

## BAB II TINJAUAN PUSTAKA

### A. Penelitian Terkait

Sebagai referensi dan juga acuan mengenai berbagai hal dalam penelitian maka digunakan beberapa sumber yang memiliki relevansi dengan penelitian ini, berikut adalah beberapa penelitian yang serupa dengan penelitian ini adalah:

Tabel 2. 1 Penelitian Terkait

| No  | Penulis & Tahun                                      | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|-----|------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                                                  | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 1   | M Sahyudi & Erliyan Redy Susanto (2025)              | Judul Penelitian : Analisis Implementasi Sistem Keamanan Basis Data Berbasis Role-Based Access Control (RBAC) pada Aplikasi Enterprise Resource Planning.<br>Metode : Systematic Literature Review (SLR)<br>Tools/Software : <i>Machine Learning (Decision Tree &amp; Random Forest)</i> NLP, Blockchain.<br>Hasil : Integrasi Machine Learning, NLP, dan Blockchain meningkatkan efektivitas implementasi RBAC, dengan peningkatan keamanan hingga 37%. Otomatisasi RBAC juga menurunkan biaya operasional sebesar 42% dan waktu pengelolaan hingga 65% dibandingkan pendekatan manual.. |
| 2   | Loay Abdelrazek, Leyli Karacay, & Marin Orlic (2025) | Judul Penelitian : <i>Authorization of Knowledge-base Agents in an Intent-based Management Function</i> .<br>Metode : Perancangan Hybrid Authorization Framework<br>Tools/Software : RBAC, ABAC, RelBAC, Knowledge Graph, Zero Trust Architecture.<br>Hasil : Penelitian mengembangkan <i>Hybrid Authorization Framework</i> yang menggabungkan RBAC, ABAC, dan RelBAC untuk menghasilkan otorisasi yang lebih dinamis dan granular. Framework menerapkan <i>least privilege</i> , <i>continuous authorization</i> , dan <i>dynamic policy update</i> pada lingkungan multi-tenant.       |
| 3   | Saurabh Deochake & Vrushali Channpattan (2022)       | Judul Penelitian : <i>Identity and Access Management Framework for Multi-Tenant Resources in Hybrid Cloud Computing</i> .                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

| No  | Penulis & Tahun                                    | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----|----------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                                                | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|     |                                                    | <p>Metode : Perancangan dan implementasi <i>Identity and Access Management (IAM) Framework</i>.</p> <p>Tools/Software : <i>Google Cloud Platform (GCP)</i>, <i>LDAP</i>, <i>Kerberos</i>, <i>OAuth 2.0</i>, <i>Vault</i>.</p> <p>Hasil : Penelitian menghasilkan <i>IAM Framework</i> berbasis <i>mirror identity</i> untuk mengelola akses pada lingkungan <i>hybrid cloud multi-tenant</i>. Framework menerapkan prinsip <i>Authentication</i>, <i>Authorization</i>, <i>Auditing (AAA)</i> dan <i>Least Privilege</i> guna meningkatkan keamanan serta skalabilitas pengelolaan identitas.</p>                                                                                                                                                                                                                                                                                                                                                                                                                        |
| 4   | Aziz Catur Laksono & Bety Wulan Sari (2025)        | <p>Judul Penelitian : Pengembangan Framework Tata Kelola Akses <i>Multi-Tenant</i> untuk Mitigasi Ancaman Insider di Cloud Publik.</p> <p>Metode : <i>Research and Development (R&amp;D)</i> berbasis <i>Design Science Research (DSR)</i> dengan evaluasi komparatif terhadap RBAC dan ABAC pada lingkungan simulasi <i>OpenStack multi-tenant</i>.</p> <p>Tools/Software : <i>OpenStack</i>, <i>RBAC</i>, <i>ABAC</i>, <i>Markov Chain</i>, <i>STRIDE</i>, <i>Apache Benchmark</i>, <i>Splunk</i>, <i>Zero Trust Architecture</i>.</p> <p>Hasil : Framework yang dikembangkan mengintegrasikan kontrol akses, evaluasi perilaku kontekstual, dan mitigasi adaptif berbasis <i>trust score</i> untuk mendeteksi ancaman insider pada lingkungan cloud publik multi-tenant. Hasil evaluasi menunjukkan framework memiliki akurasi deteksi dan efisiensi respons yang lebih baik dibandingkan RBAC dan ABAC, meskipun menghasilkan peningkatan overhead yang masih berada dalam batas operasional yang dapat diterima</p> |
| 5   | Fajar Krismahendra Bintang Pangestu, et al. (2023) | <p>Judul Penelitian : Pengembangan Model Kontrol Akses Berbasis Atribut (ABAC) yang Selaras dengan NIST untuk Keamanan Rekam Medis Elektronik pada Lingkungan <i>Hybrid Cloud</i>.</p> <p>Metode : Studi literatur, perancangan model arsitektur berbasis XACML, dan analisis skenario uji.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

