e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

Rancang Bangun *Centralized Monitoring* Jaringan dan Server Berbasis *Embedded System* (Studi Kasus CV. Intek Solusindo)

Alif Gufron Pratama, Sardjono, Marwondo
Fakultas Tekonologi dan Informatika, Universitas Informatika dan Bisnis Indonesia
Email: alifgufronp@gmail.com, sardjono@unibi.ac.id marwondo@unibi.ac.id

Abstrak

CV. Intek Solusindo Mandiri merupakan perusahaan yang bergerak dibidang jasa teknologi informasi (TI) dan *system integrator* (SI), Saat ini sudah memiliki banyak klien yang tersebar di seluruh wilayah Jawa Barat. Untuk memberikan pelayanan yang maksimal maka CV. Intek Solusindo harus memilki *server* yang dapat monitor perangkat jaringan dan *server* secara *real-time*. Dalam melakukan monitor jaringan dan server antar klien masih menggunakan *software Paesseler PRTG* dengan keterbatasan tidak dapat memberikan notifikasi *alert* secara *real-time*. Melalui perencanaan, pengembangan dan pengujian dengan menggunakan beberapa *dependency software* seperti influxdb, grafana, telegraf, bash scripting dan bahasa pemrograman C++ yang berjalan pada perangkat keras *embedded system* amlogic S905X sebagai server monitor jaringan dan NodeMCU ESP8266 pendukung *Internet Of Things* (IoT). Sistem monitoring ini memiliki fitur notifikasi *alert* menggunakan *telegram* jika terjadi kendala pada perangkat jaringan. Dengan adanya sistem monitoring jaringan dan *server* secara terpusat berbasis *Embedded System* dapat membantu admin jaringan dalam menjalankan tugasnya untuk me-monitoring jaringan dan *server* tanpa harus *standby* di depan komputer.

Kata Kunci: Monitor Jaringan, *Embedded System*, IoT, Server, Telegram, Sumber Daya Minimal.

Abstract

CV. Intek Solusindo Mandiri is a company engaged in information technology (IT) and system integrator (SI), Currently has many clients spread throughout West Java. To provide maximum service, CV. Intek Solusindo must have servers that can monitor network and server devices in real time. To monitor network and server between clients curently using Paesseler PRTG, in monitoring the network still has limitations not being able to provide real-time notifications. Through planning, development and testing using several dependency software such as influxdb, grafana, telegraf, bash scripting and C ++ which runs on embedded system hardware amlogic S905X as a network monitor server and NodeMCU ESP8266 supporting Internet Of Things (IoT). This monitoring system has a feature of telegram notification if there is a problem with the network device. With the centralized network and server monitoring system based on Embedded System can help network admins in running their tasks to monitor networks and servers without having to stand by in front of the computer.

Keywords: *Network Monitoring, Embedded System, IoT, Server, Telegram, Minimal Resources*

1 PENDAHULUAN

Teknologi Informasi dan Komunikasi (TIK) telah menjadi bagian penting dalam pembangunan teknologi di seluruh dunia. Dengan berbagai inovasi dan kemajuan yang terus menerus, TIK sekarang menawarkan berbagai cara untuk mempercepat

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

komunikasi, mempromosikan informasi, dan membantu orang untuk meningkatkan produktivitas. Dengan kemajuan teknologi ini, orang dapat mengakses informasi yang lebih cepat dan akurat, menjalankan bisnis dengan lebih sukses, dan menikmati kenyamanan dan fleksibilitas yang baru. Menurut Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) mengungkapkan bahwa penggunaan internet di Indonesia sudah mencapai 77,02% dari tahun 2021-2022. (Bayu, 2022).

Dibalik kemajuan teknologi informasi dan komunikasi tersebut maka muncul suatu kebutuhan bagaimana melakukan monitoring jaringan. Monitoring jaringan merupakan kegiatan yang dilakukan untuk memantau setiap perubahan yang terjadi secara *real time* pada *environment* perangkat jaringan baik *server*, *router*, *switch* bahkan perangkat *Internet of Things* (IoT). Monitoring jaringan menjadi suatu hal yang sulit dilakukan apabila jaringan komputer pada suatu organisasi memilki jangkauan yang luas atau bahkan tersebar secara geografis. Masalah-masalah pada jaringan komputer yang sering terjadi diantaranya adalah kerusakan perangkat jaringan, kabel yang rusak dan putus, penggunaan *bandwidth* yang melebihi batas maksimum, *service* pada *server* mengalami *down* dan lain sebagainya. Dimana hal-hal tersebut tidak dapat diketahui secara langsung dan akan membutuhkan waktu pemeriksaan jaringan yang cukup lama. Di sinilah admin jaringan harus mampu menjaga dan mengelola kestabilan infrastruktur jaringan yang dikelolanya.

CV. Intek Solusindo Mandiri merupakan perusahaan yang bergerak dibidang Teknologi Informasi (TI) dan *system integrator* (SI). Jasa yang sediakan mulai dari *Networking Management, Cabling, Hardware, Closed Circuit Television* (CCTV), *Fiber Optik, Local Area Network* (LAN), *Wireless Fidelity* (WIFI), *Private Automatic Branch eXchange* (PABX). Pada saat laporan ini dibuat CV. Intek ini sudah memiliki banyak klien yang tersebar di seluruh wilayah Jawa Barat., yang tentunya memiliki banyak perangkat jaringan seperti *server*, *switch*, *router* dan *environment* lainnya yang harus selalu dipantau dan dimonitoring performa dan ketersediaan layanannya selain dapat me-monitoring trafik dan *environment* jaringan sistem juga dapat mendeteksi jika adanya kebakaran pada ruang *server* atau rack *server*.

Ruang lingkup masalah penelitian ini agar membantu mengidentifikasi masalah yang akan dibahas dan membatasi jangkauan proses yang akan dibahas adalah sebagai berikut:

1. Menggunakan *Embedded System* ARM-V8, dengan *Operating System* Armbian Ubuntu sebagai *server monitoring*.
2. Menggunakan NodeMCU ESP8266 Sebagai *Exporter* Sensor.
3. Menggunakan telegram sebagai *notification alert*.
4. Penelitian berfokus pada monitoring jaringan dan *server* berupa trafik, dan environment jaringan.
5. Penelitian ini berfokus untuk mengatasi monitoring jaringan dan *server* secara terpusat.

Penelitian ini bertujuan untuk merancang dan membangun sebuah sistem *monitoring* jaringan dan *server* secara terpusat berbasis *Embedded System* yang berdaya rendah dan *portable*, dengan terwujudnya sistem monitoring secara terpusat ini memudahkan admin jaringan dalam melakukan *monitoring*, menganalisa dan mengambil keputusan terhadap perangkat jaringan yang dikelolanya. Kemudian sistem dapat memberikan *notifikasi* secara *real-time* jika terjadi masalah atau gangguan pada perangkat jaringan dan *server*, selain dapat me-monitoring trafik dan environment jaringan sistem dapat mendeteksi jika adanya kebakaran.

2 KAJIAN PUSTAKA

2.1 Perangkat Lunak

Software dalam Bahasa Indonesia adalah Perangkat Lunak, Perangkat Lunak berisi tentang perintah atau instruksi yang berfungsi sebagai pengatur aktifitas komputer. Perangkat lunak sebagai sistem program untuk mengolah data supaya user atau

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

pengguna bisa berkomunikasi dengan komputer.

