

Perancangan Sistem Monitoring dan Forecasting Sumber Daya Server Berbasis Grafana di PT. Dirgantara Indonesia

1st Lucky Indriani Shabana

Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

luckyindrianishabana@student.telkomuniversity.ac.id

2st Muhammad Iqbal

Fakultas Ilmu Terapan
Universitas Telkom
Bandung, Indonesia

miqbal.staff.telkomuniversity.ac.id

3st Hendar Rusman

PT. Dirgantara
Indonesia(persero)
Tbk

Bandung, Indonesia
rusman@indonesian-aerospace.com

Abstrak - Pengelolaan sumber daya server secara tepat sangat dibutuhkan dalam mendukung kinerja sistem informasi yang andal dan stabil, terutama di tengah meningkatnya beban kerja yang dapat menyebabkan terjadinya overload. Penelitian ini dilakukan untuk merancang sistem pemantauan dan peramalan (*forecasting*) terhadap penggunaan sumber daya utama server, yaitu CPU, memori, ruang penyimpanan (*disk*), dan *bandwidth* jaringan, pada sistem operasi Linux berbasis virtualisasi. Sistem ini memanfaatkan Prometheus dan Node Exporter untuk mengumpulkan data secara berkala, kemudian menyimpannya ke dalam InfluxDB sebagai basis data *time-series*. Selanjutnya, visualisasi data dilakukan menggunakan Grafana agar hasil pemantauan dapat disajikan secara real-time dan mudah dipahami. Untuk memprediksi tren pertumbuhan sumber daya, digunakan algoritma Prophet dari pustaka Python yang bekerja berdasarkan pola historis selama 20 hari terakhir. Dari hasil pengolahan data, ditemukan bahwa CPU dan memori menunjukkan tren peningkatan yang cukup signifikan, sementara penggunaan disk dan *bandwidth* cenderung stabil. Visualisasi yang dihasilkan memberikan gambaran yang jelas terhadap kondisi sistem dan dapat dimanfaatkan untuk mendeteksi potensi gangguan sejak dini. Dengan demikian, sistem ini dapat mendukung proses perencanaan kapasitas yang lebih matang dan responsif terhadap kebutuhan perusahaan di masa mendatang.

Kata Kunci — *monitoring server, forecasting, perencanaan kapasitas, Prometheus, Grafana*

I. PENDAHULUAN

PT Dirgantara Indonesia merupakan perusahaan manufaktur pesawat terbang yang berperan penting dalam industri kedirgantaraan nasional. Dalam menjalankan aktivitas operasionalnya, perusahaan sangat bergantung pada infrastruktur teknologi informasi (TI), terutama untuk mendukung proses desain pesawat, simulasi teknik, pengelolaan data produksi, serta komunikasi internal maupun eksternal. Seiring dengan meningkatnya kompleksitas pekerjaan dan volume data yang diolah, pengelolaan sumber daya komputasi seperti CPU, memori, aplikasi, dan *bandwidth* menjadi semakin krusial.

Ketiadaan sistem monitoring dan perencanaan kapasitas yang baik dapat menyebabkan berbagai kendala, seperti *overload*, *downtime*, hingga pemborosan sumber daya TI. Saat ini, PT Dirgantara Indonesia menggunakan server berbasis VMware dan Fedora untuk menjalankan berbagai layanan dan aplikasi secara virtual. Namun, belum adanya sistem monitoring yang terintegrasi dengan peramalan (*forecasting*) menyebabkan departemen TI kesulitan dalam memetakan kebutuhan infrastruktur ke depan. Salah satu permasalahan yang kerap terjadi di lingkungan pusat data adalah keberadaan *comatose servers*, yakni server yang tidak digunakan secara optimal namun tetap aktif dan mengonsumsi energi, yang pada

umumnya dapat mencapai 20–30% dari total server yang ada[1].

Untuk menjawab tantangan tersebut, dibutuhkan sistem *monitoring* dan *forecasting* yang mampu memberikan informasi real-time mengenai kinerja server sekaligus memprediksi tren penggunaan sumber daya di masa mendatang. Sistem ini dirancang menggunakan bahasa pemrograman Python dan memanfaatkan *library* Prophet sebagai metode *forecasting*. Data hasil peramalan kemudian disimpan dalam InfluxDB, sebuah basis data *time-series* yang dirancang khusus untuk menangani data berbasis waktu secara efisien.

InfluxDB dipilih dalam penelitian ini karena memiliki sejumlah keunggulan, antara lain optimalisasi penyimpanan data berbasis *timestamp*, kemampuan tinggi dalam *data ingestion* (mampu menangani jutaan data per detik) [4], efisiensi ruang penyimpanan yang lebih baik dibandingkan dengan database lain seperti Cassandra, serta dukungan query yang menyerupai SQL dan kemudahan integrasi dengan Grafana. Melalui integrasi dengan Grafana [5], sistem ini juga menyediakan tampilan visual yang interaktif, sehingga mempermudah tim TI dalam memantau dan menganalisis performa infrastruktur secara komprehensif.