| No  | Penulis & Tahun                             | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
|-----|---------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                                         | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|     |                                             | <p>Tools/Software : Standar NIST, Hybrid Cloud, XACML.</p> <p>Hasil : Model ABAC yang dirancang mampu menerapkan kebijakan akses secara granular dan dinamis berdasarkan atribut subjek, objek, dan lingkungan. Pendekatan ini dinilai lebih fleksibel dibandingkan RBAC dalam mengelola kontrol akses pada lingkungan hybrid cloud serta mendukung kepatuhan terhadap standar NIST.</p>                                                                                                                                                                                                                                                                                                                                                                                    |
| 6   | Samuel Nugraha & Dian W. Chandra (2023)     | <p>Judul Penelitian : Peningkatan Keamanan Database pada Layanan Azure Melalui Metode <i>Multi-Tenant</i> Dengan Pendekatan <i>Separate Database</i>.</p> <p>Metode : mplementasi arsitektur <i>multi-tenant</i> dengan pendekatan <i>separate database</i> pada layanan <i>Microsoft Azure</i> menggunakan <i>Azure App Service</i> dan <i>Azure SQL Database</i>.</p> <p>Tools/Software : Microsoft Azure, Azure App Service, Azure SQL Database.</p> <p>Hasil : Pendekatan <i>separate database</i> meningkatkan keamanan data antar tenant karena setiap tenant memiliki basis data yang terpisah sehingga mengurangi risiko kebocoran data. Selain itu, penggunaan <i>Azure App Service</i> memberikan efisiensi biaya dibandingkan <i>Azure Virtual Machine</i>..</p> |
| 7   | Yoga Agung Prasetya & Danny Manongga (2024) | <p>Judul Penelitian : <i>Role-Based Access Control</i> (RBAC) untuk Sistem Otorisasi Terpusat Berbasis Flask: Studi Kasus PT.XYZ</p> <p>Metode : Studi literatur, identifikasi masalah, perancangan sistem, implementasi, dan pengujian sistem menggunakan pendekatan RBAC.</p> <p>Tools/Software : Python Flask, PostgreSQL, RBAC, PL/pgSQ.</p> <p>Hasil : Penelitian menghasilkan sistem otorisasi terpusat berbasis RBAC yang menyederhanakan pengelolaan hak akses dengan mengelompokkan pengguna berdasarkan jabatan. Pendekatan ini meningkatkan efisiensi administrasi hak akses, mempermudah pemeliharaan sistem, serta lebih</p>                                                                                                                                   |

