https://bufnets.tech
https://doi.org/10.59688/bufnets

BULLETIN OF NETWORK ENGINEER AND INFORMATICS
E-ISSN 2986-8017 | P-ISSN 2987-4858
Vol. 1 No 1 – April 2023

APLIKASI BERBASIS WEB GUNA MEMONITORING KE AKTIFAN IP PUBLIC

WEB-BASED APPLICATIONS FOR MONITORING PUBLIC IP ACTIVITIES

Ismail Puji Saputra[1], Danang Prabowo[2]

Universitas Muhammadiyah Metro
Jl. Ki Hajar Dewantara No.116, Iringmulyo, Kec. Metro Tim., Kota Metro, Lampung, Indonesia

e-mail: ismailpujisaputra@gmail.com[1], danang32@gmail.com[2]

Received : 16 February 2023	Accepted : 18 February 2023	Published : 20 April 2023

Abstract

Web servers, internet servers, and network devices are the backbone of information systems, the industrial era 4.0 made information systems inseparable from people's lives, all aspects of people's lives related to information systems need internet networks and Public IPs are used to access an information system, in an institution, it is not easy to do troubleshooting if a problem occurs, for this reason, it is necessary to build a notification system if a server down occurs so that the information system manager knows there has been a disturbance that causes the server to go down and cannot be accessed. The purpose of this research is to build a web-based application that can provide notifications when a Public IP is down, the web server application will be placed outside the network so that it can show that the Public IP cannot be accessed outside the network. The method for building applications will use the extreme programming (XP) method which starts from the process of planning, designing, coding, and testing. The results of this study are a website that can provide notifications if there is a disturbance via Telegram. From the testing process carried out from 10 trials, the application has 100% sentitivity in providing notifications when a server-down incident occurs.

Keywords: *Telegram, Notification, Server Down, System Monitoring*

Abstrak

Web servers, internet server dan perangkat jaringan menjadi tulang punggung di dalam sistem informasi, era industri 4.0 menjadikan sistem informasi tidak dapat di pisahkan dari kehidupan masyarakat, segala aspek kehidupan masyarakat yang berhubungan dengan sistem informasi sangat membutuhkan jaringan internet dan IP Public yang digunakan untuk mengakses sistem informasi, pada suatu instansi tidak mudah untuk melakukan troubleshoot apabila terjadi masalah, untuk itu perlu di bangun sebuah sistem notifikasi apabila terjadi insiden server down sehingga administrator sistem informasi tersebut mengetahui telah terjadi gangguan yang menyebabkan server down dan tidak dapat diakses. Tujuan dari penelitian ini adalah untuk membangun aplikasi berbasis web yang dapat memberikan notifikasi apabila terdapat IP Public yang down, webserver aplikasi akan di taruh di luar jaringan sehingga dapat menunjukan bahwa IP Public tersebut tidak dapat di akses melalui luar jaringan. Metode dalam membangun aplikasi akan menggunakan metode extreme programming (XP), yaitu di mulai dari proses planning, design, coding dan testing. Hasil dari penelitian ini adalah sebuah website yang dapat memberikan notifikasi apabila terdapat gangguan melalui Telegram. Dari proses pengujian yang di lakukan dari 10 kali percobaan, aplikasi memiliki sensitivitas 100% dalam memberikan notifikasi ketika terjadi insiden server down.

Kata Kunci: *Telegram, Notifikasi, Server Down, Monitoring Sistem*

All articles published in BUFNETS will be immediately and permanently free for everyone to read and download. Under the CC-BY license

1. PENDAHULUAN

Monitoring sebuah server maupun perangkat jaringan yang dapat di akses publik sangat perlu untuk di monitoring [1], sistem di tuntut dapat diakses kapan saja dan di mana saja [2], pentingnya memonitoring jaringan adalah untuk mengetahui permasalahan sejak dini, sehingga administrator sistem dapat segera menindaklanjuti notifikasi tersebut[3] agar sistem dapat bekerja kembali setelah mengalami *downtime*.

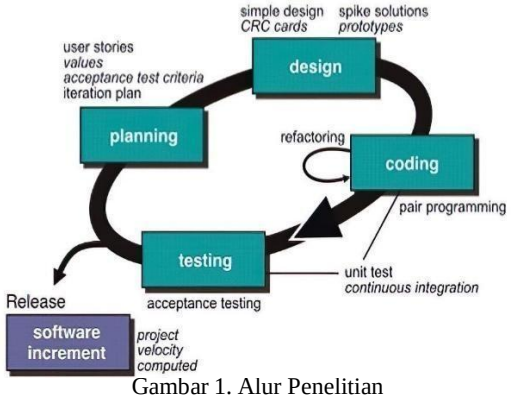
Server memiliki *IP Public* yang merupakan alamat dari *server* tersebut yang di gunakan untuk proses transaksi data pada jaringan internet, alamat *IP Public* ini akan di manfaatkan untuk di monitoring, mekanisme menggunakan *internet control message protocol (ICMP) packet* [4], yaitu dengan melakukan *ping* ke IP Public, apabila *ping* berhasil di lakukan maka *host* dianggap *ON* namun apabila *server* tidak merespon packet *ICMP* maka dianggap *host* tersebut *down* [5].

Monitoring pada jaringan dengan memanfaatkan fitur *ping* telah berhasil di lakukan, di antaranya penelitian [1] berhasil memafaatkan fitur *netwatch* pada MikroTik yang memanfaatkan *ping* kepada *IP Address* yang telah di daftarkan dan akan memberikan notifikasi menggunakan *Bot API (Application Programming Language)* Telegram. Namun aplikasi yang di bangun masih dapat mengalami kendala apabila koneksi internet terputus maka mekanisme notifikasi tidak akan berjalan, karena jaringan internet yang putus sangat di butuhkan untuk mengirimkan pesan melalui *API* Telegram, untuk itu penelitian ini akan mengajukan penempatan *server* pada jaringan publik *(cloud)* sehingga tidak akan tergantung pada koneksi internet yang ada pada jaringan yang di monitoring.

Penelitian ini akan membangun sebuah aplikasi berbasis *web* yang di gunakan untuk mendaftarkan *host* yang akan di monitoring, aplikasi akan melakukan mekanisme *pin*g untuk melihat status ke aktifan *host*, apabila *host down*, maka aplikasi akan mengirimkan notifikasi melalui telegram, yang membedakan dari penelitian [1], yaitu *platform* yang di gunakan dalam bentuk *web* bukan menggunakan *netwatch* MikroTik, selain itu topologi atau penempatan *server* monitoring yang berbeda, dimana pada penelitian [1] ditempatkan pada MikroTik yang ada di dalam jaringan, sedangkan penelitian ini akan di tempatkan pada *cloud server*.

2. METODE PENELITIAN

Metode penelitian menggunakan metode *extreme programming (XP), extreme programming* merupakan metode pembangunan *software* yang dapat digunakan untuk meningkatkan kualitas *software* dan dapat responsif *dengan kebutuhan yang di inginkan* [6][7]. langkah dalam melakukan *extreme programming* dapat di lihat pada Gambar 1.



Gambar 1. Alur Penelitian
[Sumber: Nurlaela, L., Sopian, A., & Alfiah, F.]

Penjelasan gambar 1:

a). *Planning:* Proses *planning* menggambarkan apa saja yang harus di lakukan di dalam melakukan pengembangang aplikasi monitoring *IP Public* berbasis *web*.

b). *Design:* Proses *design* meliputi *design* alur sistem dan *design* halaman antar muka pengguna.

c). *Coding:* merupakan proses implementasi *design* ke dalam kode, kode di bangun dengan bahasa pemrograman PHP dan database MySQL.

d). *Testing:* testing yang di lakukan untuk menguji apakah aplikasi yang di bangun telah sesuai dengan kebutuhan, selain itu testing akan di lakukan untuk menguji sensitivitas dari aplikasi dengan melakukan uji coba 10 kali. Berikut ini persamaan 1 yang digunakan untuk mengukur sensitivitas.

$$\text{Sensitivitas} = TP\ (TP+FN) \quad (1)$$

dimana *TP* adalah *TRUE POSITIVE* yaitu apabila terdapat *hos*t yang *offline* dan aplikasi memberikan notifikasi *offline*, *FN* adalah *FALSE NEGATIVE* yaitu apabila *host offline* namun aplikasi tidak memberikan notifikasi *offline*[2].

All articles published in BUFNETS will be immediately and permanently free for everyone to read and download. Under the CC-BY license

3. HASIL DAN PEMBAHASAN

Deskripsi Data

Data pada penelitian ini terdiri dari beberapa host, host ini merupakan beberapa IP Public yang ada pada instansi, setiap host memiliki beberapa variabel, berikut ini tabel 1 yang merupakan isi dari data yang akan di uji coba pada aplikasi monitoring.

