Things*. Dalam kasus ini peneliti menggunakan papan pemrosesan alternatif dari *Raspberry Pi* dengan nama *Orange Pi Zero 2*.

Perangkat ini nantinya akan ditanami kode untuk *Telegram Bot* berinteraksi dengan pengguna dan peladen *Moodle* secara daring melalui Internet. Berikut ini adalah bentuk topologi dan aliran pemrosesan dari model purwarupa:



Gbr 1. Topologi jaringan model purwarupa

Dalam topologi yang ditunjukkan di Gambar 1, memperlihatkan komunikasi antar perangkat melalui jaringan Internet. Namun khusus untuk komunikasi antara *Moodle Server* dengan *Telegram Bot Manager* ditambah dengan konektivitas khusus melalui *Virtual Private Network* karena tidak memiliki IP publik statis. Jaringan VPN ini menggunakan penyedia pihak ketiga, sehingga semua perangkat yang terhubung tidak perlu memerlukan IP statis dan cukup menggunakan jaringan Internet yang ada. Untuk alur proses dari model sendiri ditunjukkan dengan gambar berikut:



Gbr 2. Alur proses model yang diusulkan.

Alur proses yang ditunjukkan di Gambar 2 merupakan ilustrasi proses dari model yang diusulkan. Di mulai dari perangkat *Internet of Things* yang melakukan inisialisasi *Bot* beserta *Handler* fungsinya sehingga dapat memahami perintah dari klien. Setelah proses inisialisasi selesai di purwarupa model, dilanjutkan dengan masuk ke mode *Listening* untuk menunggu permintaan yang masuk dari klien. Di sisi klien dapat memulai aplikasi *Telegram* dan mengirimkan permintaan ke aplikasi tersebut. Secara otomatis permintaan akan dikirimkan ke *Telegram Bot Manager*. Permintaan ini akan dipetakan sesuai jenis permintaannya lalu diteruskan ke *Moodle Server*. Server akan menerima permintaan ini dan membalas kembali ke *Telegram Bot Manager* dan dikembalikan ke klien. Proses ini akan terus menerus berulang hingga klien tidak mengirimkan permintaan lagi ataupun karena gangguan eksternal seperti listrik dan koneksi Internet yang buruk.

### 3.3. Tahap Implementasi dan Pengujian

Purwarupa yang sudah didesain, kemudian diimplementasikan ke dalam perangkat *Internet of Things* dengan spesifikasi sebagai berikut:

1. 4 Inti Prosesor *Allwinner H616*
2. Memori *SDRAM 1GB*
3. 32 GB *MicroSD*
4. Nirkabel *IEEE 802.11 a/b/g/n/ac*
5. 1000M/100M/10M *Ethernet*

Kemudian untuk kebutuhan *Moodle Server* sendiri, peneliti menggunakan *Virtual Private Server* (VPS) lokal di dalam komputer. Namun akses via Internet daripada VPS ini didapatkan melalui jaringan khusus *Virtual Private Network* agar bisa diakses melalui Internet tanpa harus memiliki pengalaman statis publik. VPS ini memiliki spesifikasi berupa:

1. 2 inti *vCPU*
2. 2GB memori internal
3. Penyimpanan 20GB
4. Sistem Operasi *Debian 12 Bookworm*
5. *Apache2* dan *MariaDB*
6. *Moodle*

Sesudah semua kebutuhan purwarupa sudah diimplementasikan ke masing-masing perangkat, maka langkah selanjutnya adalah pengujian. Peneliti menggunakan dua pengujian untuk menguji kemampuan *Telegram Bot* dalam menerima dan mengirimkan permintaan. Serta kemampuan perangkat *Internet of Things* dalam mengirimkan banyak permintaan ke server. Dengan menggunakan perangkat lunak *Locust*, peneliti dapat menguji kemampuan pelayanan dari purwarupa sendiri. Untuk skenario pengujian yang akan dilakukan dengan menggunakan *Locust* dimulai dengan menyiapkan target pengaksesan. Berikut ini adalah skenario konfigurasi permintaan yang digunakan untuk pengujian:

Tabel 1. Konfigurasi Permintaan *Locust*

No	Permintaan	Jumlah	Jenis
1	Akses <i>Home (/)</i>	1	GET
2	Permintaan <i>Token (login.php)</i>	1	POST
3	Akses Fungsional ( <i>server.php</i> )	17	POST

Dalam pembuatan skenario di Tabel 1, peneliti menentukan satu (1) kali akses ke halaman *Home*, satu (1) kali akses *Token (Login)*, dan 17 kali akses fungsionalitas *Moodle*. Jumlah 17 untuk akses fungsionalitas ditentukan oleh jumlah mata kuliah beserta mahasiswa yang dapat dijelajahi oleh *Telegram Bot*. Sehingga untuk konfigurasi permintaan ini terdapat total 19 permintaan untuk satu pengguna. Namun dalam pengujiannya, peneliti akan menggunakan 1000 pengguna sebagai simulasi ke *Moodle Server*. Untuk durasinya sendiri, peneliti menjalankan skenario selama satu (1) jam. Dengan skenario ini, peneliti dapat

mengetahui kemampuan model purwarupa beserta *Moodle Server* dalam memenuhi permintaan pengguna.

#### 4. HASIL DAN PEMBAHASAN

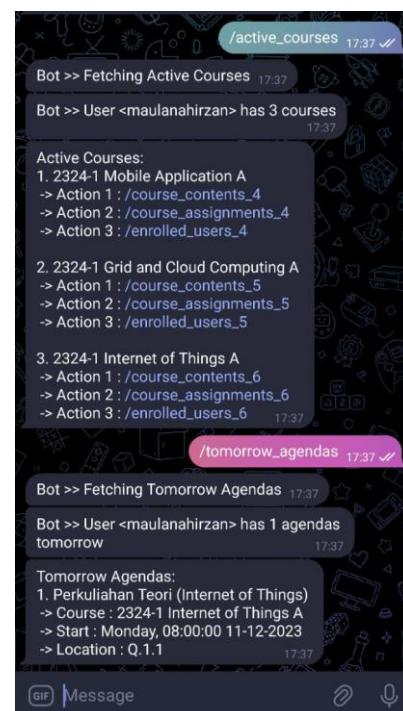
Dari hasil pengujian yang telah dilakukan, peneliti berhasil mengirimkan permintaan dari aplikasi *Telegram* ke *Telegram Bot* yang kemudian dilanjutkan ke *Moodle Server*. Pesan ini kemudian dijawab oleh *Moodle Server* sesuai dengan permintaan yang diberikan oleh klien. Daftar perintah yang tersedia saat ini dalam table.

Tabel 2. Daftar Perintah *Telegram Bot*

No	Nama Perintah	Deskripsi
1	<code>/start</code>	Menampilkan semua perintah
2	<code>/login</code>	Mendapatkan <i>Token</i>
3	<code>/userinfo</code>	Informasi pengguna
4	<code>/active_courses</code>	Daftar mata kuliah
5	<code>/week_agendas</code>	Jadwal satu minggu ke depan
6	<code>/tomorrow_agendas</code>	Jadwal besok
7	<code>/course_contents</code>	Daftar isi topik mata kuliah
8	<code>/course_assignments</code>	Daftar tugas mata kuliah
9	<code>/enrolled_users</code>	Daftar peserta mata kuliah
10	<code>/get_grades</code>	Daftar nilai mata kuliah per peserta

Daftar perintah yang ditampilkan di Tabel 2 merupakan perintah-perintah yang dapat digunakan dalam *Telegram Bot*. Perintah yang dapat digunakan tanpa akses *token* adalah *start* untuk melihat menu-menu yang tersedia. Kemudian juga terdapat menu *login* untuk masuk ke dalam system dan mendapatkan *token* semi permanen yang dapat tersimpan selama *bot* tidak dinyalakan ulang. Perintah selanjutnya adalah *userinfo* untuk mendapatkan informasi pengguna yang sudah masuk ke dalam system. Sedangkan menu-menu yang berkaitan dengan *Moodle* berupa *active\_courses* untuk melihat semua mata kuliah aktif yang pengguna ajar. Perintah *week\_agendas* untuk melihat jadwal

