

# Aplikasi Sistem *Monitoring Server* Menggunakan *Device Orange Pi* Berbasis *Web Service* Studi Kasus PT. MNC Televisi Indonesia – MNC Group

Cepi Muh. Usman<sup>1</sup>, Muhamad Rusli<sup>2</sup>, Muhamad Femy Mulya<sup>3</sup>, Tri Wahyu Widyaningsih<sup>4</sup>

<sup>1,2,4</sup>Program Studi Teknik Informatika & <sup>3</sup>Program Studi Sistem Informasi, Tanri Abeng University, Jakarta

<sup>1</sup>cepi@student.tau.ac.id, <sup>2</sup>m.rusli@student.tau.ac.id, <sup>3</sup>femy.mulya@tau.ac.id, <sup>4</sup>tri.widyaningsih@tau.ac.id

Diterima : 20 Februari 2022

Disetujui 27 Maret 2022

**Abstrak**— *Server* merupakan salah satu komponen utama dari sistem jaringan komputer yang memiliki fungsi untuk memberikan suatu *service* terhadap penggunanya. Setiap aktifitas dan operasional pelayanan suatu *server* terhadap *client* dalam penerapannya terdiri dari beberapa jenis proses untuk memenuhi segala permintaan atau *request client* yang dikirimkan oleh *server*. Oleh sebab itu, diperlukan suatu aplikasi sistem monitoring yang dapat melakukan pemantauan segala macam aktifitas di dalam *server*, dengan demikian *administrator (operator)* dapat memantau dan memberikan peringatan apabila terjadi permasalahan pada *server* yang digunakan. Pada aplikasi ini dikembangkan dengan menggunakan metode *scrum* dan perancangan sistem aplikasi dengan menggunakan *Unified Modeling Language (UML)*, serta bahasa pemrograman berbasis *Java interface* dan *PHP code*. Penelitian ini menghasilkan aplikasi *web* yang mampu melakukan pemantauan dan pengumpulan informasi dari seluruh perangkat *server* yang berada di PT. MNC Televisi Indonesia - MNC Group. Pada penelitian ini juga menggunakan protokol *TCP/IP (Transmission Control Protocol/Internet Protocol)* untuk melakukan pemantauan pada perangkat *server*. Aplikasi monitoring ini juga, mampu memberikan beberapa informasi terkait utilisasi memori (RAM), utilisasi CPU, utilisasi *storage* dan utilisasi jaringan pada *server*, serta pada aplikasi ini juga mampu mengirimkan pesan berupa peringatan dalam bentuk *email* kepada *administrator* saat terjadi kesalahan atau *malfunction* pada *server*, dengan demikian proses eskalasi penanganan *server* menjadi lebih cepat dilakukan. Lalu pada penelitian ini juga, menggunakan metode pengujian *black box* yang terdiri dari dua kategori umum pengujian, yaitu pengujian *black box* untuk *admin* dan pengujian *black box* untuk *user*, dengan beberapa parameter-parameter kelas uji.

**Keywords**— *Server, TCP/IP, PHP, Email*

## I. PENDAHULUAN

### A. Latar Belakang Masalah

Perkembangan teknologi saat ini semakin cepat dan pesat dari waktu ke waktu, khususnya teknologi informasi dan komunikasi. Teknologi merupakan salah satu bagian alat bantu yang sangat sering digunakan dalam aktifitas

pekerjaan sehari-hari, peran teknologi dalam melakukan pengolahan informasi menjadi sangat penting, karena dapat membantu menghasilkan informasi yang akurat dan valid bagi penggunanya. Dengan demikian, penulis merancang dan membangun sistem monitoring *server* dengan menggunakan *device orange Pi* berbasis *web service*, agar mempermudah

memonitoring *server* pada perusahaan PT. MNC Televisi Indonesia – MNC Group.

*Server* merupakan suatu sistem komputer besar yang terintegrasi dan tersusun pada suatu jaringan komputer besar yang menyediakan suatu layanan bagi para pengguna yang biasa disebut sebagai *client* [1]. Suatu *server* umumnya telah menjalankan banyak proses untuk membantu dalam memenuhi permintaan dari *client*. Oleh sebab itu, sering sekali *server* mengalami beberapa kendala yang disebabkan karena tidak adanya sumber daya (*resource*) yang mumpuni atau sesuai untuk memenuhi *resource* perbaikan *server*. Hal tersebut dapat menyebabkan layanan atau *service server* mati secara mendadak dikarenakan *file system kernel* memutuskan untuk mematikan layanan pada *server* yang membutuhkan *resource* yang sangat besar. *File system kernel* adalah komponen inti atau penting dari suatu sistem operasi. *Kernel* memiliki tanggung jawab untuk menyelesaikan tugas-tugas tingkat rendah/bawah seperti untuk manajemen *storage*, manajemen *CPU* dan manajemen memori.

Permasalahan ini dapat diatasi dengan cara membangun suatu sistem monitoring *server* yang dapat mengumpulkan beberapa informasi *server* seperti utilisasi *CPU*, utilisasi memori, utilisasi *storage* dan utilisasi *network*. Adapun kebutuhan yang dibutuhkan dalam membangun sistem monitoring ini adalah protokol *TCP/IP* (*Transmission Control Protocol/Internet Protocol*) yang merupakan protokol untuk mengolah dan memonitor perangkat jaringan. *TCP/IP* juga banyak digunakan untuk komunikasi data baik untuk jaringan internet maupun *Local Area Network* (LAN) atau biasa disebut intranet.

Penelitian ini bertujuan untuk membangun suatu aplikasi *monitoring server* berbasis *web service*, agar mendukung pemantauan *server* secara *real time*. Selain itu, pada aplikasi ini juga akan mengumpulkan data *log server* untuk beberapa parameter yang bersumber dari *CPU*, *storage*, memori dan *network* yang berasal dari *server*. Adapun persentase parameter *threshold*

pada monitoring *server* yang akan dibangun terlihat pada Tabel 1 sebagai berikut;

Tabel 1. Parameter *Threshold* pada Monitoring *Server*

| No. | Kategori Monitoring | Threshold | Triger Monitoring Server    |
|-----|---------------------|-----------|-----------------------------|
| 1.  | CPU                 | >95%      | Log Warning and Email Alert |
| 2.  | Memory              | <1GB      |                             |
| 3.  | Storage/Disk        | <5GB      |                             |
| 4.  | Network (Bandwidth) | <100kbps  |                             |

Aplikasi sistem monitoring *server* ini mampu melakukan otomatisasi pengaktifan layanan *server* menggunakan *java client* dengan *database* penampung sementara dengan *influxDB*. *Tools* yang digunakan dalam membangun sistem monitoring *server* ini antara lain, *apache*, *Java*, *influxDB*, *PHP*, *Mysql* dan *telegraf*.