Dengan adanya sistem monitoring dan forecasting ini, diharapkan PT Dirgantara Indonesia dapat mengelola sumber daya TI secara lebih efisien dan tepat sasaran, serta mendukung pengambilan keputusan dalam perencanaan kapasitas infrastruktur secara berkelanjutan.

II. TINJAUAN PUSTAKA

A. Tinjauan Umum/ Deskripsi dan Skenario dan Perancangan Sistem

Dalam penelitian ini, saya mengembangkan sebuah sistem analisis dan *forecasting* yang dirancang khusus untuk kebutuhan bagian IT Infrastructure di PT Dirgantara Indonesia. Sistem ini bertujuan untuk membantu perusahaan dalam memantau dan memprediksi penggunaan sumber daya server, seperti CPU, memori, disk, dan jaringan (*network*), dalam jangka waktu tertentu. *Monitoring* dilakukan untuk melihat pertumbuhan penggunaan dari waktu ke waktu, sedangkan *forecasting* (peramalan) dilakukan agar perusahaan dapat mengambil langkah antisipatif terhadap beban kerja yang akan datang, seperti perencanaan peremajaan server atau penambahan kapasitas sistem.

Secara teknis, sistem ini dibangun pada sistem operasi Fedora dan menggunakan Node Exporter sebagai agen pengumpul data yang dipasang pada masing-masing server. Node Exporter bertugas mengambil data mentah (*raw data*) terkait performa server. Data ini kemudian dikirim dan

disimpan ke dalam Prometheus, yaitu sebuah sistem time-series database yang berfungsi sebagai pusat penyimpanan data *monitoring* dalam interval waktu tertentu.

Agar data yang dikumpulkan dapat diolah untuk peramalan, saya menggunakan script Python untuk mengeksport data dari Prometheus ke dalam format CSV. File CSV ini lalu disimpan ke dalam InfluxDB, sebuah database yang memang dirancang untuk menyimpan dan mengelola data berbasis waktu (*time-series*) secara efisien [4]. Data dalam bentuk csv tersebut kemudian diproses menggunakan metode Prophet, yaitu algoritma *forecasting* yang dikembangkan oleh Facebook dan dikenal efektif untuk memprediksi tren jangka pendek dan menengah [2]. Dari proses ini dihasilkan data prediksi penggunaan sumber daya dalam periode tertentu ke depan, misalnya 7 hari atau 30 hari.

Hasil dari peramalan tersebut juga disimpan kembali dalam bentuk CSV dan dimasukkan lagi ke dalam InfluxDB. Seluruh data baik data asli (*raw*) maupun hasil *forecasting* kemudian ditampilkan secara visual menggunakan Grafana, yaitu alat visualisasi dashboard yang memungkinkan pengguna melihat grafik penggunaan dan tren pertumbuhan secara *real-time*. Dengan visualisasi ini, tim IT dapat dengan mudah menganalisis kondisi server saat ini sekaligus memantau prediksi penggunaan di masa mendatang.

Secara keseluruhan, sistem ini memberikan solusi lengkap dari pengumpulan data, penyimpanan, pengolahan, hingga penyajian visual yang informatif. Dengan adanya sistem ini, PT Dirgantara Indonesia diharapkan dapat melakukan pemantauan infrastruktur TI secara lebih efisien, serta memiliki data yang kuat untuk pengambilan keputusan terkait pengelolaan server dan perencanaan kapasitas di masa depan.

B. Tinjauan Khusus 1 (Konsep Monitoring System)

Monitoring sistem merupakan proses penting dalam pengelolaan infrastruktur TI untuk menjaga kinerja dan mencegah gangguan. Pemantauan dilakukan terhadap CPU, memori, disk, dan jaringan secara berkala guna mendeteksi anomali serta mendukung perencanaan kapasitas.

- CPU: Digunakan untuk memantau beban kerja. Penggunaan tinggi secara terus-menerus dapat menurunkan performa sistem.
- Memory: Pemantauan RAM penting untuk mencegah penggunaan swap yang memperlambat sistem.
- Disk: Digunakan untuk melihat kapasitas, ruang tersisa, dan tingkat pemakaian agar terhindar dari kegagalan akibat disk penuh.
- Network: Metrik lalu lintas dan kesalahan jaringan dipantau untuk menjaga kestabilan koneksi.

Sistem ini menggunakan Prometheus dan Node Exporter untuk mengumpulkan metrik secara real-time, dengan metode *pull-based* dan dianalisis menggunakan bahasa query PromQL. Hasil monitoring serta prediksi dari model Prophet disimpan dalam InfluxDB, lalu divisualisasikan melalui Grafana agar pengguna dapat melihat kondisi aktual dan tren performa sistem secara interaktif.