selama satu minggu ke depan baik berupa teori maupun praktikum (dengan ketentuan pengguna sudah menjadwalkan pengajaran melalui sistem kalender *Moodle*). Perintah *tomorrow\_agendas* memperlihatkan jadwal perkuliahan besok hari. Selain perintah-perintah tersebut, terdapat perintah tambahan yang hanya dapat diakses ketika pengguna sudah menggunakan perintah *active\_courses* berupa *course\_contents* untuk melihat semua topik pembelajaran dan semua materi yang ada di mata kuliah tersebut. Perintah lain seperti *course\_assignments* akan memperlihatkan semua tugas yang ada di mata kuliah tersebut. Perintah *enrolled\_users* menampilkan semua pengguna yang ada di mata kuliah tersebut baik berupa mahasiswa maupun dosen. Sedangkan perintah *get\_grades* akan memperlihatkan semua nilai tugas yang dimiliki oleh pengguna. Berikut ini adalah contoh salah satu tampilan hasil dari perintah yang sudah dijelaskan sebelumnya.



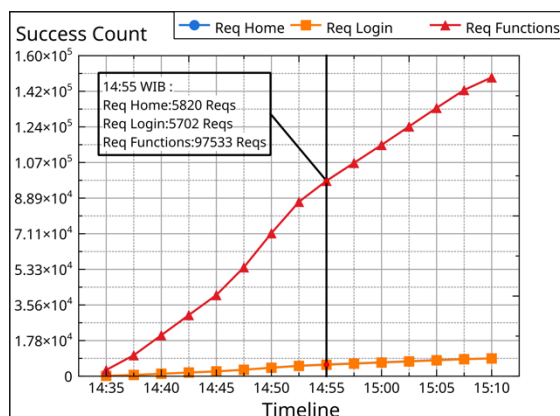
Gbr 3. Hasil pengambilan layar *Telegram*

Gambar 3 merupakan hasil tangkapan layar aplikasi *Telegram* yang memperlihatkan hasil permintaan untuk perintah *active\_courses* sehingga menampilkan mata kuliah yang diajarkan oleh pengguna. Selain menampilkan mata kuliah yang diajarkan, *Telegram Bot* juga



menampilkan aksi tambahan yang dapat dilakukan dengan masing-masing mata kuliah. Sebagai contoh, pengguna dapat melihat isi mata kuliah dengan menekan perintah yang sudah disediakan. Selain itu juga terdapat menu tambahan untuk melihat daftar tugas dan pengguna. Hasil tangkapan layar ini cukup menjadi bukti interaktif asisten kelas yang dibuat.

Pengujian berikutnya adalah berkaitan dengan seberapa banyak model *Telegram Bot* dapat mengirimkan banyak permintaan ke *Moodle Server* dan seberapa banyak layanan yang dapat dipenuhi oleh *server* tersebut. Selama durasi satu jam, peneliti berhasil mencatat lebih dari 14.000 ribu baris data catatan kinerja *Locust* dalam format *comma separated value* (csv). Namun karena data ini masih bersifat kotor, maka peneliti membersihkan data dengan membuang baris data dengan jumlah user kosong (0). Sehingga didapatkanlah total data 12.276 baris data. Karena dianggap masih terlalu banyak untuk ditampilkan. Maka peneliti melakukan teknik *Time-based Averaging* dengan meninggikan frekuensi data dari mili detik (ms) menjadi tiap lima (5) menit. Dengan teknik ini, peneliti berhasil menurunkan jumlah baris data menjadi 15 baris. Sehingga didapatkan hasil berupa jumlah permintaan yang berhasil diilustrasikan dengan grafik berikut:

