BAB II

TINJAUAN PUSTAKA

2.1. Tinjauan Pustaka

2.1.1. Optimisasi Biaya Layanan AWS EC2 dan RDS

Haryanto tahun 2021 membahas optimasi biaya pada layanan basis data di platform AWS, khususnya Amazon Relational Database Service (RDS). Penelitian tersebut bertujuan untuk mengidentifikasi konfigurasi layanan database yang lebih efisien terhadap biaya yang dikeluarkan pengguna. Hasil penelitian menunjukkan bahwa pemilihan spesifikasi dan skema penggunaan layanan yang tidak sesuai dengan kebutuhan aktual dapat menyebabkan pembengkakan biaya operasional, terutama pada layanan database yang berjalan secara kontinu. Penelitian ini menegaskan bahwa optimasi biaya cloud memerlukan analisis terstruktur terhadap pola kebutuhan sumber daya dan karakteristik beban kerja [5].

Kajian serupa juga dibahas oleh Roshandy tahun 2022 yang mengevaluasi performa sistem berbasis cloud menggunakan layanan AWS EC2 dan AWS RDS. Penelitian tersebut tidak hanya menilai kinerja sistem, tetapi juga memperhatikan hubungan antara performa dan biaya operasional yang dikeluarkan. Hasil kajian menunjukkan bahwa biaya cloud tidak dapat dipisahkan dari aspek performa, karena peningkatan spesifikasi layanan dapat berdampak langsung terhadap peningkatan biaya. Dengan demikian, evaluasi performa menjadi bagian penting dalam pengambilan keputusan optimasi sumber daya cloud [12].

Kurniawan tahun 2025 juga membahas implementasi teknologi Amazon Web Services (AWS), khususnya layanan EC2 dan RDS, serta dampaknya terhadap efisiensi operasional dan biaya sistem. Kajian tersebut menunjukkan bahwa keputusan implementasi cloud perlu mempertimbangkan aspek teknis, keamanan, performa, dan biaya secara bersamaan. Implementasi layanan AWS pada lingkungan nyata memperlihatkan bahwa konfigurasi layanan yang digunakan dapat memengaruhi efisiensi biaya secara keseluruhan. Oleh karena itu, pemilihan konfigurasi EC2 dan RDS perlu dilakukan secara hati-hati agar tidak menimbulkan pemborosan sumber daya [11].

Selain itu, Lupis dan Wahyuni tahun 2024 membahas implementasi cloud computing menggunakan Amazon Web Services untuk pembangunan infrastruktur aplikasi. Penelitian tersebut menggunakan beberapa komponen utama seperti VPC, EC2, dan RDS dalam merancang infrastruktur sistem. Kajian ini juga menyinggung pertimbangan biaya, termasuk pemanfaatan skema free tier pada tahap awal implementasi. Hal ini menunjukkan bahwa aspek biaya sudah menjadi pertimbangan sejak proses perancangan infrastruktur aplikasi berbasis cloud [13].

Berdasarkan beberapa penelitian tersebut, dapat disimpulkan bahwa optimasi biaya pada layanan AWS, khususnya EC2 dan RDS, tidak hanya berkaitan dengan perhitungan biaya semata. Optimasi juga perlu mempertimbangkan konfigurasi layanan, karakteristik beban kerja, performa sistem, serta kesesuaian penggunaan sumber daya dengan kebutuhan aktual. Namun, penelitian-penelitian tersebut umumnya masih berfokus pada layanan tertentu dan belum sepenuhnya mengintegrasikan analisis biaya, performa, dan rekomendasi optimasi dalam satu sistem yang utuh. Oleh karena itu, penelitian ini menempatkan EC2 dan RDS sebagai komponen penting dalam pengembangan FinOps Dashboard yang mampu membantu pengguna memahami estimasi biaya dan peluang efisiensi secara lebih praktis.