Tabel 1: Tabel data

No.	Nama Variabel	Keterangan
1	IP	Berupa domain atau IP address
2	Nama	Nama host untuk identifikasi
3	Status	Status host ON/OFF

Data *IP* dan nama akan di masukan ke dalam *database* MySQL, namun variabel status di dapatkan dari proses *ping* yang di lakukan oleh *server*, sehingga apabila *host* tidak dapat di *ping* maka *host* tersebut berstatus *OFF* namun apabila dapat di *ping* maka status *host* menjadi *ON*.

Pembahasan

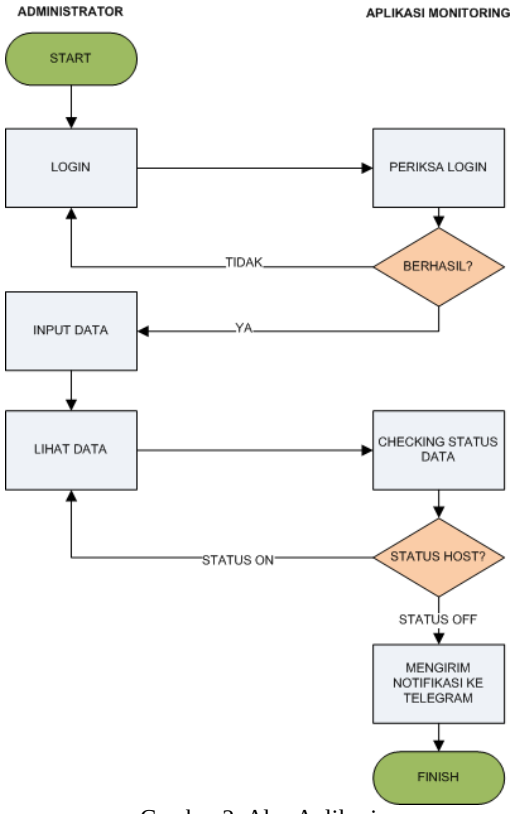
Proses *planning*, dilakukan dengan menganalisa kebutuhan apa saja yang di perlukan untuk membangun aplikasi monitoring, berikut ini tabel 2 yang berisi kebutuhan guna membangun aplikasi.

Tabel 2: Tabel kebutuhan aplikasi

No.	Kebutuhan software	Keterangan
1	Apache	Berfungsi untuk menjalankan webserver
2	MySQL	Berfungsi sebagai database untuk menampung data host (IP) dan nama host
3	WINDOWS 10	Berfungsi untuk menjalankan apache, PHP dan MySQL

Aplikasi yang di bangun berbasis *web* membutuhkan Apache sebagai *web server*, MySQL sebagai *database* dengan *OS* windows 10, maka dalam penelitian windows 10 akan menjalankan software XAMPP, aplikasi XAMPP yang dapat menjalankan Apache dan MySQL [9],[10],[11].

Tahap *design* di awali dengan menggambarkan bagaimana aplikasi web yang di bangun dapat bekerja, berikut ini gambar 2 yang menunjukan cara kerja dari aplikasi yang akan di bangun.



Gambar 2. Alur Aplikasi

Alur dari aplikasi di awali dengan administrator *login* ke sistem, sistem harus dapat melakukan *autentikasi*, apabila administrator berhasil *login* maka sistem akan mengarahkan ke halaman *form input data*, *data* yang di inputkan berupa *host (domain / IP address*) beserta nama dari *host* tersebut. administrator dapat memeriksa status dari *host*, status di dapatkan dari proses *ping*, *ping* dapat digunakan untuk memeriksa status ke aktifan suatu *host* [5],[12],[13], apabila *host* tidak merespon *ping* maka *host* di anggap *OFF* dan sistem akan memberikan notifikasi kepada administrator melalui Telegram.

Tahap *coding*, dilakukan dengan mengimplementasikan alur aplikasi ke dalam kode, di mana aplikasi yang di bangun dapat mengakomodir semua aktifitas yang di jelaskan pada alur aplikasi, untuk itu dalam proses *coding* beberapa gambaran kode semu *(Pseudo-Code)* di gunakan untuk menunjukan algoritma dari program yang di bangun [14],[15]. Beberapa *pseudo-code* yang di tampilkan akan menggambarkan bagaimana proses *checking* status suatu *host* dan bagaimana cara aplikasi mengirimkan pesan melalui telegram apabila *host* tersebut *OFF*. Berikut ini gambar 3,