C. Tinjauan Khusus 2 (Konsep Forecasting)

Peramalan merupakan metode untuk memprediksi nilai di masa depan berdasarkan data historis. Dalam sistem monitoring server, forecasting digunakan untuk memperkirakan pemakaian sumber daya seperti CPU, memori, disk, dan jaringan, agar perencanaan kapasitas bisa dilakukan lebih awal.

Metode yang digunakan dalam penelitian ini adalah Prophet, algoritma time-series forecasting berbasis model aditif yang dikembangkan oleh Facebook [6]. Prophet menggabungkan komponen tren, musiman, dan error untuk membentuk prediksi, serta dirancang agar mudah digunakan, bahkan oleh non-ahli statistik [6].

Beberapa keunggulan Prophet meliputi kemampuannya menangani data hilang, fleksibel terhadap perubahan tren, serta hasil model yang mudah diinterpretasikan [7]. Dalam penelitian ini, Prophet digunakan untuk memprediksi penggunaan harian komponen server berdasarkan data historis yang tersimpan di InfluxDB. Hasilnya kemudian ditulis kembali ke database dan divisualisasikan melalui Grafana, sehingga perbandingan antara data aktual dan prediksi dapat dilihat secara langsung oleh administrator sistem.

D. Tinjauan Khusus 3 (Visualisasi Data)

Visualisasi data berperan penting dalam sistem monitoring karena menyederhanakan informasi kompleks ke dalam grafik yang mudah dipahami. Dalam konteks ini, visualisasi membantu melihat tren penggunaan sumber daya server serta membandingkan data aktual dengan hasil peramalan untuk mendukung pengambilan Keputusan [5].

Grafana digunakan sebagai alat utama visualisasi. Platform ini mendukung banyak sumber data seperti InfluxDB dan Prometheus, serta menyediakan dashboard interaktif yang dapat dikustomisasi dengan berbagai jenis grafik, tabel, dan alert otomatis [3]. Integrasinya dengan InfluxDB memungkinkan pengguna menampilkan data historis maupun prediksi secara real-time dalam satu tampilan dashboard.

Visualisasi peramalan dilakukan dengan mengambil data dari InfluxDB menggunakan bahasa query Flux. Data seperti penggunaan jaringan, CPU, atau memori ditampilkan dalam bentuk *time series chart* untuk memudahkan identifikasi tren, fluktuasi, dan deviasi. Perbandingan visual antara nilai aktual dan forecast memberikan gambaran menyeluruh atas performa sistem dan potensi masalah ke depan.

E. Tinjauan Khusus 4 (Library untuk forecasting)

Floodlight Pengembangan sistem monitoring dan forecasting membutuhkan dukungan berbagai library Python untuk mengolah, menganalisis, dan menyajikan data secara efisien dan akurat.

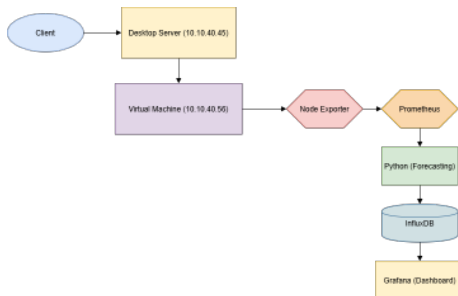
- Pandas digunakan untuk membaca, membersihkan, dan menggabungkan data mentah hasil monitoring. Library ini memudahkan transformasi data sebelum diproses lebih lanjut atau disimpan ke InfluxDB [8]
- Matplotlib dimanfaatkan untuk membuat grafik eksploratif selama proses debugging dan validasi awal hasil peramalan, sebelum divisualisasikan secara interaktif di Grafana.
- Requests berperan dalam mengambil data dari endpoint Prometheus melalui HTTP secara otomatis, sehingga sistem dapat berjalan terjadwal dan dinamis.
- Statsmodels digunakan untuk membandingkan performa model Prophet dengan metode statistik lain seperti ARIMA, guna memastikan keakuratan hasil forecasting [9]
- Datetime membantu dalam pengelolaan format waktu, penyesuaian timestamp, serta konversi waktu antar sistem, yang sangat penting dalam pengolahan data time-series.
- InfluxDB Client merupakan library resmi untuk

mengirim dan membaca data dari InfluxDB. Seluruh hasil monitoring dan forecasting disimpan di sini sebelum divisualisasikan[4]

- Grafana terhubung langsung dengan InfluxDB sebagai data source. Dashboard yang dibuat menampilkan kondisi sistem secara real-time dan memungkinkan perbandingan antara data aktual dan hasil prediksi, sehingga administrator dapat merespons perubahan dengan cepat[5].

III. PEMODELAN DAN PERANCANGAN

A. Arsitektur Sistem



GAMBAR 1
ARSITEKTUR SISTEM

Arsitektur sistem yang dibangun bertujuan untuk memantau dan memprediksi penggunaan sumber daya sistem seperti CPU, memori, disk, dan jaringan pada infrastruktur server perusahaan. Sistem ini merupakan kelanjutan dari permasalahan yang diidentifikasi pada BAB I, yaitu perlunya sistem *monitoring* dan *forecasting* penggunaan sumber daya TI untuk mendukung pengambilan keputusan operasional. Selain itu, landasan teori dari BAB II mengenai monitoring berbasis Prometheus, *forecasting* menggunakan Python, serta visualisasi data dengan Grafana turut menjadi dasar dalam perancangan arsitektur ini.

Dalam arsitektur ini, terdapat dua komponen utama yang terlibat, yaitu Desktop Server (10.10.40.45) dan *Virtual Machine* (10.10.40.56). Di dalam *Virtual Machine*, terdapat beberapa komponen yang saling terintegrasi. Node Exporter berperan dalam mengumpulkan data mentah dari sistem target, seperti CPU *usage*, memori, dan jaringan. Data tersebut kemudian diambil oleh Prometheus, yang bertugas sebagai pengumpul data.

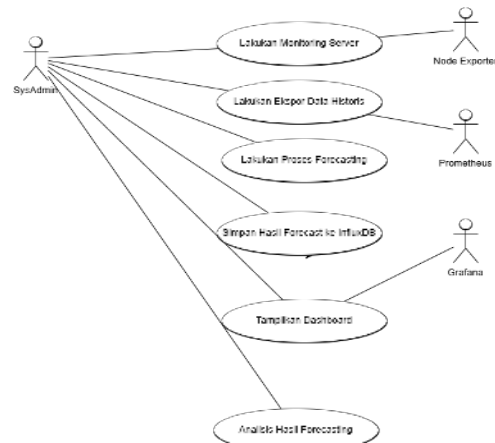
Setelah data dikumpulkan oleh Prometheus, proses dilanjutkan dengan analisis dan peramalan menggunakan script Python (*Forecasting*). Python mengambil data historis dari Prometheus atau hasil ekspor data, kemudian memprosesnya menggunakan model peramalan (Prophet) untuk memprediksi penggunaan sistem dalam periode mendatang. Hasil peramalan ini kemudian disimpan ke dalam InfluxDB, sebuah basis data *time-series* yang mendukung integrasi secara langsung dengan Grafana.

Seluruh informasi, baik data monitoring maupun hasil *forecasting*, divisualisasikan melalui Grafana, yang bertindak sebagai antarmuka dashboard. Pengguna (*Client*) dapat mengakses dashboard ini untuk melihat kondisi sistem secara real-time maupun tren prediksi ke depan, sehingga dapat digunakan sebagai alat bantu dalam pengambilan keputusan dan mitigasi risiko beban sistem.

Dengan arsitektur ini, sistem *monitoring* dan *forecasting* tidak hanya memberikan informasi terkini, tetapi juga insight prediktif terhadap penggunaan sumber daya. Pendekatan ini menjadi solusi efektif terhadap kebutuhan perusahaan dalam memantau performa infrastruktur TI secara lebih proaktif,

sebagaimana dibahas pada latar belakang dan landasan teori sebelumnya..

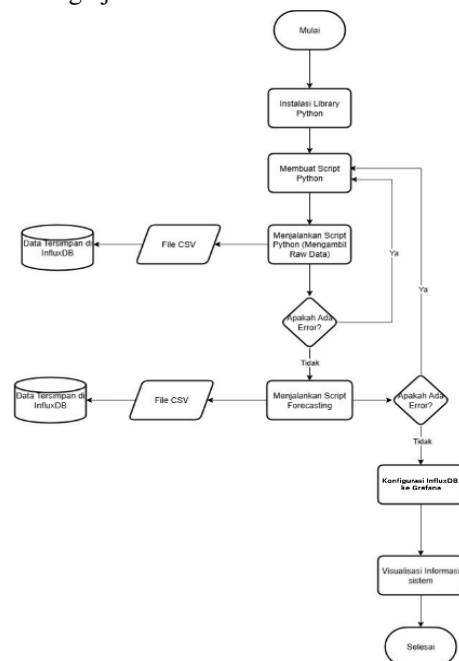
B. Use Case



GAMBAR 2
DIAGRAM USE CASE

Gambar 2 menggambarkan use case sistem monitoring dan forecasting sumber daya server yang melibatkan satu aktor utama, yaitu Administrator atau Teknisi IT. Aktor ini berinteraksi dengan enam fitur utama: memantau performa server secara real-time (melalui Prometheus dan Node Exporter), mengekspor data historis, melakukan forecasting dengan model Prophet, menyimpan hasil prediksi ke InfluxDB, menampilkan data pada dashboard Grafana, serta menganalisis hasil untuk perencanaan kapasitas. Seluruh proses dirancang untuk memberikan pemantauan berkelanjutan dan proyeksi penggunaan CPU, memori, disk, dan jaringan, guna mendukung pengambilan keputusan berbasis data.

C. Proses Pengerjaan



GAMBAR 3
DIAGRAM ALIR

Flowchart ini menggambarkan alur kerja sistem monitoring dan forecasting yang dibangun dengan Python, terintegrasi dengan InfluxDB dan Grafana. Proses dimulai dari instalasi library seperti prophet, pandas, requests, dan influxdb-client, lalu dilanjutkan dengan pengambilan data dari

API (misalnya Prometheus), yang disimpan dalam format CSV dan dikirim ke InfluxDB. Setelah data terkumpul, sistem melakukan pengecekan error. Jika tidak ada masalah, data diproses dengan model forecasting untuk memprediksi penggunaan CPU, memori, disk, dan jaringan berdasarkan tren historis. Hasil prediksi disimpan kembali ke InfluxDB, kemudian divisualisasikan di Grafana. Visualisasi ini menampilkan data aktual dan prediksi dalam satu dashboard, sehingga memudahkan analisis performa sistem dan perencanaan kapasitas. Proses mencakup pengecekan error di setiap tahap, dan akan dihentikan di akhir jika semua berjalan lancar.

IV. HASIL DAN PEMBAHASAN

Sistem yang dibangun bertujuan untuk memantau sekaligus melakukan peramalan penggunaan sumber daya server seperti CPU, memori, disk, dan *bandwidth* secara otomatis, dengan memberikan prediksi jangka pendek terhadap potensi peningkatan atau penurunan beban sistem. Implementasi dilakukan pada dua lingkungan *server*, yaitu *server* lokal (laptop pribadi) sebagai simulasi pengujian awal, dan server produksi (server milik perusahaan) sebagai sumber data aktual yang merepresentasikan kondisi operasional nyata di PT Dirgantara Indonesia.

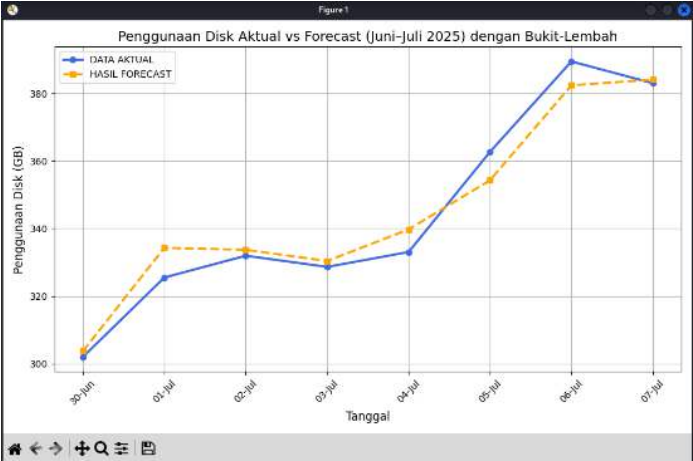
A. Perbandingan sistem dahulu dan sekarang

No	Aspek	Sistem Yang Ada di PT. Dirgantara Indonesia saat ini (Vcenter + Grafana)	Sistem Forecasting (Tugas Akhir)
1.	Fokus Fungsi	Terfokus pada monitoring status saat ini dan historis	Prediksi penggunaan sumber daya untuk waktu tertentu
2.	Sumber data	Agregat dari Vcenter/NetApp	Node exporter per server client (real time + historis)
3.	Jenis Data	CPU, RAM, Disk (tidak mencakup Bandwidth)	CPU< RAM< disk, dan Bandwidth.
4.	Metode Analisis	Visualisasi dan pelaporan	Analisis Tren yang terjadi dan forecasting menggunakan prophet
5.	Penyimpanan data	Disimpan ke dalam Grafana	Disimpan ke dalam InfluxDB dan file csv.
6.	Target Monitoring	VM agregat dan virtualisasi	Server individual, client server Perusahaan (128 VM ware)
7.	Output	Status Operasional saat ini	Estimasi pertumbuhan dan prediksi kapasitas untuk beberapa waktu tertentu
8.	Pengambilan keputusan	Berdasarkan observasi manual	Berdasarkan tren historis dan estimasi pengukuran dari system.

Perbedaan signifikan ini menunjukkan bahwa sistem *forecasting* tidak hanya memperluas fungsi *monitoring*, tetapi juga meningkatkan kemampuan perusahaan dalam menyusun strategi jangka menengah hingga panjang. Informasi yang disediakan bukan hanya bersifat informatif, tetapi juga

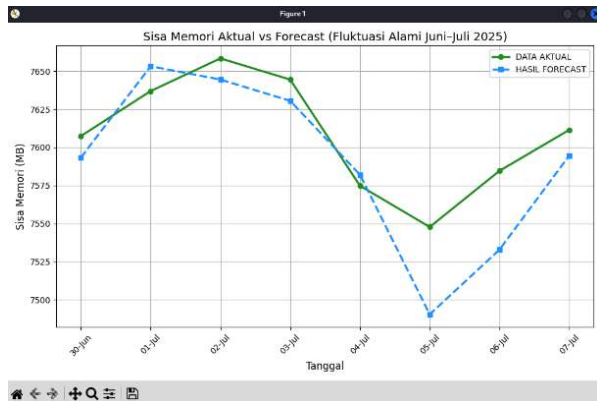
bersifat prediktif dan *strategis*. Dengan sistem baru ini, perusahaan dapat mengantisipasi kebutuhan kapasitas sebelum terjadi *overload*, mengatur jadwal pemeliharaan atau ekspansi sistem secara lebih efisien, serta mendukung pengambilan keputusan berbasis data. Selain itu, sistem ini juga mengisi celah informasi yang belum terpenuhi dalam sistem *eksisting*, seperti pemantauan *bandwidth* dan kapasitas tiap server secara individual.

B. Analisis Hasil Forecasting Pada Dashboard Grafana



GAMBAR 4
PERBANDINGAN DATA HASIL ACTUAL DAN HASIL FORECASTING PENGGUNAAN DISK

Pada awal periode, tepatnya tanggal 30 Juni, kapasitas *disk* yang digunakan berada di angka sekitar 302 GB. Dalam dua hari berikutnya, terjadi kenaikan yang cukup signifikan hingga mencapai kisaran 335 GB. Kenaikan ini kemungkinan disebabkan oleh aktivitas sistem yang cukup padat, seperti logging, penambahan file konfigurasi, atau pemrosesan data. Namun, pada tanggal 3 Juli, terlihat adanya sedikit penurunan penggunaan *disk* meskipun tidak terlalu tajam. Penurunan ini bisa jadi disebabkan oleh proses pembersihan file sementara atau rotasi log yang secara otomatis dilakukan oleh sistem. Setelah itu, kapasitas disk kembali meningkat, bahkan mencapai titik tertinggi pada tanggal 6 Juli dengan nilai sekitar 387 GB, yang kemudian sedikit menurun pada tanggal 7 Juli. Pola seperti ini cukup umum dalam sistem server, di mana aktivitas baca-tulis disk terjadi secara dinamis tergantung beban kerja dan proses yang berjalan. Jika dibandingkan dengan hasil *forecast*, garis prediksi memperlihatkan pola yang cukup mirip dengan data aktual. Meskipun tidak 100% identik, estimasi yang dihasilkan oleh model cukup akurat dalam mengikuti arah perubahan data sebenarnya. Ini menunjukkan bahwa metode *forecasting* yang digunakan mampu menangkap pola *fluktuatif* yang terjadi, meskipun tetap terdapat selisih kecil yang wajar antara prediksi dan data riil.

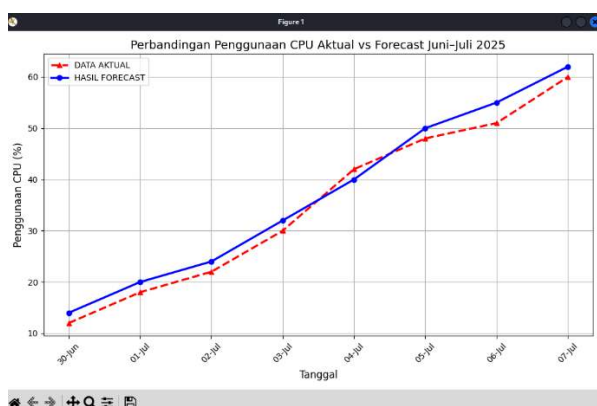


GAMBAR 5
PERBANDINGAN DATA ACTUAL DAN HASIL
FORECASTING PENGGUNAAN MEMORY

Pada tanggal 30 Juni hingga 2 Juli, terlihat bahwa sisa memori mengalami sedikit peningkatan, yang kemungkinan disebabkan oleh berkurangnya beban proses atau pelepasan alokasi memori oleh sistem (misalnya, akibat proses pembersihan cache atau berakhirnya layanan sementara). Kenaikan ini tampak cukup stabil, hingga mencapai titik tertinggi pada 2 Juli.

Setelahnya, mulai 3 Juli hingga 5 Juli, terjadi penurunan sisa memori yang cukup signifikan. Penurunan ini dapat diinterpretasikan sebagai konsekuensi dari meningkatnya aktivitas sistem, seperti penggunaan aplikasi intensif atau proses *background* yang mengonsumsi memori secara terus-menerus. Namun, mulai tanggal 6 Juli hingga 7 Juli, memori mulai kembali stabil dan meningkat secara perlahan, menandakan adanya pergeseran beban kerja atau sistem kembali ke kondisi *idle*.