#### B. Rumusan Masalah

Berdasarkan dari latar belakang yang telah dijelaskan sebelumnya, maka dapat dirumuskan beberapa permasalahan sebagai berikut:

1. Bagaimana cara menganalisa alur kerja pada proses monitoring *server* pada PT. MNC Televisi Indonesia – MNC Group?
2. Bagaimana cara mengidentifikasi dan melakukan monitoring terhadap *server* dengan *device* orange pi berbasis *web service* secara *realtime*?
3. Bagaimana cara membuat otomatisasi aplikasi monitoring *server* dengan bantuan *device* orange pi?
4. Bagaimana melengkapi aplikasi yang dibangun dengan fitur peringatan/*alert* melalui *email* jika ada *service* yang mati atau tidak bekerja dalam keadaan normal di perangkat-perangkat yang terhubung di jaringan tersebut.
5. Bagaimana membangun aplikasi monitoring *server* dengan fitur keamanan data?

#### C. Tujuan Penelitian

Berdasarkan rumusan masalah diatas, maka tujuan dari penelitian yang diusulkan adalah sebagai berikut :

1. Mengidentifikasi alur kerja pada proses monitoring *server* pada sebuah perusahaan PT. MNC Televisi Indonesia – MNC Group dengan melakukan wawancara dengan pihak staf *IT*.
2. Membangun sebuah sistem informasi berbasis komputerisasi dalam mengidentifikasi dan melakukan personalisasi terkait monitoring *server* yang dikelola dalam sebuah basis data/*database*.
3. Membangun otomatisasi sistem monitoring *server* secara *realtime* menggunakan *device* orange Pi berbasis *web service* secara intranet dengan menggunakan bahasa pemrograman *PHP* dan *MySQL* sebagai *database*.
4. Membangun otomatisasi sistem notifikasi/*remainder* melalui *email*, jika ditemukan *service* pada *server* yang mati atau tidak bekerja dalam keadaan normal.
5. Membangun sistem keamanan data dengan menggunakan MD5 agar data yang diperoleh aman untuk perusahaan tersebut.

#### D. Batasan Masalah

Adapun beberapa poin batasan masalah dari penelitian ini adalah sebagai berikut :

1. Sistem aplikasi ini dibuat dengan menggunakan *device* orange Pi, serta teknologi berbasis *web service*, dimana menggunakan bahasa pemrograman *PHP*.
2. Aplikasi yang dibangun masih menggunakan komunikasi intranet antara *device orange pi* dan *server*, akan tetapi akses *web* hanya dapat diakses secara intranet/jaringan lokal saja.
3. Basis data/*database* yang digunakan yaitu *MySQL*.
4. Pada penelitian ini, hanya membahas aktifitas yang berhubungan identifikasi *server*, serta memberikan notifikasi/*remainder email*, jika ditemukan

kondisi *server* yang mati atau dalam kondisi tidak normal.

5. Model Pengembangan Aplikasi menggunakan *Agile Scrum*.

## II. DASAR TEORI PENUNJANG

### A. Arduino

*Arduino* merupakan pengontrol mikro *single-board* yang bersifat *open-source* atau gratis, yang diturunkan dari *Wiring platform*, lalu dirancang untuk membantu memudahkan penggunaan elektronik dalam berbagai kebutuhan system terintegrasi [2]. Pada umumnya *hardware*-nya memiliki prosesor berjenis *Atmel AVR* dan *software*-nya memiliki bahasa pemrograman dengan kualifikasi sendiri. Untuk mikrokontroler mampu diprogram menggunakan bahasa pemrograman *arduino* yang pada umumnya dapat diprogram dengan menggunakan bahasa pemrograman C ataupun Java, dengan skema *hardware arduino* yang dapat dikembangkan sesuai dengan fungsi dan kebutuhan tertentu.

### B. Orange Pi

Orange Pi adalah sebuah mini PC *open source single-board computer*. Orange Pi sistem kerjanya sama dengan varian Pi yang lain, seperti *Rasbery Pi*, *Nano Pi*, dan varian Pi lainnya [3]. Orange Pi dapat dipasangkan sistem operasi *Android*, *Ubuntu*, *Debian*, dan *Rasbian*. Orange Pi sendiri merupakan mini PC untuk *web server* yang tidak memerlukan layar monitor serta storage yang besar serta sebagai protokol *web server* untuk menampung data-data yang dikirim *PC server* yang dikonitoring. Dan untuk menyimpan aplikasi dari *web* yang dijadikan *web service* untuk mengakses *url* yaitu dari orange pi yang akan dipanggil.

### C. PHP & PHP Mailer

*PHP* merupakan salah satu bahasa pemrograman *web* atau *scripting language* yang dijalankan diserver untuk mengontrol atau mengendalikan berbagai macam komponen pada suatu *web*, sehingga *web* menjadi lebih terstruktur dan seragam. *PHP* dibuat pertama

kali oleh Rasmus Lerdorf, yang pada mulanya digunakan untuk menghitung banyaknya pengunjung pada suatu *homepage* [4].

*PHP Mailer* adalah fungsi *PHP* yang digunakan untuk mengirim *email*, dan hanya berjalan untuk sesama *localhost* [5].

#### D. MySQL

*MySQL* (*My Structured Query Language*) merupakan sebuah DBMS untuk mengelola *database*, *MySQL* juga bersifat *open source* yang memiliki dua bentuk lisensi secara umum yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang terbatas pada penggunaannya) [6]. Selain itu, *MySQL* juga merupakan program pengakses *database* yang bersifat *Multiuser* (Banyak Pengguna) yang dapat diakses dari jaringan internet maupun intranet. Kelebihan utama dari *MySQL* adalah menggunakan bahasa *query* (permintaan) standard *SQL* (*Structured Query Language*).

#### E. InfluxDB

*InfluxDB* merupakan sebuah *times-series database open source* yang dikembangkan oleh *InfluxData*. *InfluxDB* dibangun menggunakan bahasa *GO* dan digunakan sebagai media penyimpanan metrik, sensor data, hingga analisis *real-time*. Fungsi dari *InfluxDB* di aplikasi ini yaitu sebagai penyimpanan data-data berupa *log* atau metrik yang dikirim dari *server* melalui *telegraf*, sehingga peranan *InfluxDB* sangat penting dan dioptimalkan untuk penyimpanan data dalam waktu cepat seperti pemantauan operasi serta memiliki perintah yang sama dengan *SQL* untuk *query data* [7].