All articles published in BUFNETS will be immediately and permanently free for everyone to read and download. Under the CC-BY license

merupakan gambar *pseudo-code* dari proses checking status.

```
exec("ping -n 1 " . $host, $output, $result);
// PROSES CHECKING STATUS HOST
$send = [
        'text' => " "."$data[nama]"." "." "."$data[ip] DOWN!!!!"."",
        // ISI PESAN TELEGRAM UNTUK MENUNJUKAN HOST SEDANG OFF
        'chat_id' => '[redacted]'; //CHAT ID TELEGRAM
if ($result != 0) // JIKA PING GAGAL MAKA STATUS DI ANGAP OFF
    $status="<font color='red'>OFF</font>";
    file_get_contents("https://api.telegram.org/bot$token/sendmessage?"
    . http_build_query($send));// PROSES PENGIRIMAN PESAN DENGAN TELEGRAM API
else ($result == 0) // JIKA PING BERHASIL MAKA STATUS DI ANGGAP ON
    $status="<font color='green'>ON</font>";
```

Gambar 3. *Pseudo-code* proses *checking* dan notifikasi

Dari gambaran *pseudo-code* pada gambar 3, menjelaskan apabila host di *ping* sebanyak 1 kali dan tidak merespon maka status *host* diangap *OFF* dan aplikasi akan mengirimkan pesan menggunakan Telegram *API* ke administrator, sebaliknya apabila *ping* berhasil maka status *host* di anggap *ON*.

Proses *testing*, proses *testing* dilakukan untuk menguji apakah aplikasi telah berhasil memenuhi kebutuhan yaitu dapat menunjukan status dari *host* yang telah di tambahkan ke dalam aplikasi oleh administrator, selain itu *testing* di lakukan dengan mematikan satu *host* dan melihat apakah aplikasi dapat mengirimkan status *OFF* dari suatu *host* melalui Telegram. berikut ini gambar 4, menunjukan daftar status dari *host* yang telah di tambahkan ke aplikasi.

IP	NAMA	STATUS	AKSI
103.213.116.83	CPANEL	ON	EDIT
103.213.116.90	VIRTUALIZOR	ON	EDIT
103.213.116.89	FORTINET	ON	EDIT

Gambar 4. Daftar *host* dan status *host*

Gambar 4 menjelaskan bahwa host yang terdaftar dalam status *ON*, sehingga aplikasi tidak melakukan apapun, namun apabila terdapat *host* yang berstatus *OFF* maka aplikasi akan mengirimkan pesan melalui administrator, bahwa host tersebut *OFF*, seperti pada gambar 5 di bawah ini.

IP	NAMA	STATUS	AKSI
103.213.116.83	CPANEL	OFF	EDIT
103.213.116.90	VIRTUALIZOR	ON	EDIT
103.213.116.89	FORTINET	ON	EDIT

Gambar 5. *Host* cpanel *OFF*

Gambar 5 menunjukan bahwa *host* cpanel berstatus *OFF* sehingga aplikasi akan mengirimkan status host kepada administrator melalui Telegram, seperti di tunjukan pada gambar 6 di bawah ini.

Gambar 6. Notifikasi telegram status *host*

Proses *testing* dari aplikasi berhasil di lakukan dan aplikasi dapat berjalan sesuai dengan fungsinya yaitu mengirimkan pesan apabila terdapat *host* yang tidak merespon *ping* dari *server*, mekanisme dapat berjalan otomatis dengan memanfaatkan *cronjob* seperti pada penelitian [16] yang memanfaatkan *cronjob*. *Cronjob* di jalankan dengan interval 5 menit sekali, sehingga setiap 5 menit aplikasi akan melakukan *checking* status *host*.

Selain melakukan testing fungsi aplikasi, testing juga di lakukan pada sensitivitas dari aplikasi yang di bangun, dari 10 kali pengujian yang di lakukan dengan menonaktifkan beberapa *host*, notifikasi selalu menghasilkan informasi yang benar *(True Positive)*, sehingga apabila di hitung nilai sensitivitasnya menggunakan persamaan 1 maka hasil sensitivitasnya adalah 1 jika di persentasikan menjadi 100%.

All articles published in BUFNETS will be immediately and permanently free for everyone to read and download. Under the CC-BY license

4. KESIMPULAN

Hasil uji coba dari aplikasi yang di bangun telah berhasil menjalankan fungsi monitoring *host* yang memiliki *IP Public* dan mengirimkan pesan melalui telegram apabila terdapat *host* yang *down (OFF),* berbeda dengan penelitian [1] yang dapat mengawasi jaringan lokal *(localhost),* namun tidak dapat mengirimkan notifikasi apabila jaringan internet lokal mati, aplikasi ini berada di luar jaringan *(cloud)* sehingga dapat mengirimkan pesan melalui telegram. selain itu penelitian ini juga menguji sensitivitas dimana hasil uji menunjukan bahwa aplikasi sangat sensitif dengan nilai sensitivitas 100%.