| No  | Penulis & Tahun            | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
|-----|----------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                        | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|     |                            | sesuai diterapkan pada organisasi dengan struktur jabatan yang jelas dibandingkan pemberian hak akses secara langsung kepada setiap pengguna.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 8   | Arun Kumar Akuthota (2025) | Penelitian menunjukkan bahwa implementasi RBAC pada lingkungan cloud modern mampu meningkatkan keamanan, efisiensi administrasi, dan kepatuhan regulasi. Integrasi <i>AI</i> , <i>Machine Learning</i> , serta <i>Zero Trust Architecture</i> membuat RBAC lebih adaptif dalam mendeteksi ancaman, mengoptimalkan manajemen peran, dan mendukung tata kelola keamanan cloud berskala besar.                                                                                                                                                                                                                                                                                                                                                                                             |
| 9   | Alexander (2026)           | Judul Penelitian : Studi Perbandingan Model Keamanan Data pada Cloud Computing.<br>Metode : <i>Systematic Literature Review</i> (SLR) terhadap berbagai studi keamanan data terindeks.<br>Tools/Software : Algoritma enkripsi (AES, RSA, ECC) serta mekanisme kontrol akses RBAC dan ABAC.<br>Hasil : Memberikan analisis komprehensif mengenai efektivitas model RBAC dan ABAC dalam melindungi privasi data pada lingkungan <i>multi-tenant</i> serta merumuskan strategi mitigasi risiko terhadap ancaman keamanan data cloud.                                                                                                                                                                                                                                                       |
| 10  | Liliana et al. (2024)      | Judul Penelitian : <i>Exploration of Software as a Service (SaaS) as a Project Management Tools</i><br>Metode : <i>Design Science Research</i> (DSR), studi literatur, perancangan sistem, implementasi menggunakan Laravel, serta pengujian <i>Black Box</i> dan kuesioner pengguna.<br>Tools/Software : <i>Laravel 8.0</i> , <i>SaaS</i> , <i>Multi-Tenant</i> , <i>Multi-Schema Database</i> , <i>Cloudflare DNS</i> , <i>Google Calendar</i> , <i>Speech-to-Text</i> .<br>Hasil : Penelitian berhasil mengembangkan aplikasi project management berbasis SaaS dengan arsitektur multi-tenant menggunakan pendekatan multi-schema sehingga setiap tenant memiliki database yang terpisah. Hasil pengujian menunjukkan 77,8% pengguna menyatakan konsep SaaS memberikan fleksibilitas |

| No  | Penulis & Tahun             | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
|-----|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                         | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|     |                             | penggunaan sistem, sedangkan 76,16% pengguna menyatakan fitur yang tersedia membantu proses manajemen proyek secara efektif.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
| 11  | Rishi Kumar Sharma (2025)   | <p>Judul Penelitian : Multi-tenant Architectures in Modern Cloud Computing: A Technical Deep Dive</p> <p>Metode : Analisis teknis mendalam (<i>technical deep dive</i>) dan tinjauan studi kasus ekstensif terhadap komponen inti arsitektur <i>multi-tenant</i>.</p> <p>Tools/Software : Ekosistem infrastruktur AWS (Lambda, ECS, Aurora Serverless, S3, IAM, KMS), protokol OAuth 2.0, serta model analitik AI dan NLP.</p> <p>Hasil : Penelitian menunjukkan bahwa arsitektur multi-tenant mampu meningkatkan efisiensi penggunaan sumber daya, menjaga isolasi data antar tenant, serta meningkatkan keamanan melalui penerapan IAM, RBAC, OAuth 2.0, KMS, dan mekanisme enkripsi tenant. Implementasi tersebut juga meningkatkan skalabilitas sistem dan menurunkan biaya operasional dibandingkan arsitektur single-tenant.</p> |
| 12  | Mukul Gupta et al. (2024)   | <p>Judul Penelitian : Exploring the Impact of Software as a Service (SaaS) on <i>Human Life</i>.</p> <p>Metode : Analisis kualitatif dampak SaaS terhadap operasional bisnis</p> <p>Tools/Software : Layanan komputasi awan AWS dan Azure (fokus pada solusi keamanan).</p> <p>Hasil : Penelitian menunjukkan bahwa implementasi SaaS meningkatkan skalabilitas bisnis melalui fleksibilitas infrastruktur, efisiensi biaya, kemudahan integrasi, otomatisasi proses, serta peningkatan kolaborasi. Meskipun demikian, keberhasilan implementasi SaaS tetap dipengaruhi oleh aspek keamanan data, integrasi sistem, dan kesiapan organisasi.</p>                                                                                                                                                                                       |
| 13  | Antony R.R. Panguraj (2025) | Judul Penelitian : Systematic Approach to Security Testing in Multi-Tenant Cloud Systems                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |

| No  | Penulis & Tahun          | Keterangan                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|-----|--------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                      | (3)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|     |                          | <p>Metode :. Kerangka kerja pengujian keamanan sistematis (<i>Security Testing Framework</i>)</p> <p>Tools/Software : Lingkungan <i>cloud-native</i> (kontainer, <i>microservices</i>, <i>serverless</i>), <i>pipeline</i> CI/CD, serta mekanisme akses RBAC dan ABAC.</p> <p>Hasil : Penelitian menghasilkan kerangka kerja security testing untuk sistem cloud multi-tenant yang menekankan penerapan <i>continuous monitoring</i>, <i>fine-grained access control</i>, RBAC, ABAC, data <i>isolation</i>, enkripsi, serta integrasi keamanan ke dalam proses <i>DevSecOps</i>. Pendekatan tersebut mampu meningkatkan deteksi dini terhadap kerentanan dan mengurangi risiko kebocoran data maupun akses lintas tenant.</p> |
| 14  | Ramanan Hariharan (2025) | <p>Judul Penelitian : <i>Zero Trust Security in Multi-Tenant Cloud Environments</i></p> <p>Metode :. Kerangka kerja keamanan <i>Zero Trust</i> untuk otorisasi dan isolasi data</p> <p>Tools/Software : IAM, AWS KMS, Azure Key Vault, <i>Open Policy Agent</i> (OPA).</p> <p>Hasil : Penelitian menunjukkan bahwa penerapan <i>Zero Trust</i> pada lingkungan cloud multi-tenant meningkatkan keamanan melalui prinsip <i>never trust</i>, <i>always verify</i>, autentikasi berkelanjutan (<i>continuous authentication</i>), <i>tenant-scoped</i> RBAC, <i>fine-grained</i> ABAC, <i>identity federation</i>, serta evaluasi akses berbasis konteks secara <i>real-time</i>.</p>                                            |

Berikut adalah tabel gap penelitian dan Novelty yang digunakan dalam penelitian.

Tabel 2. 2 Gap & Novelty

| No  | Penulisan                | Gap Penelitian                                                  | Keterbaruan Penelitian                                         |
|-----|--------------------------|-----------------------------------------------------------------|----------------------------------------------------------------|
| (1) | (2)                      | (3)                                                             | (4)                                                            |
| 1   | M Sahyudi & Erliyan Redy | Penelitian mengkaji implementasi RBAC melalui <i>Systematic</i> | Mengembangkan mekanisme <i>Dynamic Context-Aware</i> RBAC yang |

| No  | Penulisan                                      | Gap Penelitian                                                                                                                                                                                                         | Keterbaruan Penelitian                                                                                                                                                                                      |
|-----|------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                                            | (3)                                                                                                                                                                                                                    | (4)                                                                                                                                                                                                         |
|     | Susanto (2025)                                 | <i>Literature Riview</i> sehingga belum mengembangkan mekanisme otorisasi yang mempertimbangkan konteks operasional pengguna pada lingkungan aplikasi <i>multi-tenant</i> .                                            | mengintegrasikan <i>role, ownership, tenant boundary, dan resource state</i> sebagai dasar pengambilan keputusan otorisasi pada sistem manajemen tugas SaaS multi-tenant.                                   |
| 2   | Loay Abdelrazek et al (2025)                   | Penelitian mengembangkan hybrid authorization pada lingkungan Intent-Based Management, namun belum mengimplementasikan mekanisme tersebut pada aplikasi manajemen tugas berbasis workspace multi-tenant.               | Menerapkan mekanisme Dynamic Context-Aware RBAC pada aplikasi manajemen tugas dengan validasi <i>ownership, tenant boundary, dan resource state</i> untuk mendukung otorisasi adaptif.                      |
| 3   | Saurabh Deochake & Vrushali Channpattan (2022) | Penelitian berfokus pada Identity and Access Management untuk hybrid cloud, namun belum membahas validasi hak akses berdasarkan hubungan penugasan pengguna terhadap sumber daya pada level aplikasi.                  | Menambahkan mekanisme <i>ownership validation</i> sehingga hak akses terhadap task ditentukan berdasarkan hubungan penugasan yang diberikan supervisor.                                                     |
| 4   | Azis Catur Laksono & Bety Wulan Sari (2025)    | Penelitian mengembangkan framework tata kelola akses berbasis <i>trust score</i> untuk mitigasi <i>insider threat</i> , namun belum mengintegrasikan validasi <i>workflow</i> operasional pada sistem manajemen tugas. | Mengintegrasikan <i>workflow transition validation</i> bersama <i>role, ownership, dan tenant boundary</i> dalam proses otorisasi untuk memastikan perubahan status task sesuai alur kerja yang ditetapkan. |
| 5   | Fajar Krismahendra et al. (2023)               | Penelitian menerapkan ABAC pada rekam medis elektronik sesuai standar NIST, namun belum mengimplementasikan mekanisme <i>Dynamic</i>                                                                                   | Mengembangkan mekanisme otorisasi berbasis <i>Dynamic Context-Aware RBAC</i> yang memanfaatkan konteks operasional pengguna tanpa                                                                           |

| No  | Penulisan                                   | Gap Penelitian                                                                                                                                                           | Keterbaruan Penelitian                                                                                                                                                                     |
|-----|---------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                                         | (3)                                                                                                                                                                      | (4)                                                                                                                                                                                        |
|     |                                             | <i>Context-Aware</i> RBAC pada aplikasi SaaS <i>multi-tenant</i> .                                                                                                       | mengadopsi kompleksitas penuh model ABAC.                                                                                                                                                  |
| 6   | Samuel Nugraha & Dian W. Chandra (2023)     | Penelitian berfokus pada isolasi basis data menggunakan pendekatan <i>separate database</i> , namun belum membahas pengendalian akses pada level aplikasi.               | Mengimplementasikan <i>tenant boundary validation</i> pada mekanisme otorisasi sehingga akses terhadap <i>workspace</i> dan <i>task</i> dibatasi sesuai divisi ( <i>tenant</i> ) pengguna. |
| 7   | Yoga Agung Prasetya & Danny Manongga (2024) | Penelitian menerapkan RBAC <i>statis</i> sehingga keputusan akses hanya ditentukan berdasarkan role pengguna.                                                            | Mengembangkan mekanisme <i>Dynamic Context-Aware</i> RBAC dengan keputusan akses berdasarkan <i>role</i> , <i>ownership</i> , <i>tenant boundary</i> , dan <i>resource state</i> .         |
| 8   | Arun Kumar Akuthota (2025)                  | Penelitian membahas tata kelola RBAC pada cloud modern secara konseptual, namun belum mengimplementasikan mekanisme otorisasi kontekstual pada aplikasi manajemen tugas. | Mengimplementasikan mekanisme <i>Dynamic Context-Aware</i> RBAC pada backend aplikasi SaaS <i>multi-tenant</i> melalui validasi konteks operasional pengguna secara real-time.             |
| 9   | Alexander (2026)                            | Penelitian membandingkan berbagai model keamanan <i>cloud</i> tanpa mengembangkan mekanisme kontrol akses pada level aplikasi.                                           | Mengembangkan mekanisme kontrol akses berbasis <i>Dynamic Context-Aware</i> RBAC yang disesuaikan dengan kebutuhan aplikasi manajemen tugas <i>multi-tenant</i> .                          |
| 10  | Liliana et al. (2024)                       | Penelitian mengembangkan aplikasi <i>project management</i> berbasis SaaS <i>multi-tenant</i> , namun belum membahas mekanisme otorisasi yang mempertimbangkan           | Menambahkan mekanisme <i>assignment-based ownership validation</i> dan <i>resource state</i> sebagai bagian dari proses otorisasi pada sistem manajemen tugas SaaS <i>multi-tenant</i> .   |

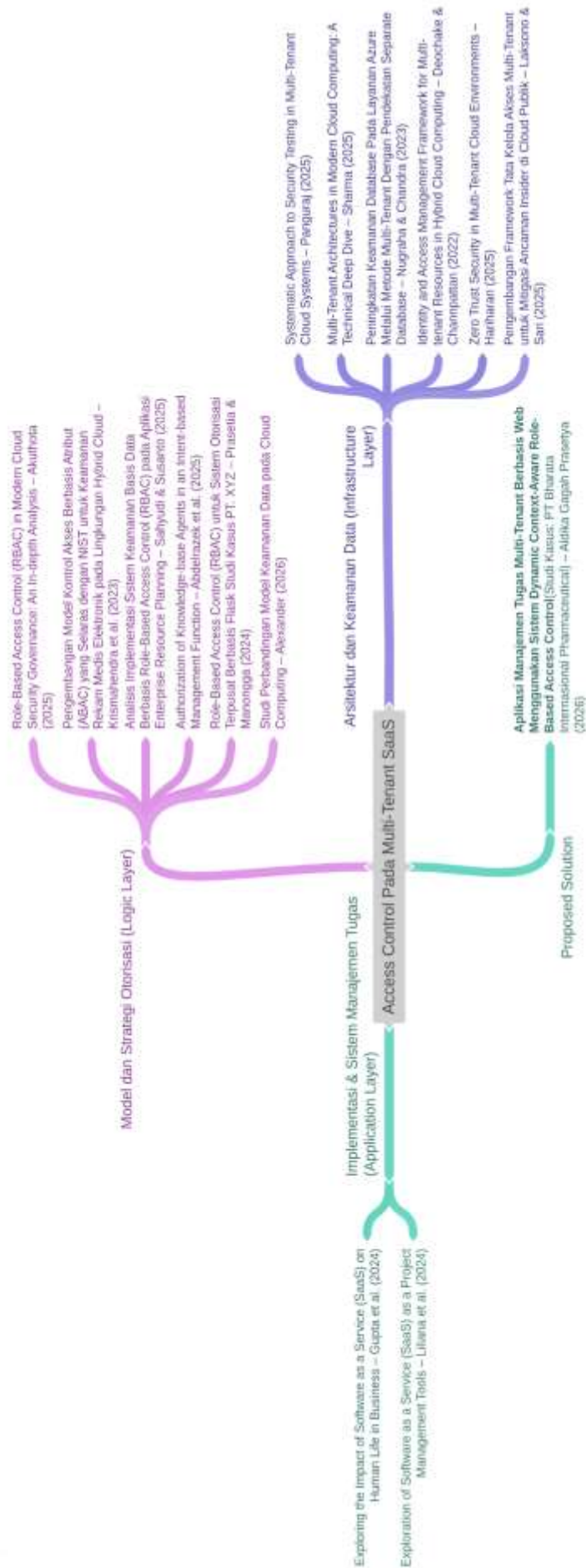
| No  | Penulisan                   | Gap Penelitian                                                                                                                                                                                                           | