



RANCANG BANGUN MONITORING *SERVER* SYSTEM BERBASIS TERMINAL LINUX DAN *WHATSAPP* GATEWAY STUDI KASUS: PT. XYZ

Choirulloh¹, Subandri^{2*}

¹Program Studi Sistem Informasi, STMIK Bani Saleh, choirulloh@gmail.com

²Program Studi Teknik Informatika, STMIK Bani Saleh, andrisubandri@gmail.com

ABSTRAK

Penelitian ini bertujuan untuk merancang system monitoring *server* berbasis terminal Linux dan *Whatsapp Gateway* untuk PT. XYZ. Tujuan dari sistem ini adalah untuk memantau kondisi *server* secara *real-time* dan memberikan notifikasi kepada admin jika terjadi masalah pada *server*. Metode yang digunakan dalam penelitian ini adalah dengan mengintegrasikan aplikasi terminal Linux dengan *Whatsapp Gateway* menggunakan bahasa pemrograman PHP. Hasil penelitian menunjukkan bahwa sistem monitoring *server* berbasis terminal Linux dan *Whatsapp Gateway* dapat memberikan notifikasi kepada admin secara *real-time* jika terjadi masalah pada *server*, sehingga dapat membantu dalam mengidentifikasi dan mengatasi masalah dengan cepat. Dengan demikian, sistem ini dapat memperbaiki efisiensi dalam pemantauan *server* dan mengurangi *downtime*.

Kata Kunci: Monitoring, *Server*, Linux, *Whatsapp Gateway*

ABSTRACT

This research aims to design a server monitoring system based on Linux terminal and Whatsapp Gateway for PT. XYZ. The purpose of this system is to monitor server condition in real-time and provide notification to the admin if there is a problem with the server. The method used in this research is by integrating the Linux terminal application with the Whatsapp Gateway using PHP programming language. The research results show that the server monitoring system based on Linux terminal and Whatsapp Gateway can provide real-time notification to the admin if there is a problem with the server, which can help in identifying and solving problems quickly. Thus, this system can improve efficiency in server monitoring and reduce downtime

Keywords: Monitoring, server, Linux, *Whatsapp Gateway*.

PENDAHULUAN

Dalam suatu organisasi perusahaan yang menggunakan dukungan teknologi informasi di dalam melakukan pekerjaan sehari-hari (Hidayat & Muttaqin, 2018; Pahlevi et al., 2018), maka operasional di bidang sistem dan teknologi informasi menjadi suatu masalah yang kritis. Secara umum, lembaga yang menangani sistem dan teknologi informasi banyak menerima keluhan dalam pelaksanaan operasional dan teknologi informasi setiap harinya. Penanganan permasalahan tersebut terkait erat dalam hal pelayanan sehingga memerlukan pengelolaan yang baik (Hidayat & Muttaqin, 2018; Tolle et al., 2017).

Salah satu teknologi informasi yang banyak digunakan aplikasi *Whatsapp* yang merupakan salah satu sosial media layanan pesan singkat pengganti *Short Message Service (SMS)* yang paling banyak digunakan oleh masyarakat Indonesia. Menurut hasil



penelitian *We Are Social* sebanyak 88,7% masyarakat Indonesia menggunakan Sosial Media *Whatsapp*. Selain sebagai sarana berkirim pesan secara elektronik. *Whatsapp* juga memiliki peran penting sebagai identitas seseorang di internet (Rahardja et al., 2018).

Banyak manfaat yang membantu admin *server* dengan memberikan notifikasi *server* penggunaan *harddisk* (NLeng & Li, 2018). Saat ini admin *server* masih menggunakan metode manual dengan melakukan pengecekan langsung melalui *server* tersebut. Metode tersebut sangat tidak efektif karena terkadang admin *server* lupa untuk mengecek penggunaan *harddisk* sehingga ada beberapa *server down* (Pranowo & Mohammad, 2021) dikarenakan *harddisk* yang digunakan sudah penuh. Untuk meningkatkan kemudahan admin *server* dibutuhkan sebuah notifikasi otomatis dengan mengirimkan pesan ke nomor *Whatsapp* admin *server* yang dapat memberikan informasi jika ada *harddisk server* yang sudah hampir penuh. Hal tersebut juga untuk mencegah hal-hal yang tidak diinginkan terhadap *server* yang dikelola. Metode yang digunakan dalam perancangan aplikasi ini yaitu menggunakan metode *Sequential Linear*. Penggunaan metode *Sequential Linear* karena dalam metode ini terdapat tahapan Uji Coba dan Verifikasi sistem sehingga dapat meminimalisir terjadinya kesalahan.

Adapun masalah yang mendasari pembuatan monitoring *server system* berbasis terminal linux dan *whatsapp gateway* ini adalah: (1) proses pengecekan *server system* berbasis terminal Linux masih menggunakan metode manual; (2) admin *Server* sering lupa untuk melakukan pengecekan *harddisk* secara rutin; (3) mencegah terjadinya *down* pada *server* yang dikelola. Batasan masalah dalam pembuatan monitoring *server system* berbasis terminal linux dan *whatsapp gateway* ini adalah: (1) *server* yang akan menjadi objek penerapan monitoring *server system* berbasis terminal Linux dan *Whatsapp Gateway* hanya *server* dengan sistem operasi Linux; (2) pengiriman notifikasi secara realtime menggunakan *Whatsapp Gateway*; (3) sistem yang dicek hanya penggunaan *harddisk* saja. Tujuan dari penelitian ini yaitu: (1) memberikan kemudahan bagi administrator dalam melakukan pengecekan *harddisk* pada *server* yang dikelola; (2) memberikan informasi penggunaan *harddisk* pada *server* yang dikelola.

TINJAUAN PUSTAKA

Sistem adalah sekelompok orang yang bekerja sama menurut aturan yang sistematis untuk membentuk kesatuan tindakan untuk mencapai tujuan. Pada saat yang sama, data yang telah diolah sedemikian rupa menjadi lebih berguna dan bermakna (Tumini & Fitria, 2021) bagi penerimanya sebagai informasi dalam pengambilan keputusan. Sistem informasi merupakan kombinasi teratur dari orang, *hardware*, *software*, jaringan komunikasi, dan sumber daya data, yang mengumpulkan, mengubah, dan menyebarkan informasi dalam sebuah organisasi (Anggraeni, 2017).