PERNYATAAN PENGHARGAAN

Ucapan terimakasih kepada seluruh penulis yang di kutip pada penelitian ini, secara tidak langsung tulisan mereka telah berkontribusi pada penelitian ini, terimakasih juga kepada seluruh Staff Pusat Teknologi Informasi dan Komunikasi Universitas Muhammadiyah Metro yang telah memberikan masukan dan dukungan, sehingga penelitian ini selesai di lakukan.

DAFTAR PUSTAKA

[1] Hidayat, A., Saputra, I. P., & Bowo, A. (2022). Bot Monitoring Jaringan Pada BMT Mentari Lampung Timur Menggunakan Mikrotik Dan API Telegram. JTKSI (Jurnal Teknologi Komputer dan Sistem Informasi), 5(3).

[2] Saputra, I. P., Utami, E., & Muhammad, A. H. (2022, October). Comparison of anomaly based and signature based methods in detection of scanning vulnerability. In 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI) (pp. 221-225). IEEE.

[3] Calvo, I., Espin, A., Gil-García, J. M., Fernández Bustamante, P., Barambones, O., & Apiñaniz, E. (2022). Scalable IoT Architecture for Monitoring IEQ Conditions in Public and Private Buildings. *Energies*, *15*(6), 2270.

[4] Koo, H., Chae, J., & Kim, W. (2023). Design and Experiment of Satellite-Terrestrial Integrated Gateway with Dynamic Traffic Steering Capabilities for Maritime Communication. Sensors, 23(3), 1201.

[5] Irawan, D. (2017). Mesin Virtual Menggunakan VMWare untuk mengoptimalkan Jaringan Internet Guna Memfasilitasi Perkuliahan. MIKROTIK: Jurnal Manajemen Informatika, 6(2).

[6] Beck, K. (1999). Embracing change with extreme programming. Computer, 32(10), 70-77.

[7] Akhtar, A., Bakhtawar, B., & Akhtar, S. (2022). EXTREME PROGRAMMING VS SCRUM: A COMPARISON OF AGILE MODELS. International Journal of Technology, Innovation and Management (IJTIM), 2(2).

[8] Nurlaela, L., Sopian, A., & Alfiah, F. (2023). Umrah Registration System Using Extreme Programming Method Towards Worship Tourism. International Journal of Cyber and IT Service Management, 3(1), 1-10.

[9] Suryana, T. (2022). Instalasi Xampp Untuk Menjalankan Webserver Apache Dan Mysql.

[10] Noviana, R. (2022). Pembuatan Aplikasi Penjualan Berbasis Web Monja Store Menggunakan Php Dan MYSQL. *Jurnal Teknik dan Science*, *1*(2), 112-124.

[11] Xuan, S. T. Z., & Selvarajah, V. (2022, October). Web Shell Attack and Mitigation. In *2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon)* (pp. 1-5). IEEE.

[12] SADIKOVA, G., AMREEV, M., MANANKOVA, O., YAKUBOVA, M., YAKUBOV, B., MUKASHEVA, A., & SERIKOV, T. (2022). ANALYSIS AND RESEARCH OF TASKS FOR OPTIMIZING FLOWS IN MULTISERVICE NETWORKS BASED ON THE PRINCIPLES OF A SYSTEMS APPROACH. Journal of Theoretical and Applied Information Technology, 100(9).

[13] Pan, L., Yang, J., He, L., Wang, Z., Nie, L., Song, G., & Liu, Y. (2022). Your Router is My Prober: Measuring IPv6 Networks via ICMP Rate Limiting Side Channels. arXiv preprint arXiv:2210.13088.

[14] Irawan, M. D. (2022). Flowchart dan Pseudo-Code: Implementasi Notasi Algoritma dan Pemrograman. Media Sains Indonesia.

[15] Del Castillo, M., Hermosa, H., Astillo, P. V., Choudhary, G., & Dragoni, N. (2023, February). Software-Defined Network Based Secure Internet-Enabled Video Surveillance System. In Information Security Applications: 23rd International Conference, WISA 2022, Jeju Island, South Korea, August 24–26, 2022,

All articles published in BUFNETS will be immediately and permanently free for everyone to read and download. Under the CC-BY license

Revised Selected Papers (pp. 89-101). Cham: Springer Nature Switzerland.
[16] Islami, M. R. R. (2022). DETEKSI DINI SERANGAN PADA WEBSITE MENGGUNAKAN METODE ANOMALI BASED. JIKO (Jurnal Informatika dan Komputer), 5(3), 224-229.

All articles published in BUFNETS will be immediately and permanently free for everyone to read and download. Under the CC-BY license