2.1.2. Estimasi dan Perbandingan Biaya Layanan Cloud

Farhans tahun 2025 melakukan studi literatur mengenai efisiensi biaya dan performa layanan serverless dengan membandingkan AWS Lambda dan Google Cloud Functions. Penelitian ini menunjukkan bahwa evaluasi biaya cloud perlu mempertimbangkan karakteristik layanan, seperti model eksekusi, pola akses, dan beban kerja, karena setiap layanan memiliki model penagihan yang berbeda. Perbedaan model biaya tersebut membuat organisasi perlu memahami pola penggunaan sebelum memilih layanan cloud tertentu. Oleh karena itu, diperlukan pemantauan dan evaluasi biaya secara berkelanjutan agar keputusan teknis tidak menimbulkan pemborosan biaya [6].

Mutiar S. tahun 2024 membahas implementasi estimasi biaya AWS menggunakan AWS Pricing Calculator dalam pengembangan portal berita. Hasil penelitian menunjukkan bahwa estimasi biaya sejak tahap perancangan sistem

penting untuk menjaga kesesuaian antara kebutuhan teknis dan batasan anggaran. Walaupun penelitian tersebut berfokus pada estimasi biaya, temuan yang dihasilkan menunjukkan bahwa aspek biaya cloud dapat diintegrasikan ke dalam proses software development. Dengan demikian, perhitungan biaya tidak hanya dilakukan setelah sistem berjalan, tetapi juga dapat menjadi bagian dari perencanaan awal pengembangan aplikasi [8].

Rahardian tahun 2025 membahas perbandingan manajemen layanan cloud computing antara Amazon Web Services (AWS) dan Google Cloud Platform (GCP). Kajian ini menyoroti perbedaan pendekatan pengelolaan sumber daya, kemudahan administrasi, serta mekanisme estimasi biaya yang disediakan oleh masing-masing penyedia cloud melalui kalkulator layanan. Pembahasan lintas platform tersebut memberikan gambaran posisi AWS dalam konteks kompetisi layanan cloud global, khususnya dari sisi manajemen dan struktur biaya. Dengan adanya perbandingan ini, pengguna dapat memahami bahwa setiap platform cloud memiliki karakteristik pengelolaan biaya yang berbeda [9].

Setiawan tahun 2024 juga membahas perbandingan biaya antara pendekatan komputasi serverless dan model virtual machine (VM). Kajian ini menekankan bahwa pemilihan arsitektur komputasi memiliki pengaruh signifikan terhadap struktur biaya operasional cloud. Faktor-faktor seperti pola beban kerja, frekuensi eksekusi, serta durasi pemakaian menjadi elemen penting dalam menentukan efisiensi biaya pada masing-masing pendekatan. Oleh karena itu, keputusan penggunaan serverless atau VM perlu mempertimbangkan pola penggunaan sistem agar biaya yang dikeluarkan tetap efisien [10].

Dari beberapa kajian tersebut, terlihat bahwa estimasi dan perbandingan biaya cloud merupakan aspek penting dalam pengambilan keputusan teknis. Penelitian-penelitian sebelumnya telah menunjukkan bahwa biaya cloud dipengaruhi oleh jenis layanan, model penagihan, pola eksekusi, durasi penggunaan, serta platform yang digunakan. Namun, sebagian besar kajian masih menempatkan estimasi biaya sebagai alat bantu perhitungan, belum sebagai bagian dari sistem analisis yang mampu menghubungkan data penggunaan resource dengan rekomendasi optimasi. Oleh karena itu, penelitian ini mengembangkan pendekatan yang tidak hanya

menghitung biaya, tetapi juga menyajikan informasi biaya dalam dashboard yang lebih mudah dipahami dan digunakan dalam proses pengambilan keputusan.

2.2. Celah Penelitian

Berdasarkan tinjauan pustaka yang telah dilakukan terhadap penelitian-penelitian terdahulu, dapat diidentifikasi beberapa celah penelitian (*research gap*) yang menjadi dasar dilakukannya penelitian ini :

Pertama, sebagian besar penelitian terdahulu masih berfokus pada satu aspek pengelolaan biaya *cloud* secara terpisah. Penelitian oleh Haryanto [5] dan Roshandy [12] menitikberatkan pada optimisasi biaya dan evaluasi performa layanan tertentu (RDS dan EC2), sedangkan penelitian oleh Setiawan [10] dan Farhans [6] lebih menekankan analisis efisiensi biaya berdasarkan perbedaan arsitektur komputasi (*serverless* dan *virtual machine*). Pendekatan tersebut belum mengintegrasikan berbagai aspek biaya dan penggunaan sumber daya *cloud* ke dalam satu sistem analisis yang menyeluruh.

Kedua, beberapa penelitian telah membahas estimasi biaya dan perencanaan penggunaan *cloud* pada tahap awal pengembangan sistem, seperti yang dilakukan oleh Mutiara Sukma Dewi [8] serta Rahardian [9]. Namun, penelitian-penelitian tersebut masih terbatas pada estimasi atau perbandingan biaya, dan belum mencakup *monitoring* biaya secara berkelanjutan, analisis pemanfaatan sumber daya aktual, serta rekomendasi optimasi berbasis data penggunaan.

Ketiga, penelitian yang mengangkat konsep FinOps dan tata kelola biaya *cloud*, seperti yang dibahas oleh Maulana et al. [7], masih berada pada tataran konseptual dan perancangan arsitektur. Implementasi nyata berupa sistem atau *dashboard* operasional yang mendukung proses FinOps secara langsung—mulai dari pengumpulan data biaya, analisis penggunaan *resource*, hingga penyajian rekomendasi optimasi—belum banyak ditemukan dalam penelitian sebelumnya.

Keempat, beberapa penelitian implementasi AWS [11][13] telah menunjukkan penerapan layanan AWS seperti EC2, RDS, dan VPC dalam membangun infrastruktur aplikasi. Meskipun demikian, penelitian-penelitian tersebut belum secara spesifik mengintegrasikan mekanisme *rightsizing* dan deteksi *unused resource* sebagai bagian dari pengelolaan biaya *cloud* yang terstruktur dan

berkelanjutan.

Berdasarkan celah-celah tersebut, dapat disimpulkan bahwa belum terdapat penelitian yang mengembangkan sebuah *dashboard* berbasis *web* yang secara terintegrasi menggabungkan analisis biaya, *monitoring* penggunaan *resource*, *rightsizing*, dan identifikasi *unused resource* dalam kerangka FinOps pada lingkungan AWS. Oleh karena itu, penelitian ini diarahkan untuk mengisi celah tersebut melalui pengembangan FinOps *Dashboard* berbasis *web* yang mendukung pengelolaan dan optimasi biaya *cloud* AWS secara terstruktur, otomatis, dan berkelanjutan. Perbandingan posisi penelitian ini terhadap penelitian terdahulu dirangkum pada Tabel 2.1.

Tabel 2.1 Perbandingan Penelitian

Penulis	Tahun	Ruang Lingkup	Metode	Kontribusi Utama	Keterbatasan
Haryanto [5]	2025	Optimisasi biaya basis data AWS RDS	Analisis konfigurasi dan pola penggunaan layanan *database cloud*	Mengidentifikasi penyebab pembengkakan biaya RDS dan pentingnya penyesuaian spesifikasi berdasarkan kebutuhan aktual	Fokus hanya pada layanan RDS dan belum terintegrasi dengan sistem *dashboard* pemantauan biaya
Farhans [6]	2025	Efisiensi biaya dan performa layanan *serverless* (AWS Lambda vs Google *Cloud* Functions)	Studi literatur dan analisis komparatif	Menunjukkan perbedaan model biaya dan performa antar layanan *serverless*	Bersifat konseptual dan tidak menghasilkan implementasi sistem operasional