Sistem operasi perangkat lunak yang menghubungkan antara pengguna komputer (*brainware*) dengan perangkat keras komputer (*hardware*). Tanpa sistem operasi maka perangkat keras komputer tidak bisa digunakan. Saat ini sudah banyak program sistem operasi yang dibuat oleh korporasi untuk digunakan oleh pengguna komputer, antara lain sistem operasi Windows, Linux, dan Macintosh (Wahid, 2019). Notifikasi adalah sebuah pesan atau informasi dari suatu program aplikasi Android yang ditampilkan diluar antar muka pengguna yang secara normal dikendalikan oleh aplikasi

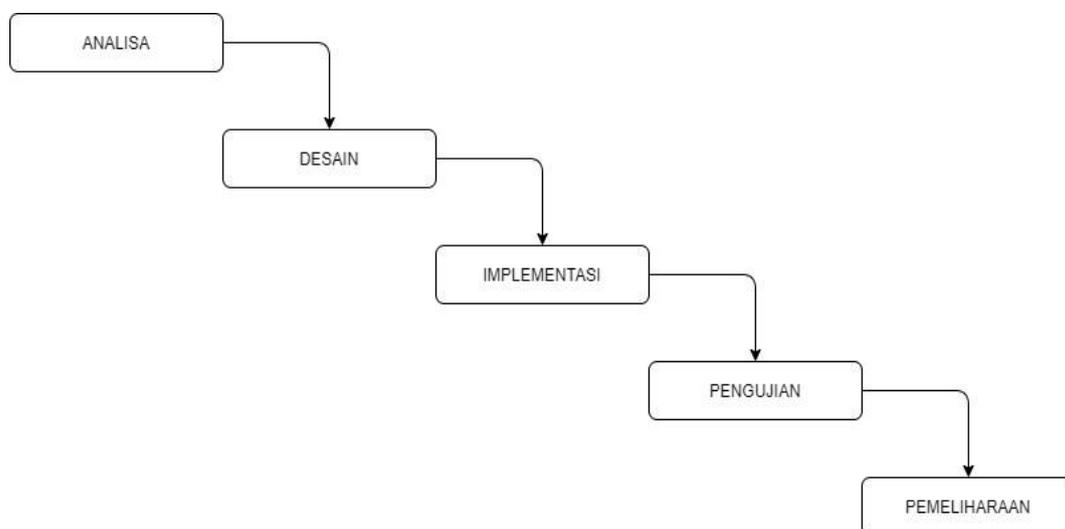
(Tolle et al., 2017). *Whatsapp* telah dimanfaatkan oleh tokoh masyarakat untuk berkomunikasi, pemberitahuan informasi yang disampaikan lebih efektif, dan lain- lain. (Debby Nisa, Inne Husein, 2020). *Server* adalah pusat penyedia layanan dan pengolahan data dalam suatu jaringan, permintaan yang dikirim oleh client akan diolah *server*. Kinerja *server* bergantung terhadap paket yang dikirim oleh client pada jaringan (Suwanto et al., 2019).



Gambar 1. Ilustrasi *Server*

METODE PENELITIAN

Model *Linear Sequential* atau model air terjun, model sekuensial linier menunjukkan sistematis, pendekatan sekuensial untuk pengembangan perangkat (Bungin, 2019). Gambar 2 merupakan tahapan *Sequential Linear* dari analisa, desain, implementasi, pengujian dan terakhir pemeliharaan.



Gambar 2. Model *Sequential Linear*

System Information Engineering and Modeling

Karena perangkat lunak selalu merupakan bagian dari sistem yang besar (atau bisnis), pekerjaan berawal dengan menetapkan persyaratan untuk semua elemen sistem (Voutama, 2022) dan kemudian mengalokasikan beberapa subset dari kebutuhan tersebut untuk perangkat lunak (Slameto & Lukman, 2017), melihat sistem ini terpenting ketika perangkat lunak harus berinteraksi dengan elemen lain seperti hardware, orang, dan *database*. Rekayasa dan analisis sistem meliputi pengumpulan persyaratan/kebutuhan pada tingkat sistem dengan sejumlah kecil desain dan analisis tingkat atas. Rekayasa Informasi yang mencakup pengumpulan persyaratan di tingkat bisnis strategis dan tingkat area bisnis.

Analisa

Proses pengumpulan persyaratan ditingkatkan dan secara khusus difokuskan pada perangkat lunak. Untuk memahami jenis program yang akan dibuat, perekayasa perangkat lunak harus memahami domain data perangkat lunak dan kinerja yang diperlukan serta fungsi antarmuka. Persyaratan sistem dan perangkat lunak didokumentasikan dan didiskusikan dengan pelanggan.

Desain

Desain perangkat lunak sebenarnya adalah proses dengan banyak langkah yang berfokus pada empat atribut yang berbeda dari program: struktur data, arsitektur perangkat lunak, representasi *interface*, dan prosedural (algoritmik) secara rinci. Proses desain menerjemahkan kebutuhan ke dalam sebuah representasi perangkat lunak yang dapat dinilai kualitasnya sebelum coding dimulai.

Implementasi

Langkah-langkah implementasi kedalam pembuatan kode program, desain diterjemahkan ke dalam bentuk yang dapat dibaca mesin, ketika desain dilakukan secara detail pembuatan kode dapat dilakukan secara mekanis.