#### F. Java

*Java* merupakan salah satu jenis bahasa pemrograman yang berorientasi pada *object*. *Java* juga tersusun dari kumpulan *class* yang dapat dipanggil sewaktu-waktu. *Class* pada *java* terdiri dari metode-metode yang menjalankan pekerjaan, serta akan mengembalikan informasi yang dibutuhkan, setelah menjalankan tugasnya. Dengan demikian, *java* merupakan bahasa pemrograman berbasis objek oriented yang dapat membuat aplikasi baik untuk *platform desktop*,

*web* maupun aplikasi *mobile* [8]. Bahasa pemrograman *java* juga dapat berjalan pada berbagai macam *platform* sistem operasi yang ada saat ini.

#### G. Apache

*Apache* merupakan sebuah perangkat lunak untuk *web server* yang gratis dan paling populer serta *open source* [9]. Selain itu, *apache* diartikan sebagai suatu *web server* yang kompak, modular, serta mengikuti standar protokol *HTTP*.

Adapun *platform* sistem operasi yang didukung oleh *Apache* saat ini yaitu: *Windows*, *MacOS*, *Linux*, *SunOs*, *UnixWare*, *FreeBSD*, *Solaris*, *AIX*, *OpenBSD*, *IRIX*, *SCO*, *NetBSD*, *HPUX*, *BSDI* dan *Digital Unix*.

#### H. Telegraf

*Telegraf* merupakan salah satu jenis *daemon* yang dapat berjalan diberbagai macam jenis *server* manapun, serta dapat mengumpulkan berbagai macam jenis informasi metrik dari sistem server (*CPU*, memori, *storage* dan lainnya). *Telegraf* juga memiliki berbagai macam jenis *plugin output* yang berfungsi untuk mengirim metrik ke berbagai jenis *datastore* termasuk *InfluxDB*, *Graphite*, *OpenTSDB* serta *Datalog* [10].

### III. METODOLOGI PENELITIAN

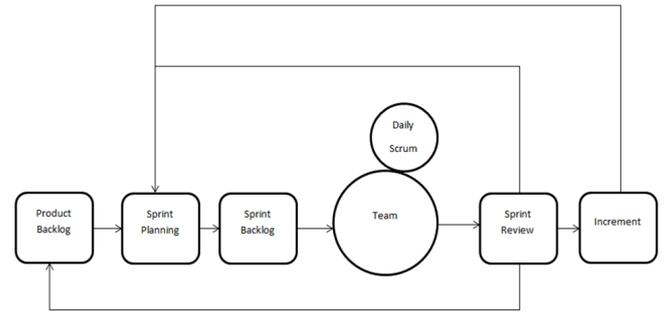
#### A. Tahapan Penelitian

Tahapan penelitian merupakan gambaran umum langkah-langkah yang akan dilakukan oleh peneliti selama penelitian berjalan. Tahapan penelitian dari identifikasi masalah dari studi kasus yang ada, perumusan masalah pada studi kasus, pengumpulan data dengan studi literatur dan pencarian sejenis, pembuatan dan pengembangan sistem dengan metode *scrum*, analisis hasil dan pengujian dengan menggunakan *black box testing*, adapun tahapan penelitiannya sebagai berikut.



Gambar 1. Tahapan Penelitian

## B. Metode Pengembangan Sistem

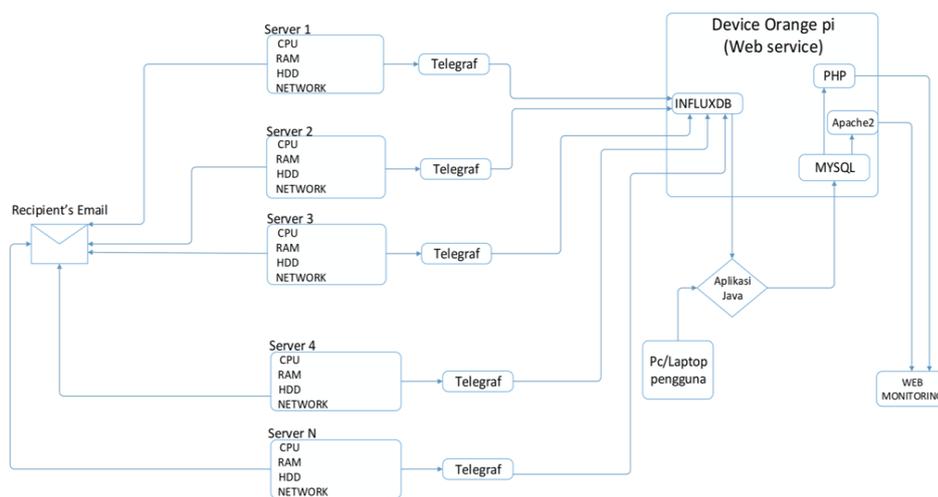


Gambar 2. Tahapan Metode Scrum [11]

Metode *Scrum* merupakan kerangka kerja yang mana setiap orang dapat mengatasi problem adaptif yang cukup kompleks, sementara secara produktif dan kreatif dapat memberikan produk dengan nilai tambah yang tinggi. *Scrum* merupakan salah satu metode rekayasa perangkat lunak dengan menggunakan prinsip-prinsip *Agile*, yang bertumpu pada kekuatan dari kolaborasi tim, *product backlog*, *sprint*, *incremental product*, dan proses iterasi untuk mewujudkan hasil akhir *product* yang terbaik.

## IV. HASIL DAN PEMBAHASAN

### A. Arsitektur Monitoring Server



Gambar 3. Arsitektur Monitoring Server

Pada gambar 3 diatas dijelaskan bahwa, pada gambar 3 diatas dijelaskan bahwa, *device orange pi* merupakan perangkat *web service* pada aplikasi monitoring *server* berbasis web. Pada *device orange pi* yaitu, *influxDB* yang merupakan *database* yang digunakan untuk

menampung data-data *telegraf* yang berasal dari *server*. Kemudian *PHP & MySQL* digunakan untuk menampilkan aplikasi dalam bentuk web yang dapat diakses secara jaringan intranet maupun internet, sedangkan *apache* digunakan

untuk menjalankan *web server* pada aplikasi monitoring *server* ini.