Penulis	Tahun	Ruang Lingkup	Metode	Kontribusi Utama	Keterbatasan
Maulana [7]	2025	Penerapan FinOps dalam arsitektur *hybrid cloud*	Perancangan arsitektur teknologi informasi	Menekankan pentingnya integrasi FinOps untuk transparansi dan akuntabilitas biaya *cloud*	Belum mengimplementasikan sistem pemantauan biaya atau *dashboard* berbasis *web*
Mutiara S. [8]	2024	Estimasi biaya AWS pada proyek pengembangan portal berita	Implementasi AWS *Pricing Calculator* dalam *software development*	Menunjukkan pentingnya estimasi biaya *cloud* sejak tahap perancangan sistem	Terbatas pada estimasi awal biaya dan belum membahas *monitoring* serta optimisasi berkelanjutan
Rahardian [9]	2025	Manajemen *cloud* AWS vs GCP	Analisis perbandingan platform	Perspektif komparatif pengelolaan & estimasi biaya	Tidak membahas FinOps & optimasi otomatis
Setiawan [10]	2024	Biaya *serverless* vs VM	Analisis biaya arsitektur	Menunjukkan pengaruh arsitektur terhadap biaya	Tidak ada *monitoring* berkelanjutan

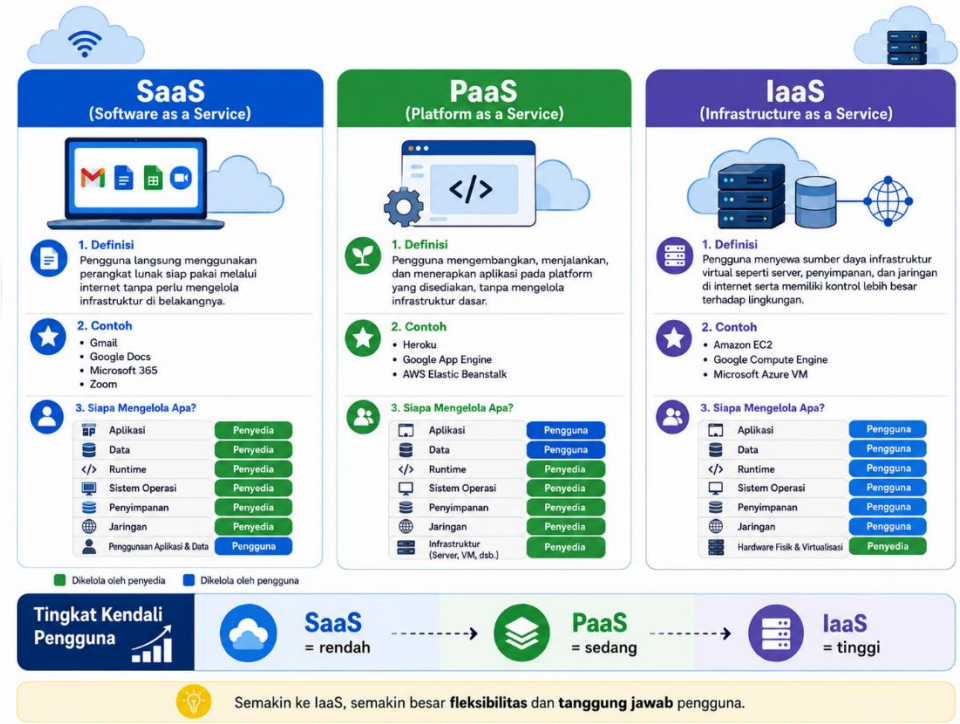
Penulis	Tahun	Ruang Lingkup	Metode	Kontribusi Utama	Keterbatasan
Kurniawan [11]	2025	Implementasi AWS (EC2, RDS)	Implementasi & evaluasi sistem	Gambaran dampak implementasi AWS terhadap biaya	Tidak fokus pada FinOps
Roshandy [12]	2022	Performa vs biaya EC2/RDS	Evaluasi performa & biaya	Mengaitkan performa dengan biaya *cloud*	Tidak menghasilkan rekomendasi optimasi
Lubis & Wahyuni [13]	2024	Infrastruktur aplikasi AWS	Implementasi *cloud computing*	Gambaran teknis *deployment* AWS	Tidak membahas tata kelola biaya
Penelitian ini (Daniel)	**2026**	**Pengelolaan biaya AWS (EC2, RDS, *unused resource*)**	**Pengembangan sistem (*software development*)**	**Mengembangkan FinOps *Dashboard* berbasis *web* untuk analisis biaya, *rightsizing*, dan deteksi *unused resource***	**Tidak melakukan eksekusi otomatis perubahan konfigurasi *resource***