Pengujian

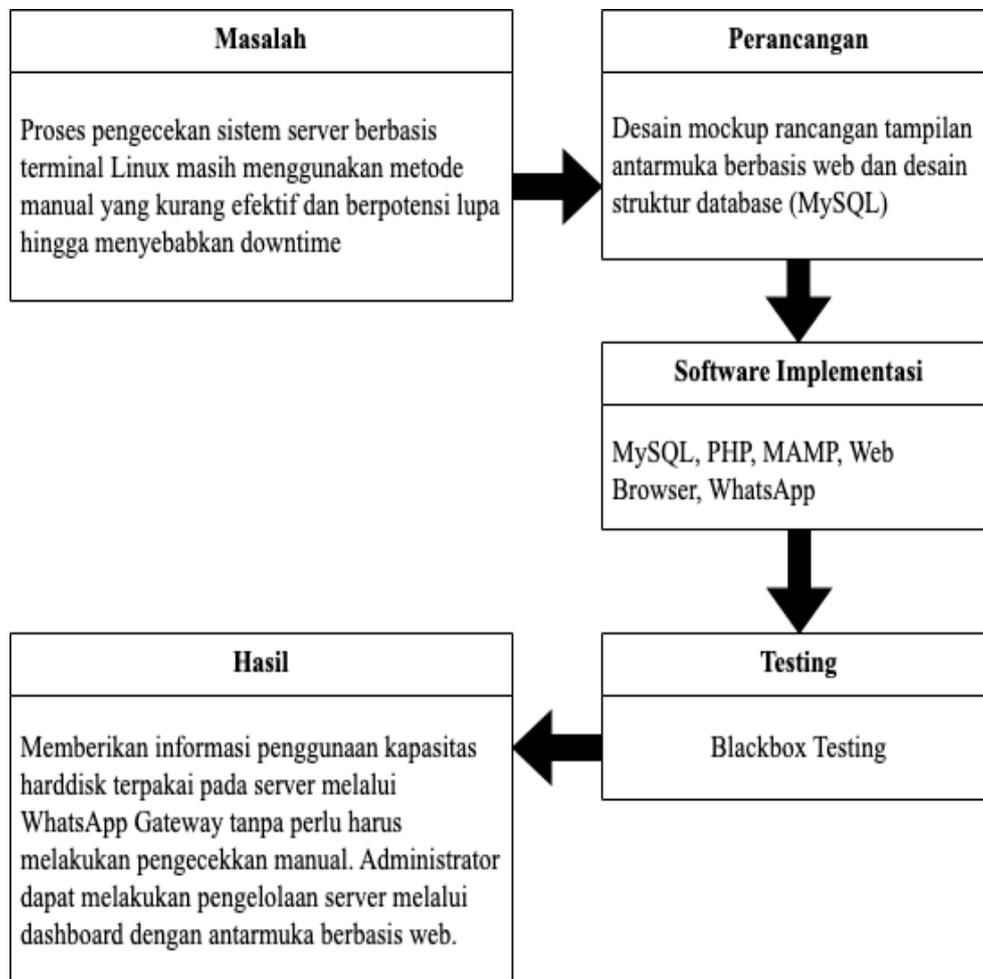
Setelah kode telah dihasilkan, pengujian program dimulai. Proses pengujian berfokus pada bagian internal logis dari perangkat lunak dan memastikan bahwa semua instruksi diuji serta bagian eksternal fungsional; yaitu, melakukan pengujian untuk mengungkap kesalahan dan memastikan bahwa pendefinisian masukan akan menghasilkan hasil aktual yang sesuai dengan hasil yang dibutuhkan.

Support

Perangkat lunak pasti akan mengalami perubahan setelah disampaikan kepada user. Perubahan akan terjadi karena memiliki kesalahan dan 2 harus menyesuaikan mengakomodasi perubahan dalam lingkungan eksternal (misalnya, perubahan yang diperlukan karena sistem operasi baru atau perangkat perifer lainnya), atau karena pelanggan memerlukan fungsional atau peningkatan kinerja lainnya. Perangkat lunak pendukung/pemeliharaan menerapkan kembali setiap tahap sebelumnya untuk program yang ada saat ini daripada instalasi yang baru.

Kerangka Pemikiran

Kerangka Pemikiran dapat menjelaskan secara garis besar logika terhadap alur berjalannya penelitian yang yang dilakukan disajikan pada Gambar 3.



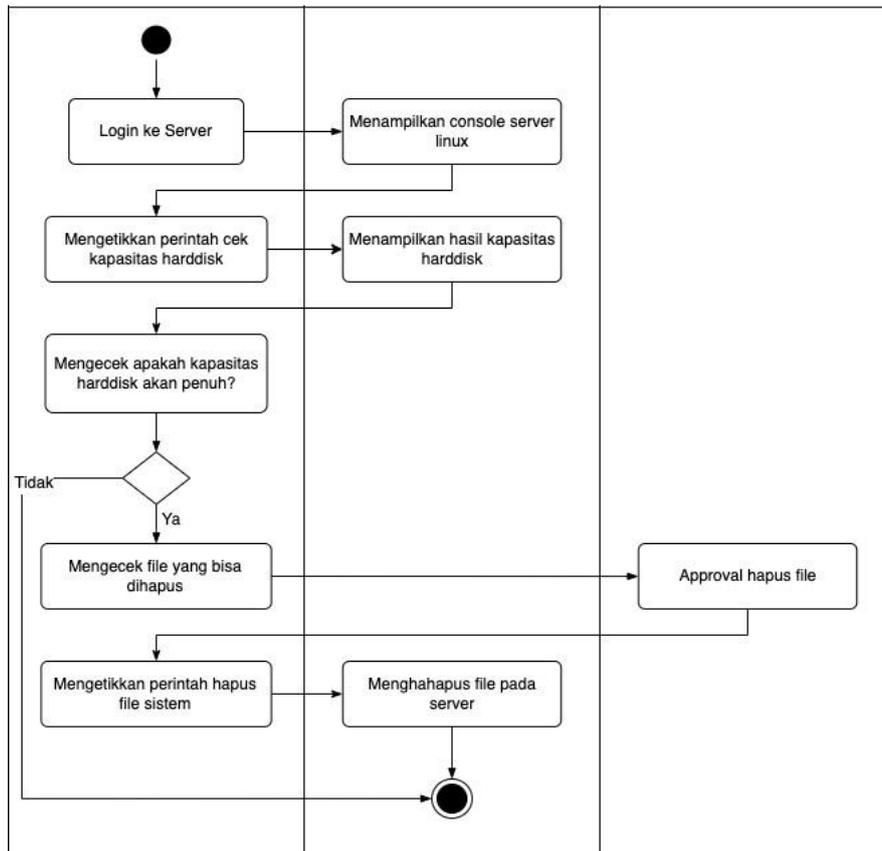
Gambar 3. Kerangka Pemikiran Penelitian

HASIL DAN PEMBAHASAN

PT. XYZ adalah sebuah perusahaan yang fokus pada jasa layanan service komputerisasi: Implementasi sistem, konsultasi dan maintenance sistem terutama dalam migrasi sistem windows ke sistem Linux.