Pengertian perangkat lunak menurut Rosa dan Salahudin (2019) menjelaskan bahwa perangkat lunak adalah program komputer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumentasi kebutuhan. Model desain, dan penggunaan (user manual). Sebuah program komputer tanpa terasosiasi dengan dokumentasi nya maka belum dapat disebut perangkat lunak (*software*). Sebuah perangkat lunak juga sering disebut dengan sistem perangkat lunak, Sistem berarti kumpulan komponen yang saling terkait dan mempunyai satu tujuan yang ingin dicapai.

Sistem perangkat lunak berarti sebuah sistem yang memiliki komponen berupa perangkat lunak yang memiliki hubungan satu sama lain untuk memenuhi kebutuhan pelanggan (*customer*) adalah orang atau organisasi yang memesan atau membeli perangkat lunak (*software*) dari pengembang perangkat lunak atau bisa dianggap bahwa pelanggan (*customer*) adalah orang atau organisasi yang dengan sukarela mengeluarkan uang untuk memesan atau membeli perangkat lunak. User atau pemakai perangkat lunak adalah orang yang memiliki kepentingan untuk memakai atau menggunakan perangkat lunak untuk memudahkan pekerjaan.

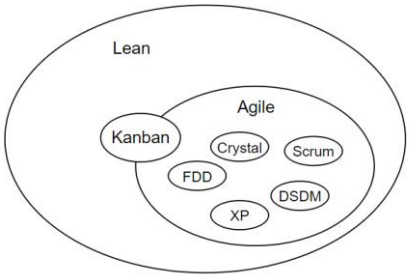
2.2 Rekayasa Perangkat Lunak

Menurut Rosa dan Salahudin (2019) menyatakan rekayasa perangkat lunak (software engineering) merupakan pembangunan dengan menggunakan prinsip atau konsep rekayasa dengan tujuan menghasilkan perangkat lunak yang bernilai ekonomi yang dipercaya dan bekerja secara efisien menggunakan mesin. Perangkat lunak banyak dibuat dan pada akhirnya seiring tidak digunakan karena tidak memenuhi kebutuhan pelanggan atau bahkan karena masalah non-teknis seperti pemakai perangkat lunak (user) untuk mengubah cara kerja dari manual ke otomatis, atau ketidakmampuan user menggunakan komputer. Oleh karena itu, rekayasa perangkat lunak dibutuhkan agar perangkat lunak yang dibuat tidak hanya menjadi perangkat lunak yang tidak terpakai.

Menurut Pressman dan Maxim (2020) *software* telah menjadi elemen kunci dalam evolusi sistem berbasis komputer dan produk. Selama 50 tahun terakhir, perangkat lunak telah berkembang pemecah masalah khusus dan perangkat informasi analisis untuk sebuah industri dalam dirinya sendiri. *Software* terdiri dari program, data, dan dokumen. Masing-masing item ini terdiri dari konfigurasi yang diciptakan sebagai bagian dari proses rekayasa perangkat lunak. Tujuan dari rekayasa perangkat lunak adalah untuk menyediakan kerangka kerja untuk membangun perangkat lunak dengan kualitas yang lebih tinggi.

2.3 Model Pengembangan Agile

Pressman dan Maxim (2020), *Agile Software Development* adalah sekumpulan metodologi pengembangan perangkat lunak yang berbasis pada pengembangan iteratif, di mana persyaratan dan solusi berkembang melalui kolaborasi antar tim yang terorganisir. Istilah ini diciptakan pada tahun 2001 ketika *Agile Manifesto* dirumuskan.



Gambar 1. *Agile* Development

Metode *Agile* umumnya mempromosikan disiplin proses manajemen proyek yang mendorong inspeksi dan adaptasi; filosofi kepemimpinan yang mendorong kerja sama dalam tim, pengorganisasian dan akuntabilitas praktek rekayasa yang memungkinkan pengiriman perangkat lunak berkualitas tinggi dengan cepat dan pendekatan bisnis yang sejalan dengan pengembangan kebutuhan pelanggan dan tujuan perusahaan.

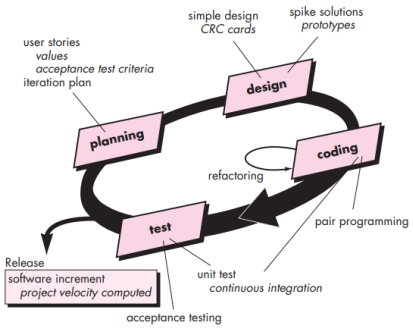
e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

2.4 *Extreme Programming* (XP)

Extreme Programming (XP) adalah salah satu metode pengembangan perangkat lunak yang dilakukan secara iteratif dan inkremental. XP memfokuskan pada pengembangan perangkat lunak yang cepat, berkualitas tinggi, dan adaptif. XP diperkenalkan oleh Kent Beck (2005) ketika ia ditunjuk untuk menangani sebuah proyek penggajian dari *Chrysler* yang dikenal dengan C3 (*Comprehensive Compensation Chrysler*). Proyek ini dimulai sekitar maret 1996, proyek tersebut terancam gagal karena rumitnya sistem yang dibuat dan kegagalan pada saat memasuki tahap uji sistem (testing). Pihak *Chrysler* akhirnya menyewa Kent Beck sebagai konsultan di bidang *software engineering*. Kemudian Kent Beck dikenal sebagai pencetus XP.

Menurut Pressman dan Maxim (2020), *Extreme Programming* adalah metodologi pengembangan perangkat lunak yang ditujukan untuk meningkatkan kualitas perangkat lunak dan tanggap terhadap perubahan kebutuhan pelanggan. Jenis pengembangan perangkat lunak semacam ini dimaksudkan untuk meningkatkan produktivitas dan memperkenalkan pos pemeriksaan di mana persyaratan pelanggan baru dapat diadopsi.

Dalam tahapanya XP memiliki nilai-nilai mendasar dalam pengembangan perangkat lunak yaitu: *Communication* (Komunikasi), *Simplicity* (Kesederhanaan), *Feedback* (Umpan Balik), *Courage* (Keberanian), *Respect* (Menghormati).



Gambar 2. Alur Kerja *Extreme Programing*

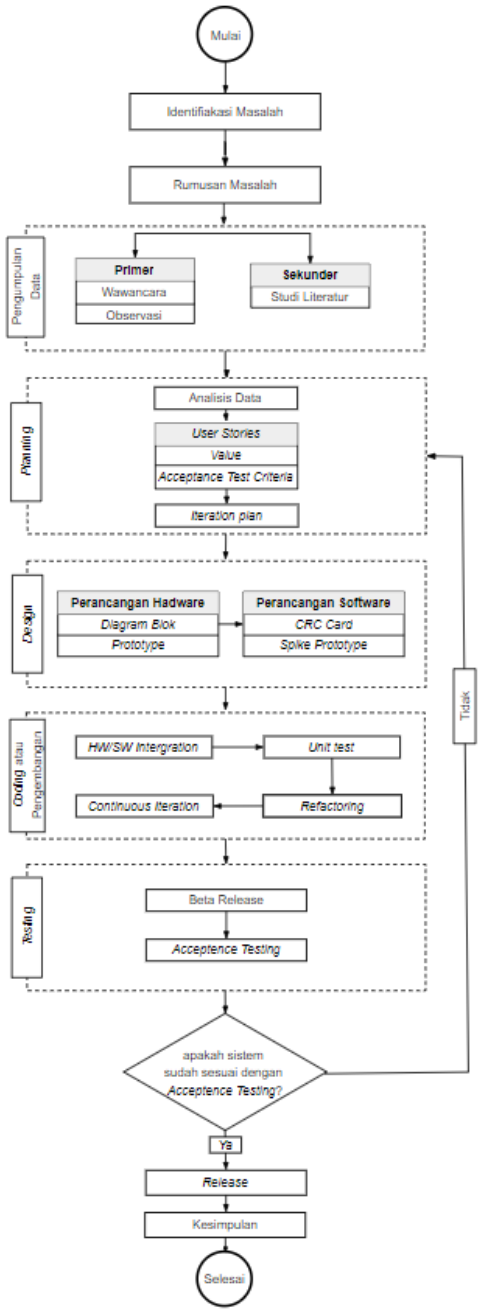
Berikut adalah tahapan dalam pengembangan *Extreme Programming*:

1. Planning (Perencanaan): Tahap ini meliputi penentuan kebutuhan pengguna, perencanaan iterasi, penjadwalan proyek, dan perencanaan pengujian.
2. Design (Desain): Desain dalam XP dilakukan secara kolaboratif oleh tim pengembang dan pengguna. Tujuan dari tahap ini adalah untuk menciptakan desain yang mudah dimengerti dan mudah diubah.
3. Coding (Kode): Tahap ini melibatkan pembuatan kode secara iteratif dan inkremental. Kode diuji dan diintegrasikan dalam setiap iterasi.
4. Refactoring (Refaktor): Refactoring dilakukan untuk memperbaiki desain dan kode yang sudah ada. Tujuan dari tahap ini adalah untuk membuat kode menjadi lebih mudah dimengerti, mudah diubah, dan efisien.
5. Testing (Pengujian): XP mendorong pengujian yang dilakukan secara otomatis dan kontinu. Setiap perubahan yang dilakukan pada kode diuji secepat mungkin untuk menemukan kesalahan dengan cepat.
6. Release (Rilis): Setiap iterasi harus menghasilkan sebuah produk yang siap dirilis. XP mendorong rilis sering dan cepat untuk memperoleh umpan balik dari pengguna.

3 METODE PENELITIAN

Metodologi penelitian dan pengembangan yang digunakan penulis adalah metode pengembangan *Extreme Programing* yang yang fokus pada pengembangan perangkat lunak dengan cepat, fleksibel, dan adaptif. Berikut adalah alur penelitian yang dilakukan oleh penulis dapat di lihat pada Gambar 3.

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023



Gambar 3. Alur Penelitian

4 HASIL DAN PEMBAHASAN

Dalam proses pengembangan sistem monitoring jaringan dan server berbasis *embedded system* ini menggunakan metode extreme programming seperti yang telah dipaparkan sebelumnya pada diagram alir penelitian pada Gambar 3.

4.1 Identifikasi Masalah

Pada tahapan ini dilakukan identifikasi masalah yang sedang terjadi pada objek penelitian. CV. Intek Solusindo sudah memiliki klien di beberapa Kota diantaranya Bandung Raya, Jakarta, Tanggerang dan Bekasi, yang tentunya memiliki banyak perangkat seperti *server*, *switch*, *router* dan *environment* lainnya yang harus selalu dipantau performa dan ketersediaan layanannya sehingga dapat mempercepat proses penanganan terhadap gangguan yang terjadi pada perangkat jaringan yang dimiliki.

4.2 Tahapan Pengumpulan Data

Untuk menyelesaikan masalah yang dihadapi maka harus dilakukan inventarisir data-data yang memungkinkan dengan melakukan wawancara dan mendengarkan kesulitan dan keinginan oleh user dalam hal ini admin jaringan. Selain wawancara dan mendengarkan (*Listening*) apa yang dinginkan oleh user, observasi juga dilakukan secara langsung dengan mengamati kondisi di lapangan dan sistem monitoring jaringan, server yang sedang berjalan di CV. Intek Solusindo. Kemudian Berdasarkan hasil wawancara maka dibuat *user stories* untuk digunakan sebagai acuan untuk membuat fitur-fitur yang harus ada pada sistem monitoring jaringan dan *server* berbasis *embedded system*.

Tabel 1. *User Stories*

Kode Stories	***User Stories (US)***
US-01	Sebagai admin jaringan, saya ingin memiliki *server* yang bisa me-monitoring perangkat jaringan, *server* dan environment lainnya.
US-02	Sebagai admin jaringan, saya ingin memiliki dashboard monitoring jaringan terpusat berbasis website.
US-03	Sebagai admin jaringan, saya ingin setiap perangkat baik router, switch, *server* termonitoring statusnya apakah *down* atau tidak
US-04	Sebagai admin jaringan, saya ingin memiliki sistem yang bisa memberikan informasi secara real-time jika terjadi kesalahan atau suatu

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

Kode Stories	*User Stories (US)*
	perangkat mengalami *down* dan memberikan informasi melalui email atau platform lainnya.
US-05	Sebagai admin jaringan, saya dapat menambahkan atau menghapus perangkat jaringan yang akan di monitoring.
US-06	Sebagai admin. Jaringan saya ingin sistem dapat me-monitoring trafik dan menyimpan *history* trafik pada rentang waktu tertentu.
US-07	Sebagai admin jaringan, saya ingin sistem dapat mendeteksi adanya api atau asap jika terjadi *power fault* baik pada perangkat jaringan atau instalasi listrik sehingga bisa menyebabkan kebakaran. Sistem dapat memberikan informasi secara real-time melalaui pesan social media.
US-04	Sebagai admin jaringan, saya ingin memiliki sistem yang bisa memberikan informasi secara real-time jika terjadi kesalahan atau suatu perangkat mengalami *down* dan memberikan informasi melalui email atau platform lainnya.

4.3 Planning (Perencanaan)

Pada tahap perencanaan ini dimulai dari analisis kebutuhan sistem berdasarkan dari data yang telah diperoleh. Kemudian pada tahap ini menyusun *user stories*, *values dan acceptance criteria* dan *iteration plan* berdasarkan hasil wawancara dan *listening* yang telah dilakukan pada tahap sebelumnya pada Tabel 1 user stories.

Tabel 1. *User Stories, Acceptance Criteria,* dan *Value*

Kode *Stories*	*User Stories (US)*	*Acceptance Criteria*	*Value*
US-01	Sebagai admin jaringan, saya ingin memiliki *small server* yang bisa me-monitoring perangkat jaringan, *server* dan environment lainnya.	Terdapat *small server* monitoring dengan arsitektur ARM	High
US-02	Sebagai admin jaringan, saya ingin memiliki dashboard monitoring jaringan terpusat berbasis website.	Terdapat dashboard monitoring berbasis website	High
US-03	Sebagai admin jaringan, saya ingin setiap perangkat baik router, switch, *server* termonitoring statusnya apakah *down* atau tidak.	Sistem dapat me-monitoring status perangkat	High
US-04	Sebagai admin jaringan, saya ingin memiliki sistem yang bisa memberikan informasi secara real-time jika terjadi kesalahan atau suatu perangkat mengalami *down* dan memberikan informasi melalui email atau platform lainnya.	Terdapat notifikasi alert melalui social media atau platform lainnya.	Medium
US-05	Sebagai admin jaringan, saya dapat menambahkan atau menghapus perangkat jaringan yang akan di monitoring.	Sistem dapat menambahkan atau menghapus perangkat jaringan yang akan di monitoring	High

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

Kode *Stories*	***User Stories (US)***	***Acceptance Criteria***	***Value***
US-06	Sebagai admin. Jaringan saya ingin sistem bisa menyimpan histori trafik dan resource yang dimonitoring dalam jangka waktu yang panjang, sehingga jika dibutuhkan untuk menganalisa data terdahulu data tersebut masih ada.	Sistem dapat me monitoring trafik dan menyimpan data *history* trafik	High
US-07	Sebagai admin jaringan, saya ingin sistem dapat mendeteksi adanya api atau asap jika terjadi *power fault* baik pada perangkat jaringan atau instalasi listrik sehingga bisa menyebabkan kebakaran. Sistem dapat memberikan informasi secara realtime melalaui pesan social media.	Sistem dapat mendeteksi jika ada indikasi kebarakan pada ruangan *server*	Medium

Setelah memberikan *acceptance criteria* dan *value* pada setiap *user stories* kemudian dibuat *iteration planning* untuk setiap *user stories*.

Tabel 2. *Iteration Plan*

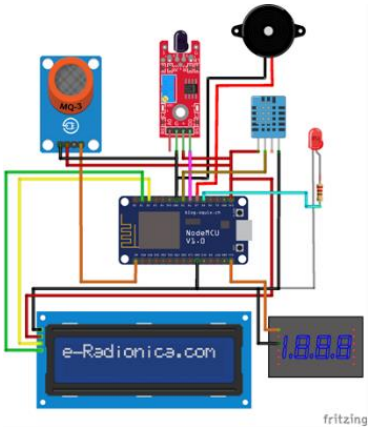
Iteration	**Kode *Stories***	***User Stories***	***Value***	***Estimasi (Hari)***
A-1	US-01	Sebagai admin jaringan, saya ingin memiliki *small server* yang bisa me-monitoring perangkat jaringan, *server* dan environment lainnya.	High	1
A-2	US-02	Sebagai admin jaringan, saya ingin memiliki dashboard monitoring jaringan terpusat berbasis website.	High	1
A-3	US-05	Sebagai admin jaringan, saya dapat menambahkan atau menghapus perangkat jaringan yang akan di monitoring.	High	2
	US-06	Sebagai admin. Jaringan saya ingin sistem dapat me-monitoring trafik dan menyimpan history trafik pada rentang waktu tertentu.	High	2
A-4	US-03	Sebagai admin jaringan, saya ingin setiap perangkat baik router, switch, *server* termonitoring statusnya apakah *down* atau tidak.	High	2
	US-04	Sebagai admin	Medium	2

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

Iteration	Kode *Stories*	*User Stories*	*Value*	*Estimasi (Hari)*
		jaringan, saya ingin memiliki sistem yang bisa memberikan informasi secara real-time jika terjadi kesalahan atau suatu perangkat mengalami *down* dan memberikan informasi melalui email atau platform lainnya.		
A-5	US-07	Sebagai admin jaringan, saya ingin sistem dapat mendeteksi adanya api atau asap jika terjadi *power fault* baik pada perangkat jaringan atau instalasi listrik sehingga bisa menyebabkan kebakaran. Sistem dapat memberikan informasi secara realtime melalaui pesan social media.	Medium	4

4.4 Design (Perancangan)

Pada tahap design terbagi menjadi beberapa tahapan yaitu perancangan *hardware* dan perancangan *software*. Pada proses perancangan perangkat keras menggunakan beberapa komponen perangkat keras yaitu: NodeMCU ESP 8266 sebagai pengendali utama untuk menjalankan proses pada pembacaan sensor MQ-2, *Flame* Sensor, DHT-11, dan komponen pendukung lainnya.



Gambar 4. Prototype NodeMCU Pendeteksi Kebakaran

Pada tahap perancangan peragkat lunak ini terbagi menjadi dua tahap, yaitu pembuatan *CRC cards* dan *spike solution prototype*.

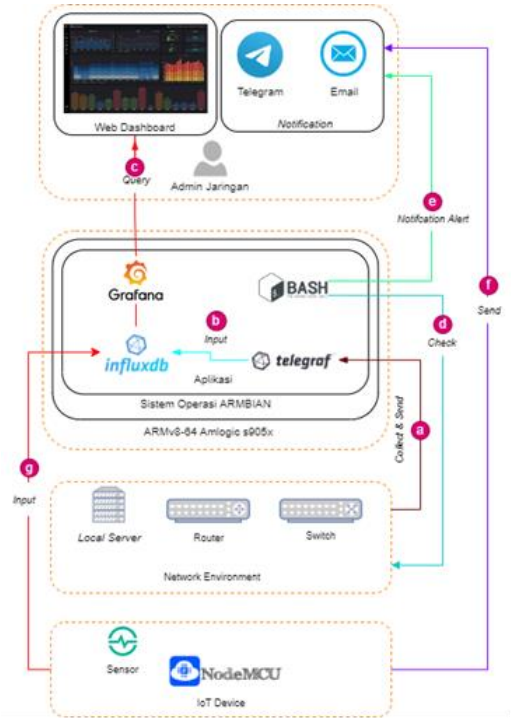
Tabel 4. CRC *Cards*

Class : Grafana	
Responbility:	**Collaborator:**
Menampilkan panel monitoring	Telegraf
	influxdb
Class : Dashboard	
Responbility:	**Collaborator:**
Menampilkan panel monitoring	Admin
	User
Class : Admin	
Responbility:	**Collaborator:**
Menampilkan halaman admin	Grafana
Mengatur tampilan panel monitoring	
Menambahkan user baru	
Menambahkan panel monitoring	Telegraf
Class : Telegraf	
Responbility:	**Collaborator:**
Me-monitoring perangkat jaringan resource dan environment	Influxdb
Class : Telegram	
Responbility:	**Collaborator:**
Mengirimkan notifikasi alert	Bash
	NodeMCU

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

Class : Sensor	
Responbility:	**Collaborator:**
Ambil data nilai	NodeMCU
Kirim notifikasi	Telegram
Class : Bash	
Responbility:	**Collaborator:**
Cek status perangkat	Perangkat jaringan
Kirim notifikasi	Telegram

Pada tahap spike solution prototype terbagi menjadi dua tahap perancangan arsitektur sistem monitoring dan antar muka sistem.

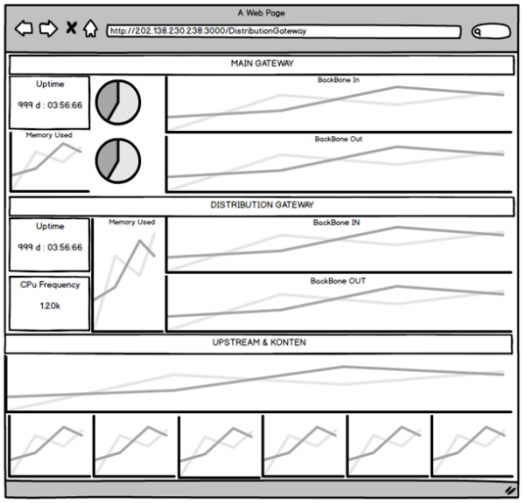


Gambar 5. Arsitektur Sistem Monitoring

Tabel 3. Deskripsi Alur Sistem

Kode	Deskripsi
A	Telegraf melakukan collect data metric berupa trafik, dan resource environment perangkat jaringan dari *server*, router dan switch.
B	Telegraf melakukan input data ke database time series influxdb.
C	Grafana melakukan query data ke influxdb dan menampilkan data berupa chart, metric dan lainnya.
D	Bash scripting akan melakukan pengecekan status perangkat jaringan apakah status perangkat down/up melalaui telegram.
E	Jika status perangkat down maka admin jaringan akan menerima notifikasi melalui email dan telegram.
F	IoT NodeMCU akan mengirimkan notifikasi alert melalaui telegram.
G	IoT NodeMCU megirimankan data metrics sensor ke database influxdb.

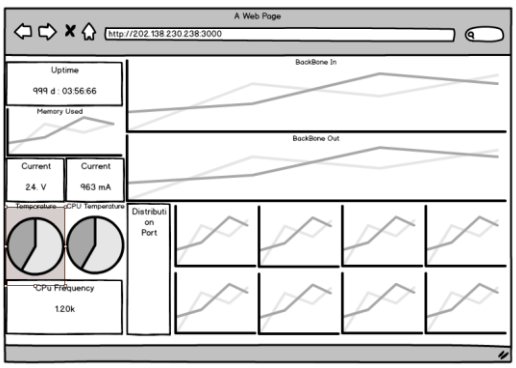
Perancangan Antrar Muka Sistem merupakan hal yang sangat penting karena yang pertama kali berinteraksi antara aplikasi dengan pengguna adalah antarmuka (*interface*). Berikut adalah perancangan antarmuka sistem monitoring jaringan dan *server* pada dashboard grafana.



Gambar 6. Main Dashboard

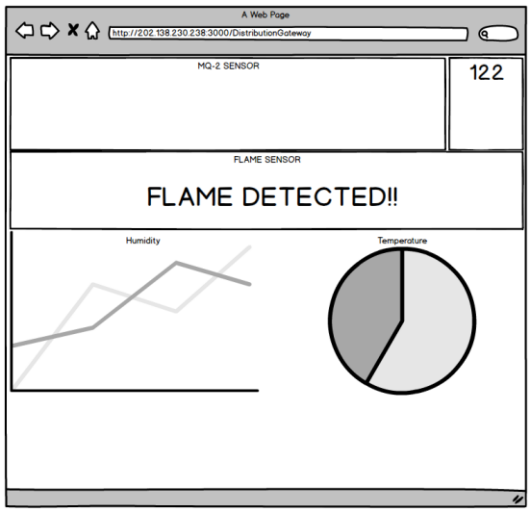
Pada Gambar 6 Merupakan prototype halaman main dashboard, halaman ini berisi tentang monitoring seluruh perangkat jaringan dan *server* yang dimonitoring baik *resource*, *trafik*, dan *environment*.

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023



Gambar 7. Main Gateway

Gambar 7 merupakan prototype halaman main gateway, halaman ini berisi tentang monitoring perangkat jaringan berupa router gateway yang dimonitoring baik *resource*, trafik, dan *environment* dari perangkat yang dimonitoring.



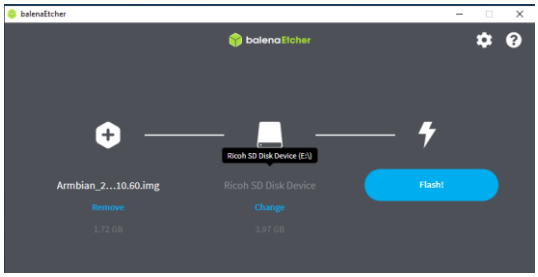
Gambar 8. IoT Dashboard

Halaman IoT merupakan halaman berisi tentang status sensor NodeMCU berupa DHT11, *Flame* Sensor dan MQ-2 Sensor.

4.5 Coding (Pengembangan)

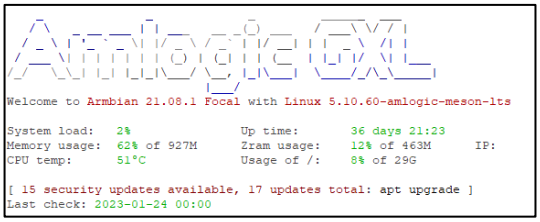
Pada tahap ini dilakukan pengembangan sistem berdasarkan *iteration plan* yang sudah di buat pada Tabel 3.

Pada tahap iterasi A-1 dilakukan instalasi dan konfigurasi sistem operasi Armbian 21 pada *Embedded System* Amlogic S905X ARM 64, tools yang dibutuhkan adalah Board Amlogic S905X, Memory SD-card 32Gb *class* 10, *BalenaEtcher*, dan sistem operasi Armbian 21, dan laptop.



Gambar 9. Instalasi Armbian 21 Dengen BalenaEtcher

Seteleh *SD-Card* terhubung dengan board *amlogic* S905X kemudian hubungkan *adaptor* catu daya ke board *Amlogic* S905X dan tunggu proses *booting* sampai selesai, jika proses *booting* selesai maka akan di minta untuk memasukan *username* dan *password* diawal.



Gambar 10. Terminal Banner Amlogic GXL

Kemudian jika sistem operasi Armbian sudah berjalan dengan normal dilakukan konfigurasi IP *Address* dan OpenSSH-Server agar dalam melakukan konfigurasi selanjutnya menggunakan remote ssh dengan putty.

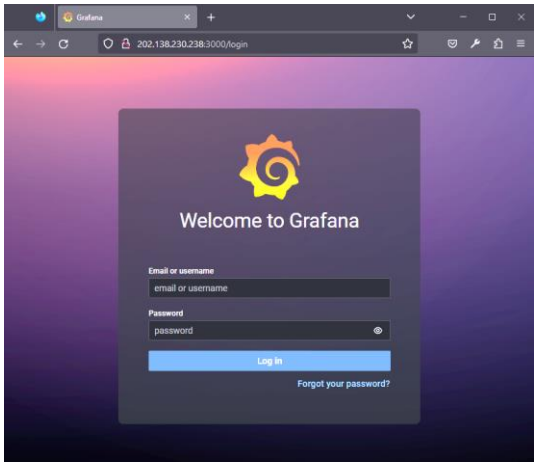
Pada tahap iterasi A-2 dilakukan instalasi dan konfigurasi *dashboard grafana* dan *dependency* web*server* Apache HTTP server untuk mengintsall package HTTP dengan memasukan perintah "apt install –y apache2" dan tunggu proses sampai selesai. Jika proses instalasi sudah selesai kemudian cek status service apache dengan perintah "systemctl status apache2.service" apakah *service running* atau tidak. Selanjutnya adalah tahap instalasi grafana *dashboard* dengan menggunakan *bash script*.

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

```
apt install wget curl gnupg2 apt-transport-https software-
properties-common -y
wget -q -O - https://packages.grafana.com/gpg.key | apt-key add -
echo "deb https://packages.grafana.com/oss/deb stable main" |
tee -a /etc/apt/sources.list.d/grafana.list
apt update -y
apt install grafana -y
echo "Selesai"
```

Gambar 11. Bash Script Install Grafana

Secara default *grafana-server* berjalan di port 3000 untuk mengecek port tersebut menggunakan perintah "ss -antpl | grep 3000" kemudian melakukan pengujian dengan mengakses service grafana menggunakan browser dengan memasukan IP Adress Server dan port grafana. Contoh 202.138.230.238:3000.



Gambar 12. Tampilan Login Grafana

Pada tahap iterasi A-3 dilakukan instalasi dan konfigurasi *time series database* influxDB dan telegraf menggunakan *bash script*.

```
wget -qO- https://repos.influxdata.com/influxdb.key | sudo apt-
key add -
source /etc/lsb-release
echo "deb https://repos.influxdata.com/${DISTRIB_ID,,}
${DISTRIB_CODENAME} stable" | sudo tee
/etc/apt/sources.list.d/influxdb.list
sudo apt update
sudo apt install influxdb
```

Gambar 13. Bach Script Install InfluxDB dan Telegraf

Setelah proses instalasi selesai cek status influxdB dan telegraf apakah *running* atau tidak dengan perintah "service influxdb status" dan cek juga service telegraf dengan perintah "service telegraf status" agar sistem dapat memonitoriing perangkat jaringan dalam hal ini sebuah router mikrotik maka dalam konfigurasi pada router mikrotik harus mengaktifkan service SNMP pada router tersebut kemudian set SNMP *Trap Community*. Tahap selanjutanya ialah pembuatan database pada InfluxDB guna untuk menyimpan data metric monitoring.

```
root@amlogic:/home/alif# influx
Connected to http://localhost:8086 version 1.8.10
InfluxDB shell version: 1.8.10
> show databases
name: databases
name
----
_internal
> create database telegraf
> show databases
name: databases
name
----
_internal
telegraf
```

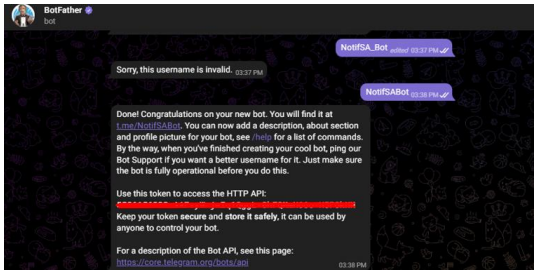
Gambar 14. Perintah Query Pembuatan Database

Kemudian tahap selanjutnya pada iterasi ini adalah mengintegrasikan influxDB dan telegraf agar nantisnya hasil SNMP dari telegraf akan di simpan pada databse influxDB yang sudah dibuat pada tahap sebelumnya.

Gambar 15. Konfigurasi Telegraf dan InfluxDB

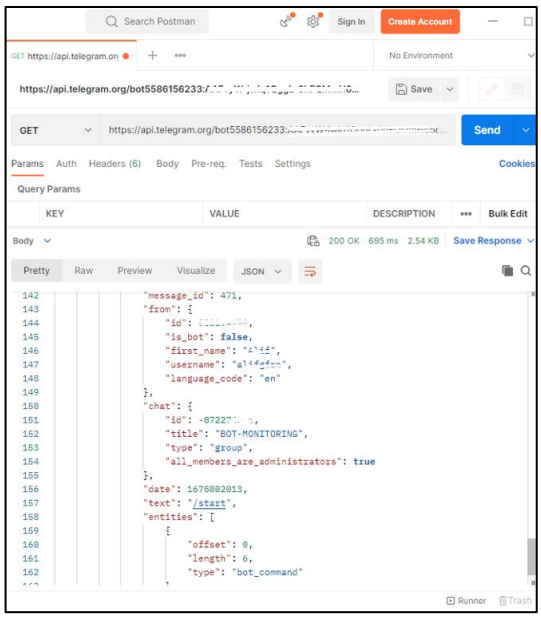
Pada tahap iterasi A-4 dilakukan pembuatan Bot Telegram melalui @botfather untuk memberikan notifikasi *alert* melalui telegram.

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023



Gambar 16. Proses Pembuatan Bot Telegram

Tahap selanjutnya adalah membuat grup yaitu "BOT-MONITORING" agar nantinya informasi alert akan dikirimkan melalui grup tersebut. Setelah proses pembuatan grup selesai selanjutnya pada chat room grup tersebut memberikan perintah "/start" guna mengecek apakah bot sudah masuk ke dalam grup dan pesan tersebut sudah terkirim atau belum, tahap berikutnya melakukan pengujian API dengan endpoint "/getUpdates" untuk mendapatkan group id chatroom.



Gambar 17. Pengujian Dengan Postman

Selanjutnya adalah tahap membuat bash script pada server utnuk mengirimkan notifikasi *alert.*

```
#!/usr/bin/bash
GROUP_ID="-8722"
BOT_TOKEN=5586156233:AAFvyW-jwIq1QggL
STATUS=$1
PESAN=$2

#if [ "$1" == "-h" ]; then
#  echo "Usage: `basename $0` \"text message\""
#  exit 0
#fi

if [ -z "$1" ]
  then
    echo "Tambahkan Pessan di argument ke-2"
    exit 0
fi

#### STATUS
if [ $1 == "DOWN" ];then
  STATUS="%E2%9D%8C+DOWN+%E2%9D%8C%0A============"
fi

if [ $1 == "failed" ];then
  STATUS="%E2%9D%8C+FAILED+%E2%9D%8C%0A============"
fi

if [ $1 == "success" ];then
  STATUS="%e2%9c%85+SUCCSESS+%e2%9c%85%0A=============="
fi

if [ $1 == "start" ];then
  STATUS="%F0%9F%94%84++START++%F0%9F%94%84%0A============"
fi

if [ $1 == "finish" ];then
  STATUS="%e2%9c%85+FINISH+%e2%9c%85%0A============"
fi

if [ $1 == "mailq" ];then

STATUS="%e2%9d%97+%e2%9a%a0+MAILQ+%e2%9a%a0+%e2%9d%97%0
A================"
fi

#####endstatus
curl -s -d "text=%0A$STATUS%0A$PESAN" -d "chat_id=$GROUP_ID"
'https://api.telegram.org/>

echo ''
```

Gambar 18. Script Bot Telegram

Tahap selanjutnya adalahpembuatan script guna memonitoring status perangkat apakah statusnya *Up* atau *Down* dan jika perangkat tersebut *Down server* akan memberikan Notifikasi melalui *Telegram*.

```
#HOSTS="8.8.8.8 google.com 172.16.1.2 172.16.1.3 172.16.1.4"
COUNT=1
for myHost in $HOSTS
do
  count=$(ping -c $COUNT $myHost | grep 'received' | awk -F',' '{
print $2 }' | awk '{ print $1 }')
  if [ $count -eq 0 ]; then
    echo "Host : $myHost is down (ping failed) at $(date)"
    bot-telegram DOWN $myHost
  fi
done
```

Gambar 19. Script Monitoring Ping

Pada iterasi A-5 ini dilakukan instalasi *hardware* IoT Device yaitu NodeMcu ESP 8266 dan modul yang sebelumnya sudah

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

dirancang sesuai *schematic* diagram. Setelah proses wiring pada tiap module selanjutnya adalah tahap pengembangan sistem dengan menggunakan Arduino IDE.

Gambar 20. Proses *Development* IoT NodeMCU

Jika tahap *development* selesai dan pengujian *code* tidak terdapat *error* maka tahap selanjutnya adalah melakukan *refactoring* terhadap *code* agar mendapatkan *clean code*.

```
void loop() {
  delay(2000);
  float h = dht.readHumidity();
  float t = dht.readTemperature();
  mq2Value = analogRead(MQ2PIN);
  flameValue = digitalRead(FLAMEPIN);

  if (isnan(h) || isnan(t)) {
    Serial.println(F("Gagal membaca sensor DHT11"));
    lcd.clear();
    lcd.setCursor(0, 0);
    lcd.print("Gagal membaca sensor DHT11");
    return;
  }

  //heat index in celcius
  float hic = dht.computeHeatIndex(t, h, false);

  Point sensor("pop");
  sensor.addTag("node_name", "BPI");
  sensor.addField("flame",flameValue );
  sensor.addField("humidity", h);
  sensor.addField("mq-2", mq2Value);
  sensor.addField("temperature", t);
  client.writePoint(sensor);

  if (!client.writePoint(sensor)) {
    Serial.print("InfluxDB write failed: ");
    Serial.println(client.getLastErrorMessage());
  }

  Serial.println("Delay 1s");
  delay(1000);

  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Humidity: ");
  lcd.print(h);
  lcd.print("%");
  Serial.print("Humidity: ");
  Serial.print(h);
  lcd.setCursor(0, 1);
  lcd.print("Temperature: ");
  lcd.print(t);
```

Gambar 21. Soource *Code* Sebelum *Refactoring*

```
    lcd.print("Koneksi Jelek");
    delay(1000);
  }

  //kirim pesan pertama
  myBot.sendMessage(id, "This Message From NodeMCU ESP8266");
  Serial.println("Pesan Ke Telegram Terkirim");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Pesan Ke Telegram Terkirim");
  delay(1000);

  //Synx TIME
  timeSync(TZ_INFO, "pool.ntp.org", "time.nis.gov");

  //Check Connection InfluxDB
  if (client.validateConnection()) {
    Serial.print("Connected to InfluxDB: ");
    Serial.println(client.getServerUrl());
  } else {
    Serial.print("InfluxDB connection failed: ");
    Serial.println(client.getLastErrorMessage());
  }

  delay(2000);

  Serial.println("Setup Complete.");
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Setup Complete.");
  lcd.clear();
}


void loop() {
  delay(2000);

  sensorDHT();
  fireDetect();
  displayLcd();
  influxQuery();
}
```

Gambar 22. Soruce *Code* Setelah *Refactoring*

4.6 Testing (Pengujian)

Pada tahapan ini sistem dilakukan small release untuk menguji *acceptance testing*. Pengujian ini dilakukan untuk menguji fungsionalitas sistem berdasarkan *acceptance criteria* dan *user stories* yang sebelumnya sudah dipaparkan oleh user. Pengujian ini dilakukan agar tidak ada terjadi kesalahan, Berikut hasil pengujian *acceptance testing* pada Tabel 6.

Tabel 6. *Acceptence Testing*

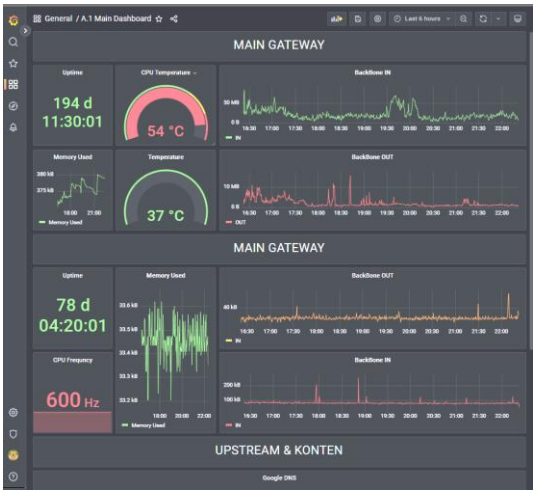
No	*Test Case*	*Acceptance Testing*	Status
1	Login Remote SSH *embedded server* armbian.	Terdapat *small server* monitoring dengan arsitektur ARM	Pass
2	Login Grafana Dashboard	Sebagai admin jaringan, saya ingin memiliki dashboard monitoring jaringan terpusat berbasis website.	Pass

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

No	*Test Case*	*Acceptance Testing*	Status
3	Lihat detail monitoring perangkat	Sistem dapat me-monitoring status perangkat	Pass
4	Cek status Up/Down dengan disable IP *Address* perangkat pada router	Terdapat notifikasi alert melalui social media atau platform lainnya.	Pass
5	Menambahkan perangkat baru dengan mmasukan Ip *Address* router/*server* pada telegraf.conf	Sistem dapat menambahkan atau menghapus perangkat jaringan yang akan di monitoring	Pass
6	Menambahkan user pada dashboard Grafana dan pada *server*.	Sistem dapat mengelola hak akses pengguna baik sisi *server* dan dashboard.	Pass
7	Cek histori trafik dan performance lainnya di 2 hari ke belakang dan cek query sample database terakhir.	Sistem dapat me monitoring trafik dan menyimpan data *history* trafik	Pass
8	Pengujian dengan melakukan pembakaran di dekat sensor	Sistem dapat mendeteksi jika ada indikasi kebarakan pada ruangan *server*	Pass

4.7 Release (Peluncuran)

Pada tahapan ini, dilakukan peluncuran sistem monitoring jaringan dan server secara terpusat berbasis Embedded System yang telah diuji dari fungsionalitas dan sesuai dengan acceptance test. Peluncuran sistem akan dilakukan langsung pada server yang sudah di deployment dan memiliki IP Address publik agar dapat di akses secara online. Berikut link monitoring http://IP-Server:3000.



Gambar 22. Halaman Main Dashboard

Halaman main dashboard adalah halaman yang menampilkan beberapa visualisasi dari beberapa *dashboard* diantaranya: main *gateway* berupa *uptime*, *memory CPU temperature*, *temperature*, *metrics trafik input* dan *metrics trafik output*. Pada distribution gateway menampilkan berupa uptime, CPU frequency, memory used, upstream konten *metrics trafik input* dan *metrics trafik output*. Pada *upstream* dan konten menampilkan status Ping terhadap *upstream* atau sumber internet pada router utama dan status ping terhadap beberapa server di internet seperti google.com.

Gambar 23. Halaman Main Gateway

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

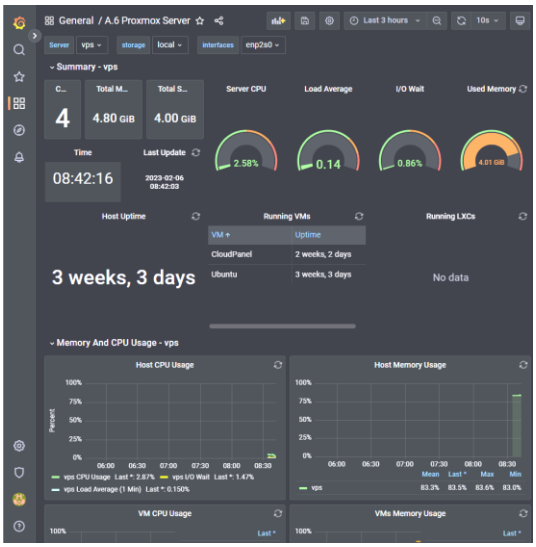
Halaman main gateway menampilkan *environment* dari *router gateway* utama yaitu *router* mikrotik CCR1016-12G baik *uptime*, *memory used*, *voltage*, *current*, *temperature*, CPU *temperature* dan CPU *frequency*. Selain menampilkan dan memonitoring environment pada *dashboard* main gateway juga me-monitoring penggunaan *traffic* seluruh *port Ethernet* pada *router Mikrotik.*

Gambar 24. Halaman Distribution Gateway

Pada halaman distribution gateway ini menampilkan *environment* router mikrtoik RB950Ah seperti *uptime*, CPU *Frequency*, *memory used*. Selain menampilkan *environment* pada router di *halaman dashboard distribution gateway* menampilkan metrics penggunaan trafik pada port *Ethernet* dan *port Virtual Local Area Network* (VLAN).

Gambar 25. Halaman Status Ping

Pada halaman status adalah menampilkan status ping terhadap beberapa host WAN (*Wide Area Network*) ini termasuk ping status Provider CBN sebagai sumber internet dan melakukan ping terhadap beberapa konten internet seperti *google*, *cloud flare*, *amazon*. Sedangkan pada bagian bawah terdapat dua monitoring status ping yaitu jaringan *local* dan jaringan VPC (*Virtual Private Cloud*) Network. VPC merupakan jaringan virtual yang dibangun dan berjalan diatas layanan cloud computing.



Gambar 26. Halaman Monitoring VPS Server

Pada halaman dashboard monitoring proxmox *server* ini menampilkan seluruh environment monitoring dari proxmox baik *uptime* server, jumlah *memory*, jumlah *swap jumlah* CPU, load server dan jumlah *Virtual machine* (VM) yang aktif, selain me-monitoring environment pada server, pada dashboard ini juga me-monitoring trafik penggunaan bandwidth keseluruhan baik server dan Virtual Machine.

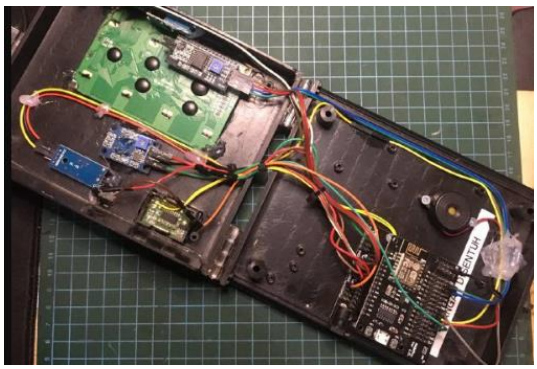
Gambar 27. Halaman Sensor IoT Device

Pada halaman dashboard IoT device menampilkan visualisasi dari 3 sensor yaitu sensor MQ-2 untuk melihat nilai *threshold* dalam mendeteksi adanya asap atau kebocoran gas. Sensor Flame akan menampilkan dua kondisi yaitu normal jika tidak mendeteksi adanya api dan *flame detect*

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

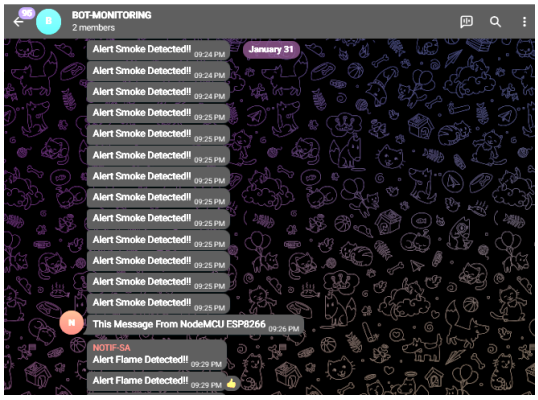
jika sensor mendeteksi adanya api. Untuk sensor DHT11 memilki dua nilai yaitu *humidity* membaca kelembapan udara pada ruangan dan *temperature* membaca suhu ruangan.

Gambar 28. IoT Device NodeMCU ESP 8266 Box

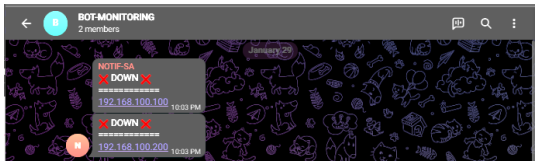


Gambar 29. IoT Device NodeMCU ESP8266 Wiring

Pada Gambar 28 merupakan tampilan luar dari perangkat IoT dengan menampilkan nilai sensor melalui LCD 20x4, terlihat pada gambar tersebut untuk sensor flame, sensor MQ-2 dan sensor DHT11 terdapat di bagian luar box guna untuk mendeteksi adanya kebakaran langsung tanpa tertutup benda lain sedangkan pada Gambar 29 adalah rangkaian elektronik, modul sensor dan ESP8266.



Gambar 30. Notification Alert Sensor IoT via Telegram



Gambar 31. Notification Alert Status Host

Pada Gambar 30 hasil dari monitoring sensor jika sensor mendeteksi adanya api atau asap maka NodeMCU ESP8266 sedangkan Gambar 31 jika salah satu host yang dimonitor statusnya mati/*down* maka script bash akan mengirimkan pemberitahuan kepada admin jaringan melalui telegram.

5 SIMPULAN

Berdasarkan hasil perancangan, implementasi dan pengujian terhadap sistem monitoring jaringan dan server, maka dapat diambil beberapa kesimpulan sebagai berikut:

1. Untuk perancangan, implementasi dan pengujian pada sistem monitoring jaringan dan server secara terpusat berbasis Embedded System menunjukan hasil yang sesuai dengan user stories ini dibuktikan dengan hasil pengujian sesuai dengan Tabel 6, sehingga secara fungsional sistem dapat digunakan.
2. Server sehingga data dapat divisualisasikan melalui dashboard grafana sesuai dengan Gambar 22
3. Notifikasi secara real-time yang dikirim melalui telegram kepada admin jaringan berjalan dengan baik di buktikan pada tahap pengujian Tabel 6 serta Gambar 30

e-ISSN: 2580-3239
p-ISSN: 2085-7993
Volume 22 No. 01 April 2023

dan 31 Notification Alert IoT via Telegram.

4. Sensor IoT yang disimpan di ruangan server dapat mendeteksi indikasi kebakaran dibuktikan dengan tahap pengujian pada Tabel 6 dan Gambar 27.

DAFTAR PUSTAKA

Bayu, D. (2022, Juni 10). *APJII: Pengguna internet Indonesia Tembus 210 Juta pada 2022*. Dataindonesia.id. https://dataindonesia.id/digital/detail/apjii-pengguna-internet-indonesia-tembus-210-juta-pada-2022

Beck, K. (2005). *Extreme Programing Explained, Second Edetion*. Boston: John Wait.

Douglass, B. P. (2011). *Design Patterns for Embedded Systems in C.* USA: Elsevier Inc.

Miftah, Z. (2019). Penerapan Sistem Monitoring Jaringan dengan Protokol SNMP pada Router Mikrotik dan Aplikasi Dude Studi Kasus STIKOM CKI. *Faktor Exacta*, *12*(1), 58-66.

Prayogi, P. K., Orisa, M., & Ariwibisono, F. X. (2020). Rancang Bangun Sistem Monitoring Jaringan Access Point Menggunakan Simple Network Management Protocol (SNMP) Berbasis Web. *JATI (Jurnal Mahasiswa Teknik Informatika)*, *4*(1), 192-197.

Pressman, R. S., & Maxim, R. B. (2020). *Software Engineering A Practitioner's Approach 9th Edition*. New York: McGraw-Hill.

Rosa, A., & Salahudin, M. (2019). *Rekayasa Perangkat Lunak Terstruktur dan Berorientasi Objek.* Bandung: Informatika.

Sasmoko, D., & Mahendra, A. (2017). Rancang bangun sistem pendeteksi kebakaran berbasis iot dan sms gateway menggunakan arduino. *Simetris: Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer*, *8*(2), 469-476.