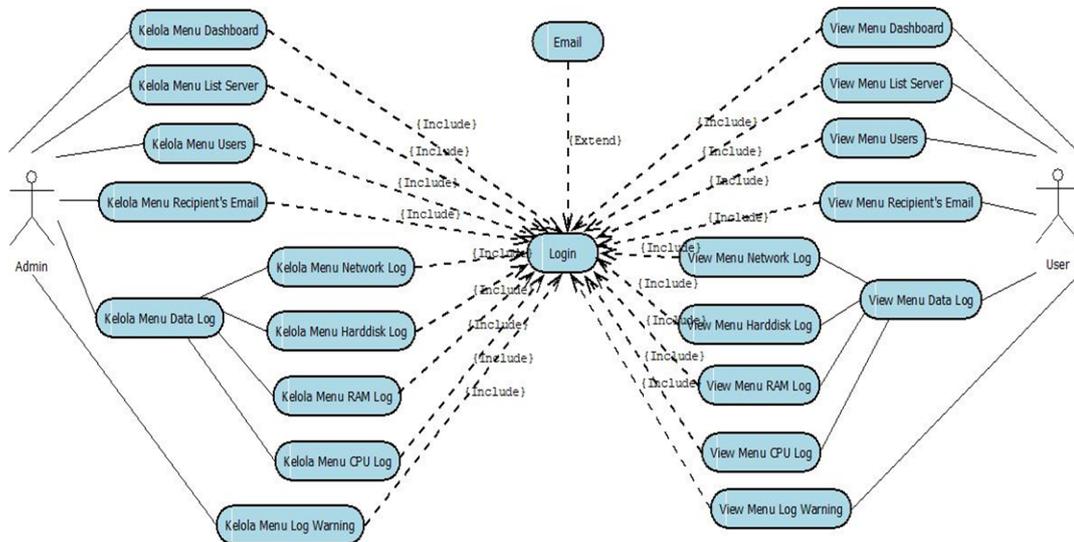
### B. Diagram Sistem

Desain dari aplikasi monitoring *server* pada penelitian ini digambarkan dengan menggunakan

UML berupa *use case diagram*, *activity diagram* dan *sequence diagram*.

#### 1. Use Case Diagram

*Use case diagram* pada aplikasi sistem monitoring *server*, terlihat pada gambar 4 sebagai berikut :



Gambar 4. Use case Diagram Aplikasi Sistem Monitoring Server

Dari *use case diagram* diatas dijelaskan bahwa pada saat pengguna *login* sebagai *admin* dan *user*, maka pengguna bisa melakukan pengelolaan menu dari aplikasi tersebut. Pada gambar tersebut, `<<include>>` berarti bahwa *admin* dan *user* harus terlebih dahulu *login* agar dapat menjalankan fungsi aplikasi tersebut, dan pada gambar `<<extend>>` berarti bahwa *admin* dan *user* baru bisa mengelola *email* apabila pesan akan dikirimkan secara otomatis.

Ketika *user* sebagai *admin*, diharuskan melakukan terlebih dahulu *login* menggunakan *username* dan *password* yang sudah di *set* sebelumnya. Dan untuk *user* baru terlebih dahulu harus ditambahkan, baru *user* baru bisa *login* dengan menggunakan *username* dan *password* yang sudah terdaftar pada *database*. Selanjutnya *admin* akan mengelola suatu menu pada aplikasi tersebut dan *user* hanya bisa *view* pada aplikasi tersebut.

Selanjutnya *admin* mengelola menu *dashboard*, *admin* atau *user* baru akan menampilkan informasi data *list server*, *users*, *recipient's email*, data *log*, *log warning* dan

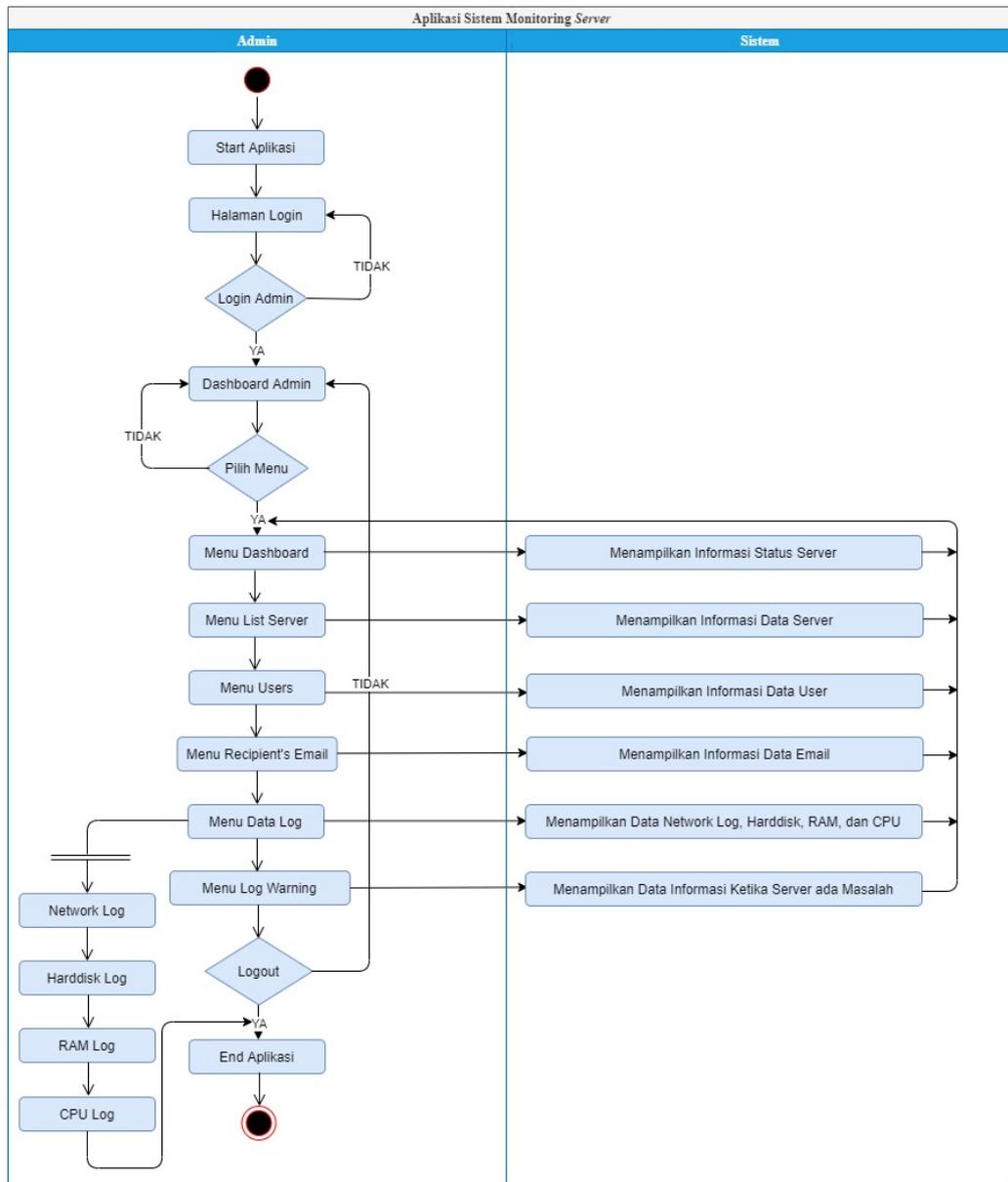
menampilkan beberapa *server* yang telah terhubung.