Analisa Sistem yang Berjalan

Berdasarkan hasil penelitian yang dilakukan pada PT.XYZ proses pengecekan *harddisk* masih menggunakan metode manual sehingga sering kali lupa untuk melakukan pengecekan *harddisk*.



Gambar 4. Analisa Sistem Berjalan

Pada Gambar 4 menjelaskan analisa sistem berjalan pada PT. XYZ yaitu: admin login ke *server*, kemudian Sistem akan menampilkan console, lalu admin menyetikkan perintah cek kapasitas *harddisk*, kemudian sistem akan menampilkan kapasitas *harddisk*, selanjutnya admin mengecek kapasitas *harddisk*, jika *harddisk* akan penuh maka admin menyetikkan perintah hapus *file* dan system akan menghapus *file*, kemudian admin mengecek kapasitas *harddisk*, jika *harddisk* tidak penuh proses selesai.

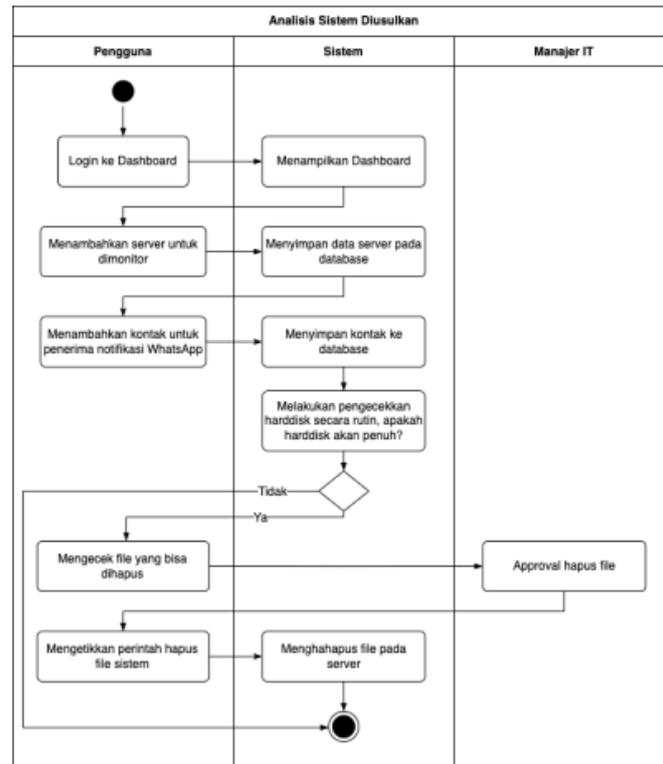
Analisa Masalah

Berdasarkan alur sistem berjalan, proses pengecekan *harddisk* pada *server* PT. XYZ, peneliti menemukan beberapa permasalahan yang dihadapi. Adapun permasalahannya adalah sebagai berikut: (1) proses pengecekan *harddisk* masih menggunakan cara manual; (2) informasi pemakaian *harddisk* tidak dapat langsung di dapatkan secara langsung, karena harus melakukan login dan menyetikkan perintah pada *server* untuk melakukan pengecekan pada *harddisk*. (3) terdapat potensi kapasitas *harddisk* penuh karena kelalaian admin yang tidak melakukan pengecekan dan dapat mengakibatkan *downtime* pada *server*.

Analisa Sistem Usulan

Adapun sistem usulan yang peneliti usulkan untuk memberikan alternatif melakukan pemecahan masalah dengan membuat *Dashboard Monitoring Server* System berbasis Terminal Linux dan *Whatsapp Gateway*, dengan harapan adanya

sistem ini akan memudahkan dan mempercepat melakukan pengecekan *harddisk* dan mendapatkan notifikasi realtime terhadap admin dan manajer jika *harddisk server* akan penuh.



Gambar 5. Analisa Sistem Usulan

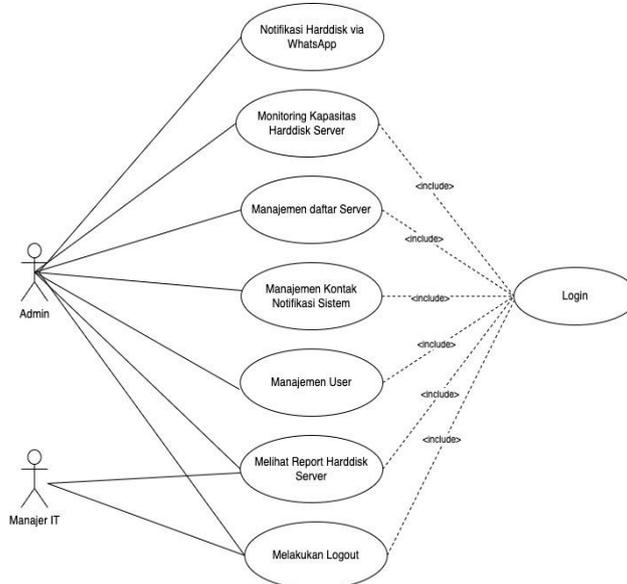
Gambar 5 merupakan analisa sistem usulan PT. XYZ yaitu: admin Login ke *Dashboard*, kemudian admin mendaftarkan *server* yang ingin dimonitor kapasitas *harddisk*nya, lalu admin mendaftarkan nomor *Whatsapp* yang akan mendapat notifikasi ketika kapasitas *harddisk* akan penuh, selanjutnya Sistem akan otomatis menjalankan script Bash setiap 15 menit, kemudian sistem insert *database* otomatis dari menjalankan Script Bash, lalu sistem akan menampilkan pemakaian *dashboard*. Jika *harddisk* tidak penuh proses selesai, jika *harddisk* akan penuh, Admin akan mendapatkan notifikasi melalui *Whatsapp*, lalu admin akan mengecek *file* yang bisa dihapus, lalu admin akan meminta approval penghapusan *file* kepada Manajer IT, kemudian Manajer IT akan mengecek *file* nya, jika aman boleh dihapus, lalu admin akan menghapus *file* nya atas persetujuan Manajer IT.

Perancangan Sistem

Pada sistem ini meliputi analisa dan perancangan terhadap aplikasi yang dibuat. Agar sistem informasi dapat diimplementasikan, maka diperlukan komponen pendukung baik berupa perangkat keras maupun perangkat lunak. Pemodelan sistem informasi yang dibuat menggunakan use case diagram, *Activity* diagram dan *sequence* diagram.

Use Case Diagram

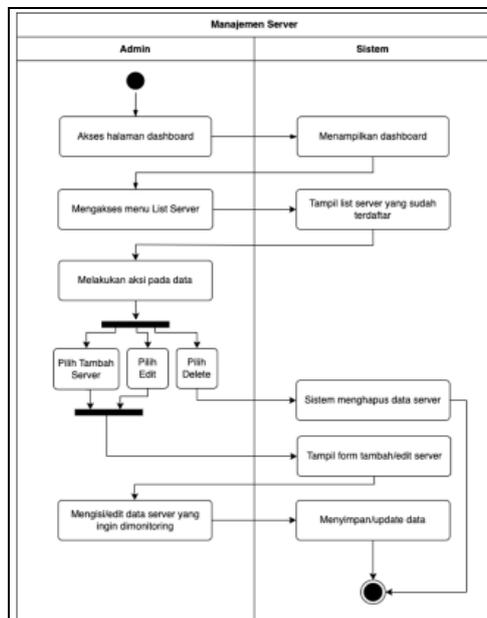
Pada Gambar 6 *use case* dalam implementasi *dashboard monitoring server system* berbasis terminal Linux dan *Whatsapp Gateway* menjelaskan peran Admin dan Manajer IT terhadap sistem yang dibangun.



Gambar 6. Use Case Diagram Sistem Usulan

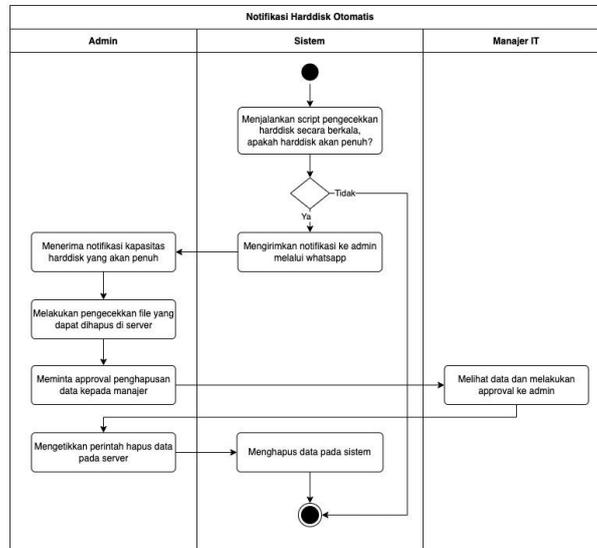
Activity Diagram

Kapasitas *harddisk* setiap *server* dapat dicek langsung pada halaman *dashboard home* halaman *website*. Kapasitas *harddisk* setiap *server* dapat dicek langsung pada halaman *dashboard* halaman *home website*.



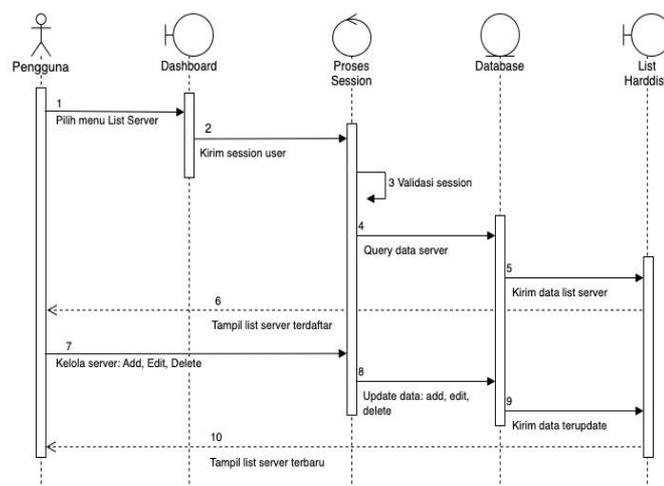
Gambar 7. Activity Diagram Kapasitas Harddisk

Activity diagram pada Gambar 7 memodelkan alur proses penggunaan sistem ketika mengelola *server*, baik itu untuk menambahkan, menyunting ataupun menghapus *server*. *Server* yang sudah terdaftar akan dimonitoring oleh sistem untuk kapasitas *harddisk*-nya.



Gambar 8. *Activity Diagram* Notifikasi Otomatis

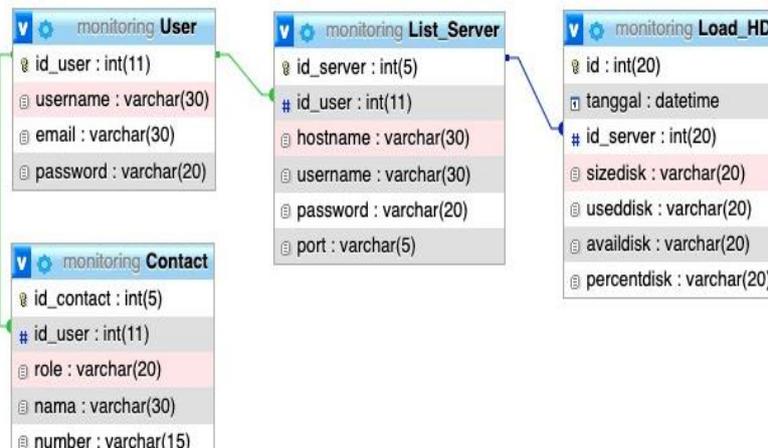
Pada Gambar 8 merupakan alur proses ketika sistem mengirimkan notifikasi otomatis ke kontak yang sudah terdaftar. Sistem akan melakukan pengecekan secara berkala pada latar belakang, jika hasil pengecekan yang didapat *harddisk server* akan penuh maka sistem akan mengirimkan notifikasi. Ketika user mengakses dan melihat laporan dari kapasitas *harddisk* pada *server* yang terdaftar. *User* dapat melihat laporan berdasarkan data harian maupun bulanan. Selain itu untuk pelaporan, admin juga dapat melakukan *export* data ke berbagai pilihan format yang tersedia.



Gambar 9. *Sequence Diagram* Manajemen Server

Elemen penyusun dari diagram *sequence* pada gambar 9 yakni sebagai berikut:
 (1) empat *lifeline*, yakni halaman *dashboard*, sistem sebagai koneksi basis data,

database dan halaman list *harddisk*; (2) satu aktor, yakni pengguna sistem yang merupakan admin; (3) sepuluh *message*, yang berisi informasi aktivitas yang terjadi pada setiap *sequence* dari diagram manajemen *server*.

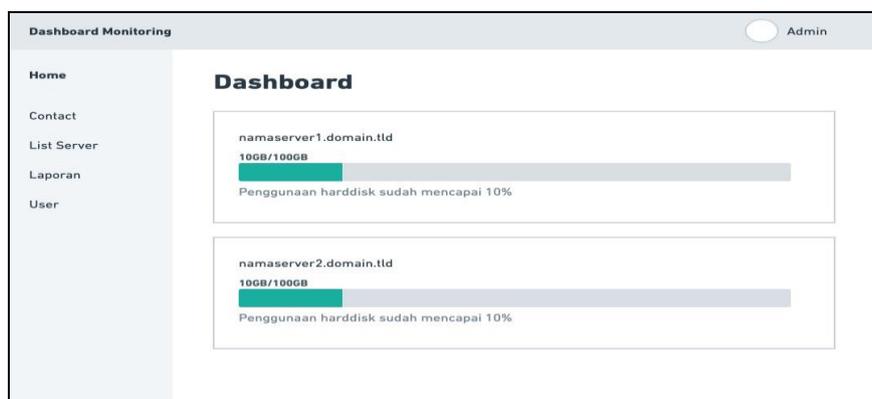


Gambar 10. Struktur *Database* Sistem

Untuk menggambarkan hubungan dan relasi antar beberapa tabel data yang ada pada basis data, peneliti membuat data model seperti pada Gambar 10.

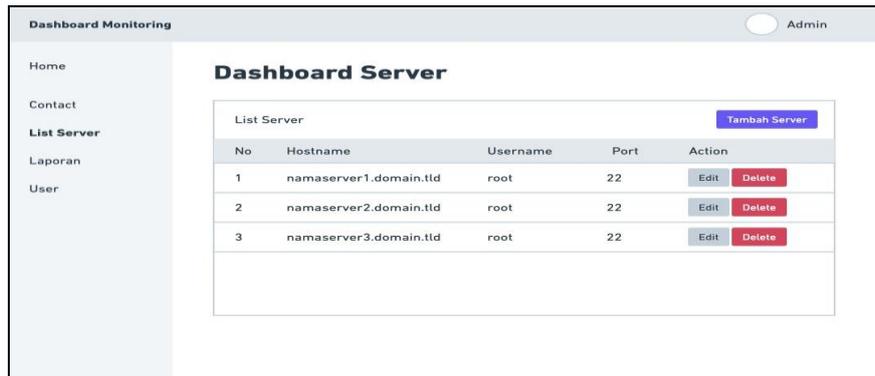
Rancangan Tampilan

Halaman *dashboard* dirancang untuk menampilkan informasi ringkasan dari penggunaan *harddisk* pada *server* yang sudah didaftarkan dan dimonitoring oleh sistem. Disajikan pada Gambar 11.



Gambar 11. Rancangan Antarmuka Halaman *Dashboard*

Pada Gambar 12, *server* yang terdaftar pada sistem untuk dimonitoring dapat dilihat pada halaman list *server*, admin juga dapat menambahkan, mengedit ataupun menghapus *server* yang ada.



Gambar 12. Rancangan Antarmuka Halaman List Server

Implementasi *Whatsapp Gateway*

Untuk melakukan pengiriman notifikasi ketika terdapat kapasitas *harddisk* yang akan penuh, peneliti menggunakan *Whatsapp gateway* sebagai pengirim pesan ke *Whatsapp* kontak terdaftar. Berikut alur pengiriman notifikasi yang dilakukan sistem melalui *Whatsapp gateway* tersebut.

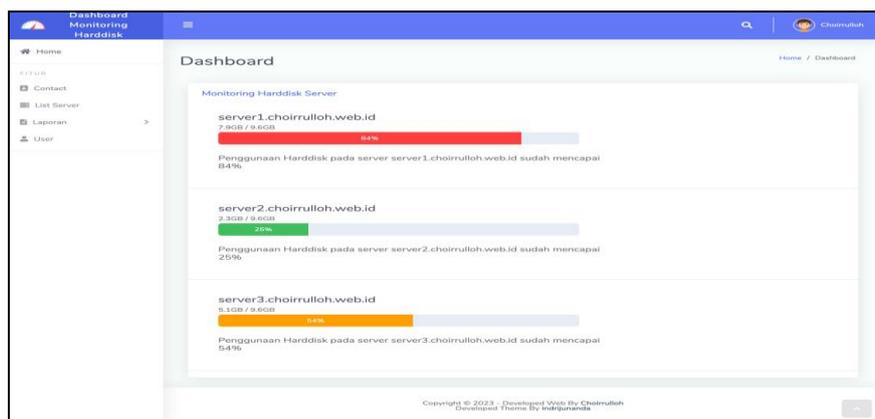
Pengecekan Kapasitas *Harddisk*

Sistem akan melakukan pengecekan kapasitas *harddisk* pada setiap *server* yang terdaftar, kemudian sistem akan mengambil persentase dari penggunaan *harddisk* tersebut.

```
$output = $ssh->exec('df -h /');
// Cari nilai persentase penggunaan harddisk
preg_match('\d+%', $output, $matches);
$used = $matches[0];
```

Implementasi *Web Aplikasi*

Setelah perancangan selesai dilakukan, setelah itu melakukan coding untuk membangun sistem tersebut dengan menggunakan bahasa pemrograman *PHP* dan *CSS Bootstrap*. Berikut hasil dari implementasi sistem yang dilakukan untuk *website* halaman *Dashboard*.



Gambar 13. Implementasi Halaman *Dashboard*

Halaman *dashboard* diimplementasikan dengan menampilkan data *server* beserta grafik bar penggunaan *harddisk* dengan warna yang berbeda untuk tergantung dari persentasi penggunaan *harddisk*.

Pengujian Sistem

Pengujian sistem web aplikasi dilakukan untuk menguji kesesuaian sistem yang dibuat. Penulis menggunakan metode blackbox testing untuk pengujian tersebut. Tujuan utama dilakukan pengujian ini yakni untuk menguji fungsi dari sistem dan memastikan sudah dapat berjalan sesuai dengan output atau hasil yang diharapkan.

Pengujian sistem dilakukan terhadap beberapa halaman web aplikasi yang sudah dibuat dan melakukan pengujian pada masing- masing fungsi. Adapun rencana pengujian sistem yang dilakukan peneliti pada sistem tersebut yakni sebagai berikut pada Tabel 1.

Tabel 1. Rencana Pengujian Blackbox

Kelas Uji	Test Uji	Jenis Uji
Login	Isi data pada form login	Blackbox
	Login menggunakan akun terdaftar	Blackbox
	Login menggunakan akun acak	Blackbox
Dashboard	Lihat list <i>server</i> terdaftar	Blackbox
	Lihat kapasitas <i>harddisk server</i>	Blackbox
Contact	Lihat daftar kontak terdaftar	Blackbox
	Tambah kontak baru	Blackbox
	Edit kontak yang terdaftar	Blackbox
	Hapus kontak yang terdaftar	Blackbox
List Server	Lihat daftar <i>server</i> terdaftar	Blackbox
	Tambah data <i>server</i> baru	Blackbox
	Edit data <i>server</i> yang terdaftar	Blackbox
	Hapus <i>server</i> yang terdaftar	Blackbox
Laporan	Lihat daftar laporan tersedia	Blackbox
	Filter tanggal data laporan	Blackbox
	Cari laporan dengan fitur search	Blackbox
	Copy data laporan	Blackbox
	Export data laporan ke CSV	Blackbox
	Export data laporan ke Excel	Blackbox
	Export data laporan ke PDF	Blackbox
	Print data laporan	Blackbox
User	Lihat daftar <i>user</i> terdaftar	Blackbox
	Tambah <i>user</i> baru	Blackbox
	Edit <i>user</i> yang terdaftar	Blackbox
	Hapus <i>user</i> yang terdaftar	Blackbox

Hasil Pengujian Sistem

Pengujian menggunakan blackbox dilakukan oleh beberapa partisipan sistem administrator sebanyak 5 orang. Pengujian tersebut juga sekaligus memastikan sistem dan fitur yang dibuat sesuai dengan harapan dari pengguna. Pelaksanaan pengujian tersebut dilakukan dengan berdasarkan rencana pengujian yang sudah dibuat

sebelumnya. Berikut pada Tabel 2 daftar pengguna yang melakukan proses pengujian *blackbox*.

Tabel 2. Daftar Penguji Sistem pada *User Acceptance Test*

No	Nama	Pekerjaan
1	Muhammad Dhenandi Putra	Sistem administrator
2	Muhammad Nugi Abdiansyah	Sistem administrator
3	Muhammad Raihan Utomo	Sistem administrator
4	Mohammad Ridwan	Sistem administrator
5	Richi Setya Maulana	Sistem administrator

Penguji tersebut melakukan pengujian dalam rangka *user acceptance test*, untuk memastikan bahwa fitur yang sudah dibuat sesuai dengan hasil yang diharapkan. Adapun pengujian yang dilakukan berdasarkan dengan rencana pengujian yang sudah dibuat sebelumnya. Berikut hasil *user acceptance test* yang sudah dilakukan oleh masing-masing penguji tersebut.

Tabel 3. Hasil *User Acceptance Test* Penguji

Kelas Uji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian
<i>Login</i>	Isi data pada form di halaman <i>login</i>	Data dapat terisi pada form	Berhasil
	<i>Login</i> menggunakan akun terdaftar	<i>Login</i> berhasil dan tampil <i>dashboard</i>	Berhasil
	<i>Login</i> menggunakan akun acak	<i>Login</i> gagal dengan pesan <i>error</i>	Berhasil
<i>Dashboard</i>	Lihat list <i>server</i> terdaftar	Tampil daftar list <i>server</i> terdaftar	Berhasil
<i>Contact</i>	Lihat kapasitas <i>harddisk server</i>	Tampil kapasitas <i>harddisk server</i> beserta persentase.	Berhasil
	Lihat daftar kontak terdaftar	Tampil daftar kontak terdaftar	Berhasil
	Tambah kontak baru	Kontak berhasil disimpan dan tampil pada list	Berhasil
<i>Dashboard</i>	Edit kontak yang terdaftar	Data kontak berhasil diperbarui pada <i>database</i>	Berhasil
	Hapus kontak yang terdaftar	Data kontak terhapus dan hilang dari list kontak	Berhasil
<i>List Server</i>	Lihat daftar <i>server</i> terdaftar	Tampil data <i>server</i> yang terdaftar	Berhasil
	Tambah data <i>server</i> baru	<i>Server</i> berhasil disimpan dan tampil pada list	Berhasil
	Edit data <i>server</i> yang terdaftar	Data <i>server</i> berhasil diperbarui pada <i>database</i>	Berhasil

PENUTUP

Simpulan

Dengan adanya aplikasi ini, akan memudahkan monitoring *server* secara tidak langsung dan mendapatkan informasi terkini mengenai kondisi *server*, karena notifikasinya sudah otomatis terkirim ke *Whatsapp*. setelah aplikasi ini diimplementasikan,



admin menjadi tidak khawatir akan kejadian *server* down lagi karena sudah di periksa terlebih dahulu sebelum *server* nya menjadi down karena kepenuhan pada *harddisk*. Hal tersebut dapat mencegah terjadinya kegagalan *server* akibat kelalaian memonitor *server* secara berkala. Admin akan mendapatkan notifikasi ketika penggunaan *harddisk* sudah mencapai 75% yang akan dikirimkan ke *Whatsapp* admin. Hal ini akan membantu admin untuk melakukan tindakan preventif agar *server* tidak mengalami masalah yang lebih serius.