2.3. Landasan Teori

2.3.1. Cloud Computing

Cloud computing merupakan paradigma komputasi yang memungkinkan penyediaan sumber daya teknologi informasi secara *on-demand* melalui internet tanpa organisasi harus memiliki seluruh infrastruktur fisik sendiri. Karakteristik utamanya meliputi kemudahan akses, elastisitas, pemanfaatan sumber daya bersama, dan layanan yang terukur. Secara umum, konsep *cloud computing* dapat dipahami melalui pembagian model layanan yang terdiri dari *Infrastructure as a*

Service (IaaS), Platform as a Service (PaaS), dan Software as a Service (SaaS), sebagaimana diilustrasikan pada Gambar 2.1. Dalam model layanan tersebut, IaaS menyediakan komputasi, penyimpanan, dan jaringan dasar sehingga pengguna masih mengelola sistem operasi serta aplikasinya; PaaS menyediakan platform dan lingkungan pengembangan agar pengguna dapat berfokus pada pembuatan serta deployment aplikasi; sedangkan SaaS menyediakan aplikasi siap pakai yang dapat diakses melalui internet tanpa pengguna perlu mengelola platform maupun infrastrukturnya [24][25].



Gambar 2.1 Gambar Ilustrasi Cloud Computing

2.3.2. Amazon Web Services (AWS)

Amazon *Web* Services (AWS) merupakan salah satu penyedia layanan *cloud computing* yang menawarkan beragam layanan komputasi, penyimpanan, basis data, jaringan, dan pemantauan. AWS memonetisasi layanannya terutama melalui penagihan berbasis konsumsi, sehingga pelanggan membayar sesuai jenis layanan dan lama pemakaian tanpa kontrak jangka panjang yang rumit. Secara umum, pendekatan harga AWS mencakup *pay-as-you-go*, *flat rate*, *save when you commit*, dan *pay less by using more*. Pada layanan komputasi, pendekatan tersebut

juga diterapkan melalui model seperti *On-Demand*, *Savings Plans*, *Spot Instances*, dan reservations, sehingga organisasi dapat memilih struktur biaya yang paling sesuai dengan pola beban kerja dan strategi optimasinya [14][26].

2.3.3. Amazon Elastic Compute Cloud(EC2)

Amazon Elastic Compute *Cloud* (EC2) merupakan layanan komputasi berbasis *virtual machine* pada AWS yang menyediakan variasi *instance* sesuai kebutuhan aplikasi. Pemilihan tipe *instance* yang tidak sesuai dengan karakteristik beban kerja dapat memengaruhi performa sekaligus biaya operasional, sehingga analisis utilisasi *resource* menjadi penting dalam proses optimasi [15]. Oleh karena itu, pemantauan pemanfaatan sumber daya EC2 menjadi dasar pengambilan keputusan dalam analisis biaya *cloud*.

2.3.4. Amazon Relational Database Service(RDS)

Amazon Relational *Database* Service (RDS) merupakan layanan basis data relasional terkelola yang mendukung berbagai *engine database* serta fitur otomasi seperti *backup*, *patching*, dan scaling. Karena biaya RDS dipengaruhi oleh konfigurasi *instance*, penyimpanan, dan pola penggunaan, evaluasi konfigurasi secara berkala diperlukan untuk menjaga efisiensi biaya *cloud* [16]. Hal ini menjadikan RDS salah satu komponen penting dalam analisis biaya *cloud*.

2.3.5. *Rightsizing*

Rightsizing merupakan proses penyesuaian kapasitas sumber daya *cloud* agar selaras dengan kebutuhan aktual aplikasi. Analisis *rightsizing* dilakukan dengan memperhatikan metrik penggunaan seperti CPU, memori, dan aktivitas jaringan, sehingga organisasi dapat menyeimbangkan performa dan biaya secara lebih proporsional [11][12]. Dengan menerapkan *rightsizing* secara rutin, penggunaan *resource cloud* dapat menjadi lebih efisien dan peluang pemborosan biaya dapat dikurangi. Oleh karena itu, *rightsizing* menjadi salah satu praktik utama dalam implementasi FinOps.

2.3.6. *Idle Resource*

Idle resource merupakan sumber daya *cloud* yang masih teralokasi atau aktif tetapi tidak digunakan secara optimal. Contoh *unused resource* meliputi *instance* komputasi yang *idle*, volume penyimpanan yang tidak terpasang, serta

alamat IP yang tidak terasosiasi. Meskipun tidak digunakan, *resource* tersebut tetap dapat menimbulkan biaya operasional. Karena itu, identifikasi *resource* yang tidak termanfaatkan menjadi langkah awal yang penting dalam optimasi biaya *cloud* [2][18]. Dalam konteks FinOps, deteksi *unused resource* membantu meningkatkan efisiensi dan transparansi penggunaan *cloud*.

2.3.7. AWS *Application Programming Interface* (API)

Layanan *cloud* modern umumnya menyediakan antarmuka pemrograman aplikasi (Application Programming Interface/API) agar sistem lain dapat berinteraksi secara terprogram. Pemanfaatan API memungkinkan aplikasi melakukan pengambilan data konfigurasi, status layanan, dan informasi penggunaan secara otomatis dan konsisten, sehingga cocok digunakan untuk membangun sistem *monitoring* dan analisis berbasis *web* [19][20]. Dalam FinOps *Dashboard*, pendekatan ini digunakan untuk mengintegrasikan data teknis dan biaya ke dalam satu sistem analisis yang terpusat.

2.3.8. AWS SDK for Python (Boto3)

AWS SDK for Python (Boto3) merupakan pustaka yang mendukung integrasi layanan AWS melalui bahasa pemrograman Python. Dalam pengembangan aplikasi berbasis *backend* Python, penggunaan SDK seperti boto3 memudahkan proses otomasi pengambilan data layanan *cloud*, pengolahan hasil respons API, serta integrasi ke modul analisis dalam aplikasi [20][21]. Dengan pendekatan ini, proses pengelolaan *cloud* dapat dilakukan secara lebih dinamis dan efisien.

2.3.9. Amazon CloudWatch

Pemantauan metrik penggunaan *resource* seperti CPU utilization, disk I/O, dan network *throughput* merupakan bagian penting dalam pengelolaan lingkungan *cloud*. Data pemantauan tersebut dapat digunakan untuk menilai tingkat pemanfaatan *resource*, mendukung analisis *rightsizing*, serta membantu identifikasi *resource* yang tidak efisien [17][21].

2.3.10. *Technology Acceptance Model* (TAM)

Technology Acceptance Model (TAM) merupakan model yang digunakan untuk memahami penerimaan pengguna terhadap suatu teknologi atau sistem

informasi. Anwar, Bingamawa, dan Ardieansyah tahun (2025) menjelaskan bahwa TAM banyak digunakan untuk menilai penerimaan dan kepuasan pengguna aplikasi berbasis digital karena mampu mengukur aspek kegunaan, kemudahan penggunaan, dan kepuasan pengguna secara komprehensif[27]. TAM menekankan bahwa penerimaan pengguna dipengaruhi oleh persepsi terhadap kegunaan sistem (*perceived usefulness*) dan kemudahan penggunaan sistem *(perceived ease of use*). Kedua aspek tersebut berpengaruh terhadap sikap pengguna dalam menggunakan teknologi baru, yang pada akhirnya dapat memengaruhi minat dan keputusan pengguna untuk menggunakan sistem secara berkelanjutan. Dalam penelitian ini, TAM tidak digunakan sebagai instrumen kuantitatif berbasis skala, melainkan sebagai kerangka konseptual untuk menyusun pertanyaan wawancara, mengelompokkan temuan, dan menafsirkan persepsi pengguna terhadap *FinOps Dashboard* secara kualitatif.