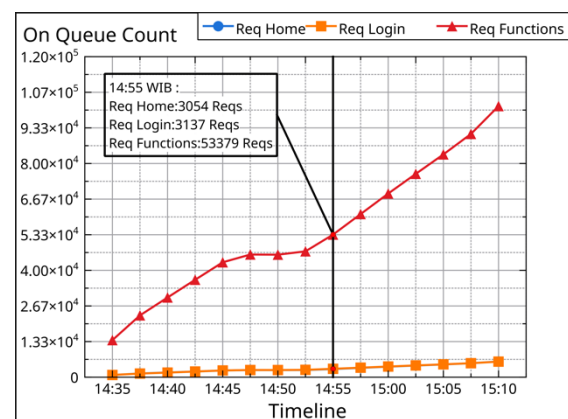


Gbr 4. Grafik akumulasi permintaan sukses per satuan waktu

Berdasarkan Gambar 4 menampilkan akumulasi permintaan sukses dari ketiga jenis permintaan. Sebagai sampel hasil di waktu 14:55 WIB, model *Internet of Things* mencatat akumulasi permintaan untuk akses *Home*

sebanyak 5.820 permintaan. Untuk akumulasi permintaan *Token (Login)* mencapai 5.702 permintaan, dan permintaan ke segala fungsi *Moodle* mencapai 97.533 permintaan. Namun jumlah akumulasi ini masih terus meningkat sesuai dengan berjalannya waktu. Rata-rata akumulasi keseluruhan dari permintaan yang sukses dilakukan mencapai 117.628 permintaan.

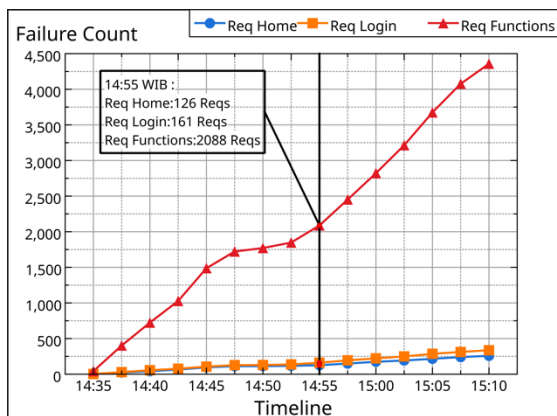
Karena perangkat tidak memiliki kapasitas yang cukup untuk memenuhi kebutuhan pengguna, maka tentu saja ada permintaan-permintaan yang masih tertunda atau dalam antrian (*On Queue*). Berikut ini adalah hasil akumulasi jumlah permintaan dalam antrian:



Gbr 5. Grafik akumulasi permintaan dalam antrian per satuan waktu

Antrian merupakan sesuatu hal yang terjadi dalam suatu sistem, maka model yang dibuat peneliti pun tidak luput dari kejadian ini. Berdasarkan Gambar 5, terjadi peningkatan permintaan dalam antrian. Sebagai sampel pada pukul 14:55 WIB terdapat antrian untuk permintaan *Home* mencapai 3054 permintaan. Untuk permintaan *Token (Login)* mencapai 3137 permintaan, dan permintaan Fungsional mencapai 53.379. Rata-rata akumulasi keseluruhan dari awal hingga akhir adalah 75.526 permintaan dalam antrian. Sama seperti grafik sebelumnya. Potensi jumlah antrian yang terjadi dapat meningkat seiring waktu.

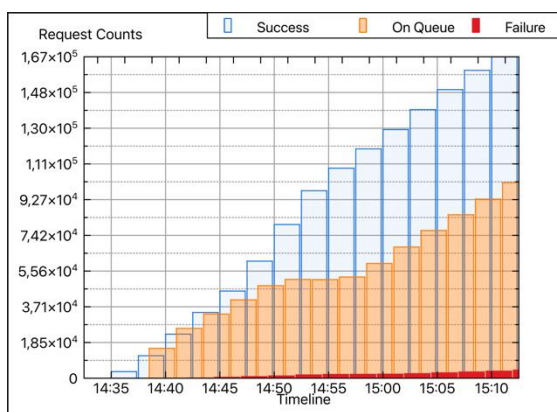
Sedangkan untuk penilaian kinerja berikutnya adalah akumulasi kegagalan permintaan ditampilkan dengan gambar berikut:



Gbr 6. Grafik akumulasi kegagalan permintaan per satuan waktu

Gambar 6 adalah hasil data yang ditemukan oleh peneliti ketika melakukan pengujian. Grafik yang ditampilkan ini berupa akumulasi kegagalan yang dialami oleh model ketika mengirimkan permintaan ke Moodle Server. Menurut sampel di grafik pada jam 14:55 WIB, akses Home mencapai kegagalan mencapai 126 permintaan. Untuk akses permintaan Token (Login), mengalami kegagalan mencapai 161 permintaan. Sedangkan untuk akses ke fungsionalitas Moodle mencapai 2.088 permintaan. Rata-rata akumulasi keseluruhan permintaan gagal mencapai 3.175 permintaan. Jika dilihat dari grafik keseluruhan, terdapat potensi kenaikan kegagalan permintaan.

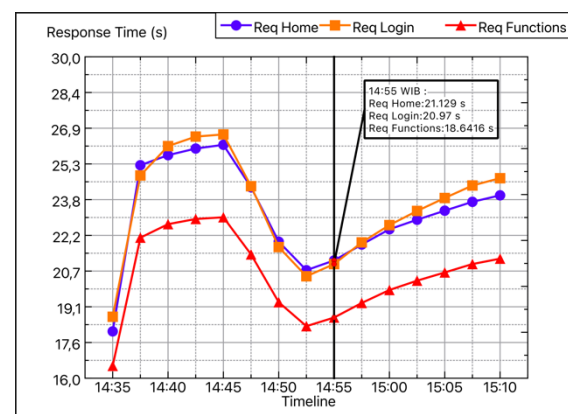
Data-data ini kemudian dikumpulkan menjadi satu dan dipisahkan berdasarkan kategori seperti: Sukses (*Success*), Antrian (*On Queue*), dan Gagal (*Failure*)



Gbr 7. Hasil agregasi seluruh permintaan dan dikategorikan berdasarkan status.

Gambar 7 merupakan hasil agregasi dari keseluruhan permintaan baik berupa akses ke Home, permintaan Token (Login), dan permintaan fungsionalitas Moodle. Agregasi ini dilakukan untuk memperlihatkan angka permintaan per kategori yang sudah ditentukan. Berdasarkan gambar tersebut, jumlah permintaan yang telah sukses (*Success*) dipenuhi oleh Moodle Server lebih tinggi dibandingkan permintaan dalam antrian (*On Queue*) dan gagal (*Failure*).

Hasil berikutnya adalah kecepatan respon terhadap permintaan. Berikut ini adalah hasil akumulasi waktu respon yang didapatkan dan diilustrasikan ke dalam bentuk gambar:



Gbr 8. Grafik akumulasi waktu respon permintaan per satuan waktu

Gambar 8 merupakan hasil pencatatan hasil respon terhadap permintaan yang dilakukan oleh model. Karena sudah bersifat *round trip*, maka peneliti tidak perlu lagi melakukan kalkulasi tambahan. Berdasarkan sampel di grafik, waktu yang dibutuhkan untuk melakukan permintaan ke Home mencapai 21,129 detik. Untuk permintaan Token (Login) sendiri membutuhkan waktu 20,97 detik. Sedangkan untuk permintaan fungsional mencapai 18,6416 detik. Rata-rata akumulasi total waktu respon keseluruhan mencapai 66,86 detik. Berbeda dengan grafik-grafik sebelumnya yang mengalami kenaikan. Grafik khusus waktu respon ini memiliki bentuk tersendi yang di mana sempat mengalami kenaikan karena *load* atau beban yang meningkat. Kemudian waktu respon menjadi lebih cepat dengan munculnya grafik menurun. Namun grafik mulai naik perlahan sesuai

dengan naiknya lalu lintas permintaan yang dibuat oleh model.