Pada saat *admin* mengelola pada halaman menu *list server*, *admin* akan menampilkan informasi data *server* serta *admin* dapat menambahkan *server* baru, mengubah data *server*, menghapus data *server*, dan bisa menyalin data *server* serta menyimpan data dalam bentuk file : *excel*, *csv*, dan *pdf*. Apabila *admin* mengelola pada halaman menu *users*, *admin* akan menampilkan informasi data *user* serta *admin* dapat menambahkan *user* baru sebagai *login admin*, mengubah data *user*, serta menghapus data *user*.

Pada halaman menu data *log*, *admin* dan *user* akan menampilkan informasi data *log* pada *network log*, *harddisk log*, *RAM log*, *CPU log* serta *admin* menyalin data semua *log* dan menyimpan data semua *log* dalam bentuk file : *excel*, *csv*, dan *pdf*.

#### 2. Activity Diagram

*Activity diagram* dalam Aplikasi Sistem Monitoring Server, dapat digambarkan sebagai berikut :



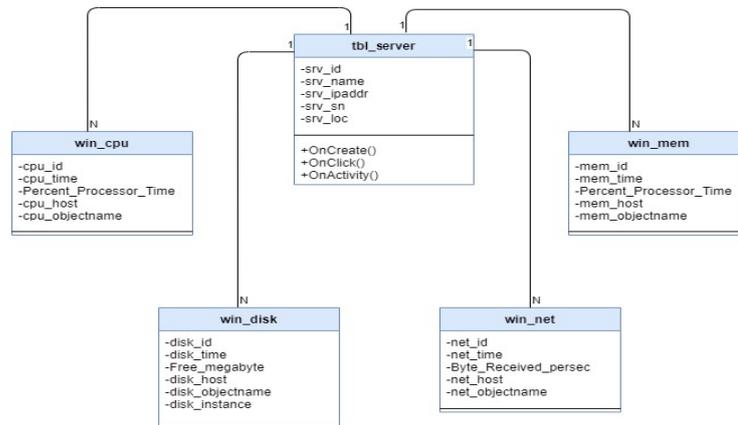
Gambar 5. Activity Diagram Aplikasi Sistem Monitoring Server

Pada *activity* diagram yang terlihat pada gambar 5. Setelah proses *login* sebagai *admin* maka aplikasi akan menampilkan semua menu dimana *admin* dapat melakukan tambah, ubah, hapus dari semua data yang ada. Pada halaman utama *dashboard* *admin* akan menampilkan informasi data *list server*, *users*, *recipient's email*, data *log*, *log warning* dan menampilkan beberapa *server* yang telah terhubung. Pada halaman menu data *log*, *admin* akan menampilkan informasi data *log* pada *network log*, *harddisk log*, *RAM log*, *CPU log* serta *admin* menyalin data semua *log* dan menyimpan

data semua *log* dalam bentuk file : *excel*, *csv*, dan *pdf*.

### 3. Class Diagram

*Class diagram* yang digunakan pada penelitian ini, menggunakan 5 *class* yang terdiri dari *tbl\_server*, *win\_cpu*, *win\_disk*, *win\_net* serta *win\_mem*. Berikut terlihat pada gambar 6 yaitu *class diagram* yang menggambarkan keterhubungan antara *class-class* yang terdapat pada aplikasi sistem monitoring *server* berbasis *web* ini.



Gambar 6. Class Diagram Aplikasi Sistem Monitoring Server

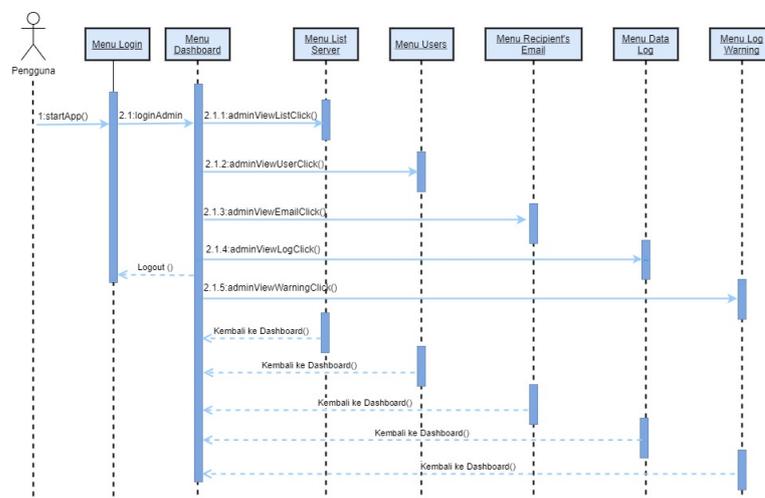
Class diagram di atas menjelaskan bahwa ketika dalam input data pada aplikasi setiap menunya, dari data *tbl\_server* diatas, setiap server akan mengelola banyak data CPU, data harddisk, data memori, dan data jaringan setelah berhasil diinput oleh admin pada menu *data log*.

Tabel *server* berasosiasi dengan tabel CPU, harddisk, memori dan Network. Tabel *user* sebagai *admin* berasosiasi dengan tabel *server*, tabel *email*, tabel data *log*, dan tabel *log warning*. Satu pengguna admin dapat masuk ke *dashboard admin*. Satu *admin* dapat melihat banyak data informasi pada setiap menunya yang ada di aplikasi, dan dapat menginput suatu data yang hanya bisa menambahkan suatu data setiap menunya, menghapus suatu data, dan merubah suatu data pada setiap masing-masing menu

aplikasi, serta admin bisa menyalin suatu data, menyimpan suatu data pada menu aplikasi ini. Pada tabel di atas, tabel *server* berasosiasi dengan tabel CPU, harddisk, memori dan tabel network. Satu server dapat mengakses banyak data dari data tabel CPU, harddisk, memori dan tabel network.

#### 4. Sequence Diagram

Sequence diagram dari aplikasi sistem monitoring server terlihat pada gambar 7 yang terhubung antara pengguna dan beberapa menu seperti menu login, menu dashboard, menu list server, menu users, menu recipients email, menu data log, serta menu log warning. Berikut ini adalah sequence diagram dari aplikasi sistem monitoring server berbasis web tersebut.

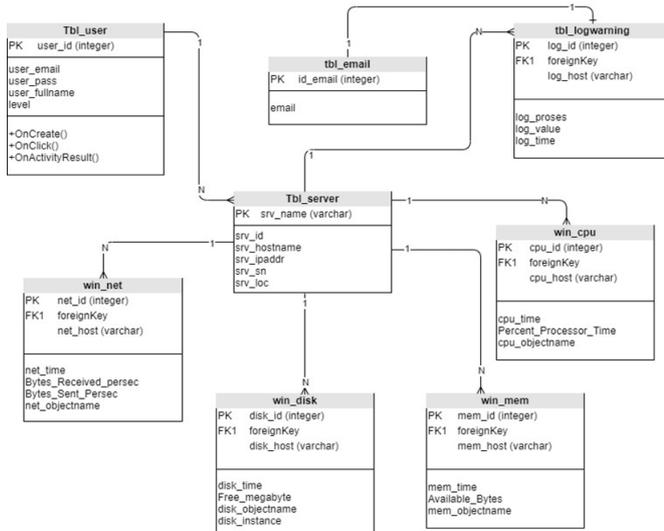


Gambar 7. Sequence Diagram Aplikasi Sistem Monitoring Server

Pada diagram *sequence* diatas, bisa dilihat yang menjadi *actor* adalah admin. *Activation boxes* biasanya memiliki garis yang memberitahu aktifitas yang terjadi ketika *actor* atau objek berinteraksi ke objek lain. Ketika *actor login* sebagai admin, maka aktifitas yang terjadi adalah menu *dashboard* admin, menu *list server*, menu *users*, menu *recipient's email*, menu data *log*, menu *log warning*.

### 5. Entity Relationship Diagram

*Entity Relationship Diagram* (ERD) pada aplikasi sistem monitoring *server* berbasis *web* memiliki entity sebanyak 8 tabel seperti *tbl\_user*, *tbl\_email*, *tbl\_logwarning*, *tbl\_server*, *win\_net*, *win\_disk*, *win\_cpu* serta *win\_mem*. Adapun *Entity Relationship Diagram* pada penelitian ini dapat terlihat pada gambar 8 berikut:



Gambar 8. Entity Relationship Diagram

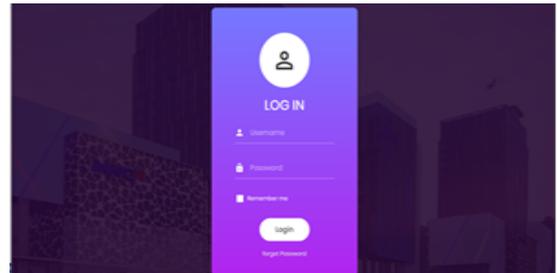
Dari diagram ERD di atas, dapat dijelaskan bahwa admin (*user*) dapat mengelola banyak *server* yang akan dikelola oleh admin. Admin dapat mengelola banyak *recipient's email* yang akan dikelola admin, dan dapat mengelola banyak *log warning* yang akan dikelola admin, serta data *log*.

### C. Hasil Pengujian Black Box dengan Teknik EP

Metode pengujian aplikasi yang digunakan dalam penelitian ini, menggunakan parameter pengujian fungsional dengan metode *black box* disertai dengan teknik *Equivalence Partitioning* (EP). Teknik EP merupakan metode pengujian

*black box* yang memecah atau membagi antara domain *input* dari program ke dalam bentuk kelas-kelas data, sehingga setiap kelas uji dapat diperoleh sesuai dengan fungsinya. Kondisi input bisa saja berupa nilai *numeric*, *range* nilai, maupun nilai yang berhubungan dengan kondisi *Boolean*. Adapun hasil pengujian *black box* fungsionalnya sebagai berikut:

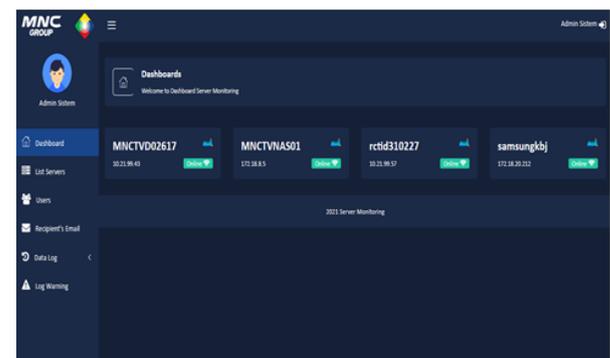
#### 1. Halaman Login



Gambar 9. Antarmuka Halaman Login

Pada halaman *login* aplikasi sistem monitoring *server*, akan memasukkan *username* dan *password* agar dapat melanjutkan ke halaman berikutnya untuk membuka aplikasi. Pada tahap ini *username* dan *password* sudah terenkripsi menggunakan sistem keamanan MD5 (*Message Direct Algorithm 5*), sehingga dapat meningkatkan keamanan pada aplikasi monitoring *server*.

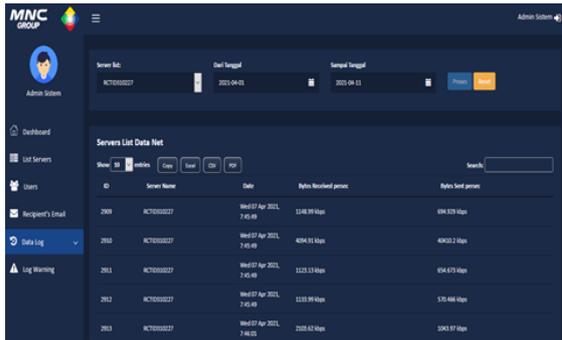
#### 2. Menu Dashboard



Gambar 10. Antarmuka Halaman Menu Dashboard

Menu *dashboard* akan menampilkan informasi *server* yang terdapat pada aplikasi ini serta menampilkan *server* dalam keadaan *online* ataupun *offline*.

#### 3. Menu Data Log



Gambar 11. Antarmuka Halaman Menu Data Log

Menu *data log* akan menampilkan informasi terkait dengan *log data* utilisasi *cpu*, utilisasi memori, utilisasi *storage* dan utilisasi *network*. Serta jika masing-masing utilisasi melebihi batas *threshold* yang telah ditentukan sebelumnya, maka akan masuk ke menu *log warning* dan langsung akan mengirimkan *alert* berupa *email* kepada *user*.

#### D. Hasil Pengujian Black Box Admin dan User

Pengujian terhadap sistem yang dibuat dilakukan dengan metode pengujian *black box* pada aplikasi sistem monitoring *server* berbasis *web service* dengan *device* orange pi kasus: PT. MNC Televisi Indonesia – MNC Group. Berikut ini tabel dari pengujian *Black Box*-nya sebagai berikut:

Tabel 2. Hasil Pengujian Black Box Untuk Admin

| No | Kelas Uji                         | Daftar Pengujian  | Kasus Uji  | Hasil Yang Di Harapkan   | Valid (√) atau Tidak Valid (x) |
|----|-----------------------------------|---|--|--|--------------------------------|
| 1. | Login                             | Input data <i>login</i>                                   | Username = email<br>Password = admin<br>Klik tombol <i>login</i>                   | Masuk ke halaman <i>dashboard</i> admin aplikasi   | √                              |
| 2. | Fungsi pada Menu <i>Dashboard</i> | Pengujian menampilkan halaman <i>dashboard</i> pada admin | Admin memilih menu <i>dashboard</i>  | Menampilkan Informasi data <i>server</i> baik yang <i>online</i> maupun <i>offline</i>       | √                              |
|    |                                   | Pengujian menampilkan halaman <i>list server</i>          | Admin memilih menu <i>list server</i>  | Menampilkan informasi data <i>server</i> pada halaman ini                                    | √                              |
|    |                                   | Tambah data <i>Server</i>                                 | Admin menambahkan data <i>server</i> baru dengan lengkap dan menekan tombol simpan | Menampilkan Pop Up dan data yang sudah ditambahkan serta data tersimpan pada <i>database</i> | √                              |
|    |                                   | Ubah data   | Admin melakukan  | Menampilkan Pop Up dan   |                                |

|    |   |   |  |   |   |
|----|---|---|--|---|---|
| 3. | Fungsi pada Menu <i>List Server</i>       | <i>Server</i>   | perubahan data <i>server</i> dan melakukan klik tombol ubah                      | data yang sudah dirubah serta data tersimpan pada <i>database</i>   | √ |
|    |   | Hapus data <i>Server</i>                                    | Admin melakukan hapus data <i>server</i> dan melakukan klik tombol hapus         | Menampilkan Pop Up dan menampilkan informasi pada halaman apakah sudah terhapus serta secara otomatis akan terhapus pada <i>database</i>      | √ |
| 4. | Fungsi pada Menu <i>User</i>              | Pengujian menampilkan halaman data <i>user</i>              | Admin memilih menu <i>user</i>   | Menampilkan halaman informasi data <i>user</i> pada aplikasi  | √ |
|    |   | Tambah data <i>user</i>                                     | Admin menambahkan data <i>user</i> baru dengan lengkap dan menekan tombol simpan | Menampilkan Pop Up dan data yang sudah ditambahkan serta data tersimpan pada <i>database</i>  | √ |
|    |   | Ubah data <i>user</i>                                       | Admin melakukan perubahan data <i>user</i> dan melakukan klik tombol ubah        | Menampilkan Pop Up dan data yang sudah dirubah serta data tersimpan pada <i>database</i>  | √ |
|    |   | Hapus data <i>user</i>                                      | Admin melakukan hapus data <i>user</i> dan melakukan klik tombol hapus           | Menampilkan Pop Up dan menampilkan informasi pada halaman apakah sudah terhapus serta secara otomatis akan terhapus pada <i>database</i>      | √ |
| 5. | Fungsi pada Menu <i>Recipient's Email</i> | Pengujian menampilkan halaman data <i>Recipient's Email</i> | Admin memilih menu <i>Recipient's Email</i>                                      | Tampil halaman data <i>Email</i>  | √ |
|    |   | Tambah Data <i>Email</i>                                    | Admin memasukkan data <i>Email</i> dengan lengkap dan menekan tombol simpan      | Menampilkan Pop Up dan data <i>email</i> yang sudah ditambahkan dan data tersimpan pada <i>database</i>                                       | √ |
|    |   | Ubah Data <i>Email</i>                                      | Admin menekan tombol ubah data pada salah satu <i>Email</i>                      | Menampilkan Pop Up dan halaman data <i>email</i> yang telah diubah serta data otomatis akan ditampilkan sesuai perubahan yang dilakukan admin | √ |
|    |   | Hapus Data <i>Email</i>                                     | Admin melakukan hapus data dan melukan klik tombol hapus                         | Menampilkan Pop Up dan data <i>email</i> otomatis akan terhapus pada halaman ini serta akan terhapus data dari <i>database</i>                | √ |

|    |                              |  |  |  |   |
|----|------------------------------|--|--|--|---|
| 6. | Fungsi pada Menu Data Log    | Pengujian menampilkan halaman data <i>log</i>  | Admin memilih menu data <i>log</i>         | Menampilkan halaman informasi data <i>log</i>  | √ |
| 7. | Fungsi pada Menu Log Warning | Pengujian menampilkan halaman data <i>log warning</i> dan mengirim <i>alert</i> ke email | Admin memilih menu data <i>log warning</i> | Menampilkan halaman informasi data <i>log warning</i> dan mengirim <i>alert</i> ke email ketika server ada masalah | √ |

Tabel 3. Hasil Pengujian *Black Box* Untuk *User*

| No | Kelas Uji                         | Daftar Pengujian   | Kasus Uji  | Hasil Yang Di Harapkan  | Valid (√) atau Tidak Valid (x) |
|----|-----------------------------------|--|--|---|--------------------------------|
| 1. | <i>Login</i>                      | Input Data login   | <i>Username</i> = email<br><i>Password</i> = user klik tombol <i>login</i> | Masuk ke halaman <i>dashboard</i> user aplikasi   | √                              |
| 2. | Fungsi pada menu <i>Dashboard</i> | Pengujian menampilkan <i>Dashboard</i> pada <i>user</i>                                  | User memilih menu <i>Dashboard</i>   | Menampilkan informasi data <i>server</i> baik yang <i>online</i> maupun <i>offline</i>                                    | √                              |
| 3. | Fungsi pada Menu Data Log         | Pengujian menampilkan halaman data <i>log</i>  | User memilih menu data <i>log</i>  | Menampilkan halaman informasi data <i>log</i>   | √                              |
| 4. | Fungsi pada Menu Log Warning      | Pengujian menampilkan halaman data <i>log warning</i> dan mengirim <i>alert</i> ke email | User memilih menu data <i>log warning</i>                                  | Menampilkan halaman informasi data <i>log warning</i> dan mengirim <i>alert</i> ke email ketika <i>server</i> ada masalah | √                              |

## V. SIMPULAN & SARAN

### A. Simpulan

Berdasarkan pembahasan dan hasil pengujian yang diperoleh, maka dapat ditarik beberapa poin simpulan sebagai berikut:

1. Dengan dilakukannya demo secara langsung dan pengetesan oleh dosen pembimbing serta beberapa penguji lainnya, dengan hasil yang valid maka telah berhasil dibangunnya sistem monitoring *server* berbasis *web*.
2. Aplikasi sistem monitoring *server* berbasis *web* dapat mempermudah untuk memonitoring *server* secara *remote*.
3. Pengguna baru lebih cepat memahami alur penggunaan aplikasi ini karena fungsi pada aplikasi ini mudah dipahami.
4. Aplikasi ini berhasil memberikan informasi mengenai aktifitas *server* yang dipantau.

5. Aplikasi ini berhasil mengirim *alert* ke email ketika *server* mengalami mati secara tiba-tiba.
6. Dengan adanya sistem monitoring ini, diharapkan dapat memberikan informasi terkait *log server* seperti status penggunaan *CPU*, penggunaan memori, koneksi *network* serta *utilisasi storage*.
7. Pelaporan melalui email terbukti memberikan respon yang cepat agar *administrator* dapat segera menangani permasalahan yang ada.
8. Hasil pengujian aplikasi menggunakan metode pengujian *black box* pada sistem, menunjukkan bahwa sistem mempunyai tingkat fungsionalitas yang baik, semua fungsi yang berada pada sistem dapat bekerja dengan baik.

### B. Saran

Sistem ini masih banyak keterbatasan dan kekurangan, sehingga dibutuhkan beberapa saran membangun diantaranya :

1. Aplikasi sistem monitoring *server* berbasis *web*, dapat dikembangkan dengan pengembangan aplikasi berbasis *android*, agar lebih mudah dan praktis penggunaannya.
2. Metode pengambilan data *log server* dengan *influxdb*, diharapkan dapat dikembangkan dengan metode lain yang lebih cepat.
3. Aplikasi sistem monitoring *server*, dapat dikembangkan dengan menambahkan fitur baru pada aplikasi ini.
4. Aplikasi sistem monitoring *server*, dapat dikembangkan dengan menggunakan *device* dan protokol yang lain.

## DAFTAR PUSTAKA

- [1] F. Sains and D. Teknologi, "PERANCANGAN SISTEM MONITORING SERVER DENGAN MENGGUNAKAN BOT TELEGRAM SEBAGAI MEDIA PROGRAM STUDI TEKNIK INFORMATIKA."
- [2] P. Putu, T. Winata, I. Wayan, A. Wijaya, and I. M. Suartika, "Rancang Bangun Sistem Monitoring Output dan Pencatatan Data pada Panel Surya Berbasis Mikrokontroler Arduino," *J. SPEKTRUM*, vol. 3, no. 1, pp. 1–6, 2016, Accessed: Feb. 10, 2022. [Online]. Available: <https://ojs.unud.ac.id/index.php/spektrum/article/view/21638>.
- [3] A. Mappatoba, I. Haruna, and A. Agussalim, "Prototype Sistem Monitoring Volume Bak Sampah Online Berbasis Arduino Nano dan

- Orange Pi,” *JITCE (Journal Inf. Technol. Comput. Eng.*, vol. 4, no. 01, pp. 22–27, Mar. 2020, doi: 10.25077/JITCE.4.01.22-27.2020.
- [4] A. Fanggidae, A. M. Fanggidae, H. Hermawan, and H. I. Pratiwi, “Sistem Monitoring Server Dengan Menggunakan SNMP,” *WIDYAKALA J. Pembang. JAYA Univ.*, vol. 6, no. 2, pp. 163–166, Sep. 2019, doi: 10.36262/widyakala.v6i2.218.
- [5] S. A. Puji Utami, S. Agustian, and I. Fauzi Aditya Sayogo, “PERANCANGAN ONLINE NETWORK MONITORING BERBASIS PHP DAN SNMP,” *Semin. Nas. Apl. Teknol. Inf.*, 2006.
- [6] “Perancangan Web Sistem Monitoring Stop Kontak Terkontrol menggunakan Pemrograman PHP dan MySQL | Quzania | Transient: Jurnal Ilmiah Teknik Elektro.” <https://ejournal3.undip.ac.id/index.php/transient/article/view/28118> (accessed Feb. 10, 2022).
- [7] H. - and W. Andriyani, “STUDI KOMPARASI MENYIMPAN DAN MENAMPILKAN DATA HISTORI ANTARA DATABASE TERSTRUKTUR MARIADB DAN DATABASE TIDAK TERSTRUKTUR INFLUXDB,” *J. Teknol. TECHNOSCIENTIA*, pp. 168–174, Feb. 2020, doi: 10.34151/TECHNOSCIENTIA.V12I2.2663.
- [8] R. Nurfaiza<sup>1</sup>, I. Winarno, M. Sst, and <sup>2</sup> Kom, “Aplikasi Monitoring Proses pada Komputer Server dengan Menggunakan Java,” Accessed: Feb. 10, 2022. [Online]. Available: <http://www.pcmmedia.co.id/detail.asp?Id=2159&Cid=22&Eid=54>.
- [9] S. Taftazanie, A. B. Prasetijo, and E. D. Widiyanto, “Aplikasi Pemantau Perangkat Jaringan Berbasis Web Menggunakan Protokol SNMP dan Notifikasi SMS,” *J. Teknol. dan Sist. Komput.*, vol. 5, no. 2, p. 62, May 2017, doi: 10.14710/JTSISKOM.5.2.2017.62-69.
- [10] D. Rahman, H. Amnur, and I. Rahmayuni, “Monitoring Server dengan Prometheus dan Grafana serta Notifikasi Telegram,” *JITSI J. Ilm. Teknol. Sist. Inf.*, vol. 1, no. 4, pp. 133–138, Dec. 2020, doi: 10.30630/JITSI.1.4.19.
- [11] J. S. Komputer, D. Kecerdasan, B. Volume, and V. Nomor, “Aplikasi Sistem Personalisasi Dan Monitoring Pengunjung Berbasis Smartcard RFID (Radio Frequency Identification) Studi Kasus Candi Borobudur,” *J. SISKOM-KB (Sistem Komput. dan Kecerdasan Buatan)*, vol. 5, no. 1, pp. 1–10, Sep. 2021, doi: 10.47970/SISKOM-KB.V5I1.217.