Hasil *forecasting* yang dihasilkan oleh Prophet secara umum mengikuti arah pola perubahan data aktual. Meskipun terdapat perbedaan nilai di beberapa titik, terutama pada 4–5 Juli, pola naik-turunnya masih berada dalam rentang yang logis dan tidak menyimpang secara drastis dari data aktual. Ini menunjukkan bahwa metode Prophet berhasil memetakan pola musiman dan tren jangka pendek, walaupun ada keterbatasan dalam menangkap anomali sesaat atau fluktuasi yang sangat cepat.



GAMBAR 6
PERBANDINGAN DATA ACTUAL DENGAN HASIL
FORECASTING PENGGUNAAN CPU

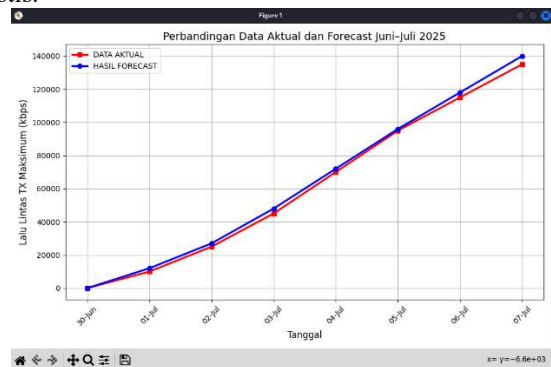
Pengujian Pada tanggal 30 Juni hingga 2 Juli, terlihat bahwa penggunaan CPU meningkat secara bertahap dari nilai terendah menuju kisaran 20–30 persen. Kenaikan ini kemungkinan mencerminkan mulai aktifnya layanan atau aplikasi yang berjalan lebih intens, misalnya karena

permintaan pengguna meningkat pada awal bulan.

Memasuki 3 hingga 5 Juli, laju kenaikan penggunaan CPU menjadi lebih curam. Ini bisa diinterpretasikan sebagai puncak aktivitas sistem, di mana beban kerja bertambah signifikan seperti proses batch, laporan bulanan, atau *traffic* pengguna yang lebih padat. Terlihat bahwa pada 5 Juli, penggunaan CPU sudah mendekati 50 persen, menandakan sistem beroperasi pada kapasitas yang lebih tinggi.

Selanjutnya, pada 6–7 Juli, kenaikan masih berlanjut hingga melewati 60 persen pada 7 Juli. Hal ini menunjukkan tren penggunaan CPU yang konsisten naik selama satu minggu, yang perlu diwaspadai untuk perencanaan kapasitas agar tidak terjadi *overload* atau penurunan performa.

Hasil *forecasting* yang dihasilkan oleh Prophet secara umum mengikuti pola kenaikan yang sama dengan data aktual. Walaupun ada selisih nilai pada beberapa tanggal terutama di sekitar 5–6 Juli arah tren yang dihasilkan tetap serupa dan realistis.



GAMBAR 7
PERBANDINGAN DATA ACTUAL DENGAN HASIL
FORECASTING NETWORK

Pada periode 30 Juni hingga 2 Juli, terlihat bahwa lalu lintas *TX* maksimum meningkat secara bertahap. Kenaikan ini bisa diartikan sebagai bertambahnya aktivitas pengguna atau aplikasi yang memanfaatkan bandwidth lebih besar. Lonjakan permintaan akses jaringan pada awal bulan merupakan hal yang wajar, terutama jika ada proses sinkronisasi data atau aktivitas rutin lainnya.

Memasuki 3 hingga 5 Juli, laju pertumbuhan lalu lintas jaringan semakin curam. Data menunjukkan peningkatan yang cukup signifikan dari sekitar 40.000 kbps hingga hampir 100.000 kbps. Hal ini dapat mencerminkan beban trafik yang lebih padat, misalnya akibat aktivitas layanan daring (*online services*), update sistem massal, atau penggunaan aplikasi yang lebih intens pada jam-jam sibuk.

Selanjutnya, pada 6–7 Juli, lalu lintas *TX* maksimum terus naik mendekati 140.000 kbps. Pola ini memperlihatkan tren kenaikan yang konsisten tanpa penurunan berarti, menandakan bahwa jaringan harus siap menangani permintaan tinggi secara berkelanjutan. Jika tidak diantisipasi, hal ini berpotensi menimbulkan kemacetan (*congestion*) atau penurunan kualitas layanan.

Hasil *forecasting* menggunakan Prophet secara umum mengikuti bentuk kurva aktual dengan sangat baik. Kedua garis hampir saling tumpang tindih, terutama pada 3–7 Juli. Meski ada sedikit perbedaan pada awal periode (30 Juni–1 Juli), selisihnya tidak signifikan. Ini menunjukkan Prophet mampu menangkap tren jangka pendek dan pola musiman dengan cukup akurat dalam kasus ini.

V. KESIMPULAN DAN SARAN

A. Kesimpulan

Penelitian ini menghasilkan sistem monitoring dan

forecasting sumber daya server di PT Dirgantara Indonesia yang mampu memberikan informasi akurat untuk perencanaan kapasitas. Hasil analisis menunjukkan bahwa penggunaan CPU dan memori mengalami tren peningkatan, dengan memori berpotensi melewati ambang batas kritis, sedangkan disk dan bandwidth cenderung stabil. Sistem yang dibangun dengan Node Exporter, Prometheus, Python, dan InfluxDB terbukti efektif dalam mengumpulkan dan menyimpan data secara real-time. Proses peramalan menggunakan Prophet memberikan prediksi yang andal, sementara visualisasi melalui Grafana memudahkan tim TI dalam memantau performa dan tren penggunaan sistem. Dengan demikian, sistem ini mendukung pengambilan keputusan yang lebih tepat dalam pengelolaan kapasitas server.

B. Saran

Untuk meningkatkan efektivitas sistem, disarankan agar ditambahkan fitur notifikasi atau *alert* otomatis yang akan aktif ketika penggunaan CPU atau memori melebihi ambang batas tertentu. Hal ini akan membantu tim TI dalam merespons risiko beban berlebih secara lebih cepat dan proaktif. Selain itu, sistem perlu dikembangkan lebih lanjut untuk mendukung prediksi jangka panjang, karena saat ini forecasting masih terbatas pada jangka pendek. Penggunaan model pembandingan juga penting agar hasil prediksi lebih akurat dan dapat divalidasi. Integrasi sistem secara penuh ke lingkungan produksi juga menjadi langkah penting berikutnya. Proses pengambilan data, peramalan, hingga visualisasi sebaiknya berjalan otomatis tanpa intervensi manual untuk meningkatkan efisiensi operasional. Terakhir, pemanfaatan data historis dalam rentang waktu yang lebih panjang dan bervariasi, seperti data bulanan atau tahunan, perlu dilakukan agar model dapat menangkap pola musiman dan tren jangka panjang dengan lebih baik.

REFERENSI

- [1] Mukta, K., & Harikumar, S. (2024). *Time Series Models for Web Service Activity Prediction*. *International Journal of Computational Science and Engineering*.
- [2] J. Turnbull, *The Prometheus Monitoring System and Time Series Database*. [Online]. Available: <https://prometheus.io/docs/introduction/overview/>. Accessed: Jun. 20, 2025.
- [3] P. B. C. Pragathi, H. Maddirala, and S. M., "Implementing an Effective Infrastructure Monitoring Solution with Prometheus and Grafana," *International Journal of Computer Applications*, vol. 186, no. 38, pp. 7–15, Sep. 2024. doi: 10.5120/ijca2024923873.
- [4] R. Sayfan and D. C. Ongaro, "How InfluxDB handles large-scale time series data ingestion," *InfluxData Tech Paper*, Oct. 2021. [Online]. Available: <https://www.influxdata.com>. Accessed: Jun. 20, 2025.
- [5] A. Sharma and R. Chatterjee, "Efficient Time Series Data Management in Modern Monitoring Systems," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 3, pp. 100–108, Mar. 2021. doi: 10.14569/IJACSA.2021.0120313.
- [6] Satrio, C. B. A., Darmawan, W., Nadia, B. U., & Hanafiah, N. (2021). Time series analysis and forecasting of coronavirus disease in Indonesia using

ARIMA model and PROPHET. *Procedia Computer Science*, 179, 524–532.

- [7] . I. A. Zunic, B. Filipovic, and D. Rancic, "Comparison of Prophet and ARIMA Models for Forecasting of Product Sales," *Facta Universitatis*, vol. 35, no. 2, pp. 159–170, 2020. doi: 10.2298/FUEE2002159Z.
- [8] A. Kumar and P. Dix, "Web Traffic Forecasting Using Facebook Prophet and LSTM," *International Journal of Recent Technology and Engineering (IJRTE)*, vol. 9, no. 5, pp. 101–107, Jan. 2021. doi: 10.35940/ijrte.E6278.019520.
- [9] M. Topping, M. Ashmore, and A. L. Hodgson, "Forecasting Ambient NO₂ Concentrations Using Facebook Prophet," *Environmental Modelling & Software*, vol. 134, 104873, 2020. doi: 10.1016/j.envsoft.2020.104873.
- [10] M. Szafraniec, "Visualizing Influx metrics with Grafana," *j-labs Tech Blog*, 20 Mar. 2021.