Setelah menjelaskan hasil yang didapatkan, peneliti kemudian melanjutkan ke bagian diskusi di mana peneliti akan membahas tiap detail yang peneliti temukan. Dimulai dari implementasi perintah dari *Telegram Bot*. Perintah-perintah yang dibuat untuk *Telegram Bot* ini didesain menurut ketersediaan *Application Programming Interface* (API) yang disediakan oleh *Moodle* sendiri. *Moodle* menyediakan banyak sekali API yang dapat digunakan oleh peneliti. Tetapi, sebagian besar API yang dapat digunakan bersifat administratif untuk administrator dan bukan ke manajemen mata kuliah yang dapat digunakan oleh pengajar. Sehingga perintah yang dapat disediakan oleh peneliti menjadi terbatas. Pembahasan berikutnya adalah mengenai konfigurasi dari pengujian yang dilakukan peneliti. Khusus untuk pengujian fungsional yang terdiri dari 17 fungsi. Angka ini didapatkan dengan mengumpulkan semua kemungkinan perintah-perintah yang dapat diperintahkan oleh *Telegram*. Jumlah perintah-perintah ini dapat meningkat sesuai dengan jumlah dari mata kuliah yang diajarkan dan jumlah pengguna (khususnya) mata kuliah yang mengikuti satu mata kuliah tersebut. Pembahasan ketiga mengenai *benchmark* yang dilakukan peneliti untuk mengetahui kemampuan dalam mengirimkan permintaan dari model purwarupa ke *Moodle Server*. Berdasarkan hasil yang ditemui, baik akumulasi permintaan sukses dan gagal sama-sama meningkat. Sedangkan untuk waktu sendiri, mengalami naik turun. Hal ini dipengaruhi faktor-faktor eksternal seperti: kemampuan model purwarupa dalam mengirimkan permintaan, kemampuan *Moodle Server* dalam menerima permintaan, dan konektivitas di antara keduanya. Model yang digunakan menggunakan perangkat keras dari *Orange Pi* yang memiliki kemampuan yang cukup untuk komputasi biasa, namun tidak cukup apabila dibandingkan dengan komputasi *server*. Sehingga sangat wajar apabila model tidak dapat melayani banyak pengguna. Sedangkan untuk faktor dari *Moodle Server*, disebabkan oleh kurangnya sumber daya yang dimiliki oleh peneliti. Sehingga peneliti menggunakan mesin virtual dengan komputasi yang terbatas dan tidak menggunakan *server* asli. Meskipun

terbatas, namun masih bisa melayani permintaan 1000 pengguna dengan cukup baik. Faktor terakhir adalah konektivitas yang digunakan untuk komunikasi model. Karena peneliti tidak memiliki akses ke IP statis publik berbayar, maka peneliti menggunakan alternatif lain yaitu sebuah *Virtual Private Network* yang diatur oleh pihak ketiga secara gratis. Sehingga kedua perangkat (model dan *Moodle Server*) dapat saling berkomunikasi satu sama lain tanpa memerlukan IP publik statis. Kecepatan yang dihadirkan oleh penyedia VPN ini terbatas karena bersifat publik dan tidak berbayar. Maka secara otomatis pita lebar yang disediakan oleh mereka juga terbatas. Pembahasan terakhir adalah total akumulasi dari masing-masing *benchmark*. Sehingga hasil dari satu waktu diakumulasikan ke hasil berikutnya. Berdasarkan hasil pengujian dan diskusi, peneliti bisa mengambil kesimpulan bahwa model yang diusulkan mampu menjadi asisten kelas interaktif *Moodle* dengan jumlah yang banyak tapi tetap terbatas. Namun dapat ditingkatkan dari segi perangkat keras.

## 5. KESIMPULAN

- Dunia pendidikan adalah hal yang sangat penting untuk dipunyai setiap manusia yang ada. Namun untuk mencapainya terkadang mengalami kendala seperti pandemi yang telah terjadi di tahun sebelumnya
- Banyak pengajar dari berbagai tingkat yang mengalami kendala dikarenakan oleh migrasi besar-besaran dari pengajaran luring menjadi daring. Sehingga banyak dari mereka yang memiliki literasi digital yang minim untuk dapat memahami sistem pembelajaran daring.
- Peneliti mengusulkan sebuah model purwarupa berbasis *Internet of Things* dan teknologi *Telegram Bot* sebagai asisten kelas interaktif untuk membantu para pengajar (khususnya senior) dalam mengakses *Moodle*
- Dari hasil pengujian yang dilakukan. Peneliti berhasil mendesain model, dan melakukan permintaan melalui aplikasi *Telegram*.
- Untuk performanya sendiri, model mampu melayani 1000 pengguna dengan cukup baik. Dari hasil rata-rata total akumulasi keseluruhan yang didapatkan, permintaan



sukses mencapai 117.628; permintaan dalam proses antrian mencapai 75.526; permintaan gagal mencapai 3.175 permintaan; dan waktu respon membutuhkan waktu 67 detik.

## UCAPAN TERIMA KASIH

Penulis mengucapkan terima kasih kepada Lembaga Penelitian dan Pengabdian kepada Masyarakat atas dukungan dalam bentuk material maupun dukungan moral dalam melaksanakan penelitian ini

## DAFTAR PUSTAKA

- [1] A. Aldiab, H. Chowdhury, A. Kootsookos, F. Alam, and H. Allhibi, "Utilization of Learning Management Systems (LMSs) in higher education system: A case review for Saudi Arabia," *Energy Procedia*, vol. 160, pp. 731–737, 2019, doi: <https://doi.org/10.1016/j.egypro.2019.02.186>.
- [2] S. H. P. W. Gamage, J. R. Ayres, and M. B. Behrend, "A systematic review on trends in using Moodle for teaching and learning," *Int. J. STEM Educ.*, vol. 9, no. 1, p. 9, Jan. 2022, doi: 10.1186/s40594-021-00323-x.
- [3] M. Gachkova and E. Somova, "MOODLE PLUG-INS FOR DESIGN AND DEVELOPMENT OF GAMIFIED COURSES," in *INTED2020 Proceedings*, in 14th International Technology, Education and Development Conference. Valencia, Spain: IATED, Mar. 2020, pp. 2187–2195. doi: 10.21125/inted.2020.0676.
- [4] H. Athaya, R. D. A. Nadir, D. Indra Sensuse, K. Kautsarina, and R. R. Suryono, "Moodle Implementation for E-Learning: A Systematic Review," in *Proceedings of the 6th International Conference on Sustainable Information Engineering and Technology*, in SIET '21. New York, NY, USA: Association for Computing Machinery, 2021, pp. 106–112. doi: 10.1145/3479645.3479646.
- [5] C.-J. Chen, H.-J. Tsai, M.-Y. Lee, Y.-C. Chen, and S.-M. Huang, "Effects of a Moodle-based E-learning environment on E-collaborative learning, perceived satisfaction, and study achievement among nursing students: A cross-sectional study," *Nurse Educ. Today*, vol. 130, p. 105921, 2023, doi: <https://doi.org/10.1016/j.nedt.2023.105921>.
- [6] Jeong Seri Hwang Hyunyong, "Do we need Moodle in medical education? A review of its impact and utility," *KMJ*, vol. 38, no. 3, pp. 159–168, 2023, doi: 10.7180/kmj.23.139.
- [7] A. Goltsiou and C. Sofianopoulou, "Moodle In The Primary School – Investigation Of Resources And Implementation Possibilities," in *ICERI2019 Proceedings*, in 12th annual International Conference of Education, Research and Innovation. Seville, Spain: IATED, Nov. 2019, pp. 9933–9941. doi: 10.21125/iceri.2019.2434.
- [8] R. QUANSAH and C. ESSIAM, "The use of learning management system (LMS) moodle in the midst of covid-19 pandemic: Students' perspective," *J. Educ. Technol. Online Learn.*, vol. 4, no. 3, pp. 418–431, Sep. 2021, doi: 10.31681/jetol.934730.
- [9] Maria Borodina, Tatiana Ivashkina, Tatyana Golubeva, Oleg Afanasiev, Yulia Pronina, and Kirill Berlov, "Changes in the use of the moodle platform by students at different levels of training depending on the period of restrictions due to Covid-19," *Rev. Conrado*, vol. 18, no. 88, Oct. 2022, Accessed: Dec. 09, 2023. [Online]. Available: <https://conrado.ucf.edu.cu/index.php/conrado/article/view/2571>
- [10] A. Taamneh, A. Alsaad, H. Elrehail, M. Al-Okaily, A. Lutfi, and R. P. Sergio, "University lecturers acceptance of moodle platform in the context of the COVID-19 pandemic," *Glob. Knowl. Mem. Commun.*, vol. 72, no. 6/7, pp. 666–684, Jan. 2023, doi: 10.1108/GKMC-05-2021-0087.
- [11] T. Aikina and L. Bolsunovskaya, "Moodle-Based Learning: Motivating and Demotivating Factors," *Int. J. Emerg. Technol. Learn. IJET*, vol. 15, no. 2, pp. 239–248, Jan. 2020.
- [12] S. Kumar, P. Tiwari, and M. Zymbler, "Internet of Things is a revolutionary approach for future technology enhancement: a review," *J. Big Data*, vol. 6, no. 1, p. 111, Dec. 2019, doi: 10.1186/s40537-019-0268-2.

- [13] A. Khanna and S. Kaur, "Internet of Things (IoT), Applications and Challenges: A Comprehensive Review," *Wirel. Pers. Commun.*, vol. 114, no. 2, pp. 1687–1762, Sep. 2020, doi: 10.1007/s11277-020-07446-4.
- [14] K. Mershad and A. Hamieh, "Using Internet of Things to Enhance Remote Experiments in Learning Management Systems," in *2019 IEEE International Smart Cities Conference (ISC2)*, Oct. 2019, pp. 458–464. doi: 10.1109/ISC246665.2019.9071786.
- [15] K. Mershad, A. Damaj, and A. Hamieh, "Using Internet of Things for Automatic Student Assessment during Laboratory Experiments," in *2019 IEEE International Smart Cities Conference (ISC2)*, Oct. 2019, pp. 317–323. doi: 10.1109/ISC246665.2019.9071654.
- [16] T. Kita, C. Nagaoka, N. Hiraoka, and T. Molnár, "Development of a Moodle UI Using LINE Chat for Casual Learning as a Part of a Learner Assistive LMS," in *2020 IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE)*, Dec. 2020, pp. 927–929. doi: 10.1109/TALE48869.2020.9368321.
- [17] S. Susanto, E. Muafiah, A. Desrani, A. W. Ritonga, and A. R. Hakim, "Trends of Educational Technology (EdTech): Students' Perceptions of Technology to Improve the Quality of Islamic Higher Education in Indonesia," *Int. J. Learn. Teach. Educ. Res.*, vol. 21, no. 6, pp. 226–246, 2022, doi: <https://doi.org/10.26803/ijlter.21.6.14>.
- [18] A. Shoufan, "Lecture-Free Classroom: Fully Active Learning on Moodle," *IEEE Trans. Educ.*, vol. 63, no. 4, pp. 314–321, Nov. 2020, doi: 10.1109/TE.2020.2989921.
- [19] R. Sudarmani, K. Venusamy, S. Sivaraman, P. Jayaraman, K. Suriyan, and M. Alagarsamy, "Machine to machine communication enabled internet of things: a review," *Int. J. Reconfigurable Embed. Syst.*, vol. 11, no. 2, p. 126, 2022, doi: <http://doi.org/10.11591/ijres.v11.i2.pp126-134>.
- [20] Laiby Thomas and Subramanya Bhat, "A Comprehensive Overview of Telegram Services - A Case Study," *Int. J. Case Stud. Bus. IT Educ. IJCSBE*, vol. 6, no. 1, pp. 288–301, May 2022, doi: 10.47992/IJCSBE.2581.6942.0165.
- [21] I. Ahmad, E. Suwarni, R. I. Borman, Asmawati, F. Rossi, and Y. Jusman, "Implementation of RESTful API Web Services Architecture in Takeaway Application Development," in *2021 1st International Conference on Electronic and Electrical Engineering and Intelligent System (ICE3IS)*, Oct. 2021, pp. 132–137. doi: 10.1109/ICE3IS54102.2021.9649679.
- [22] A. Ehsan, M. A. M. E. Abuhaliqa, C. Catal, and D. Mishra, "RESTful API Testing Methodologies: Rationale, Challenges, and Solution Directions," *Appl. Sci.*, vol. 12, no. 9, 2022, doi: 10.3390/app12094369.
- [23] G. P. Tiwary, E. Stroulia, and A. Srivastava, "Compression of XML and JSON API Responses," *IEEE Access*, vol. 9, pp. 57426–57439, 2021, doi: 10.1109/ACCESS.2021.3073041.
- [24] J. C. Viotti and M. Kinderkhedia, "A Survey of JSON-compatible Binary Serialization Specifications." 2022.