Saran

Untuk kedepannya agar aplikasi ini dapat dikembangkan kembali bisa menambahkan fitur baru yaitu penambahan atau pengeditan isi notifikasi melalui *dashboard*, memperbaiki masalah pada *server* atau melakukan tindakan perbaikan secara otomatis. Agar aplikasi ini tetap relevan bisa dikembangkan dengan menggunakan *framework* seperti *codeigniter* atau *laravel* yang memiliki metode MVC, Model, View dan *Controller*. Menambahkan fitur notifikasi melalui media komunikasi lain, agar tidak ketergantungan pada aplikasi *Whatsapp* untuk mengirim notifikasi. Hal ini dapat menjadi masalah jika terjadi masalah pada aplikasi *Whatsapp* seperti gangguan jaringan atau pembatasan pengiriman pesan. Meningkatkan keamanan atas risiko keamanan data yang mungkin dapat terjadi karena notifikasi dikirimkan melalui aplikasi *Whatsapp* yang dapat diakses oleh orang yang tidak berhak. Hal ini dapat menimbulkan risiko keamanan data yang serius jika tidak dikelola dengan baik.

REFERENSI

- Anggraeni, E. Y. (2017). *Pengantar Sistem Informasi*. CV. Andi Offset.
- Bungin, B. (2019). *Metodologi penelitian kuantitatif: komunikasi, ekonomi, dan kebijakan publik serta ilmu-ilmu sosial lainnya* (2nd ed.). Kencana Prenada Media Group.
- Debby Nisa, Inne Husein, D. P. W. (2020). Aplikasi Pengelolaan Presensi Berbasis Web dan Whatsapp Gateway di SMAN 2 Mojokerto. *E-Proceeding of Applied Science*, 6(2), 1699–1704. <https://openlibrarypublications.telkomuniversity.ac.id/index.php/appliedscience/article/download/12223/12015>
- Hidayat, T., & Muttaqin, M. (2018). Pengujian Sistem Informasi Pendaftaran dan Pembayaran Wisuda Online menggunakan Black Box Testing dengan Metode Equivalence Partitioning dan Boundary Value Analysis. *Jutis (Jurnal Teknik Informatika)*, 6(1), 25–29. <https://doi.org/10.33592/jutis.Vol6.Iss1.38>
- NLeng, J., & Li, T. (2018). Research on Computer System Information Hiding Anti-Forensic Technology. *Proceedings of the 8th International Conference on Social Network, Communication and Education (SNCE 2018)*. <https://doi.org/10.2991/snec-18.2018.12>
- Pahlevi, O., Mulyani, A., & Khoir, M. (2018). Sistem Informasi Inventori Barang dengan Meerode Oriented di PT.LivazaTeknologi Indonesia Jakarta. *PROSISKO: Jurnal Pengembangan Riset Dan Observasi Sistem Komputer*, 5(1), 27–35.



<https://e-jurnal.lppmunsera.org/index.php/PROSISKO/article/view/587/612>

- Pranowo, D., & Mohammad, K. (2021). Pengujian Performance Server Terhadap Realtime Dashboard Monitoring Menggunakan Metode Scripting Automation Process. *Jurnal Gerbang STMIK Bani Saleh*, 11(2), 12–19. <https://jurnal.stmik.banisaleh.ac.id/ojs2/index.php/JIST/article/view/77>
- Rahardja, U., Aini, Q., & Santoso, N. P. L. (2018). Pengintegrasian Yii Framework Berbasis API pada Sistem Penilaian Absensi. *Jurnal Sisfotenika*, 8(2), 140. <https://doi.org/10.30700/jst.v8i2.403>
- Slameto, A. A., & Lukman, L. (2017). Penerapan Openssh dan Bash Script untuk Simultaneous Remote Access Client pada Laboratorium STMIK Amikom Yogyakarta. *Jurnal Respati*, 9(27). <https://doi.org/10.35842/jtir.v9i27.79>
- Suwanto, R., Ruslianto, I., & Diponegoro, M. (2019). Implementasi Intrusion Prevention System (IPS) Menggunakan Snort dan IPTable pada Monitoring Jaringan Lokal Berbasis Website. *Jurnal Komputer Dan Aplikasi*, 07(1), 97–107. <https://doi.org/10.26418/coding.v7i01.32690>
- Tolle, H., Pinandito, A., Kharisma, A. P., & Dewi, R. K. (2017). *Pengembangan Aplikasi Perangkat Bergerak*. Universitas Brawijaya Press.
- Tumini, & Fitria, M. (2021). Penerapan Metode Scrum pada E-Learning STMIK Cikarang Menggunakan Php dan Mysql. *Jurnal Informatika Simantik*, 6(1), 12–16.
- Voutama, A. (2022). Sistem Antrian Cucian Mobil Berbasis Website Menggunakan Konsep CRM dan Penerapan UML. *Komputika : Jurnal Sistem Komputer*, 11(1), 102–111. <https://doi.org/10.34010/komputika.v11i1.4677>
- Wahid, A. A. (2019). Analisis Sistem Keamanan Pada Sistem Operasi Microsoft Windows, Linux dan Macintosh. *Jurnal Teknik Informatika*, 1(3), 17. https://www.researchgate.net/profile/Aceng_Wahid/publication/330277459_analisis_sistem_keamanan_pada_sistem_operasi_microsoft_windows_linux_dan_macintosh/links/5c36c15092851c22a368cc5e/analisis-sistem-keamanan-pada-sistem-operasi-microsoft-windows-linux-d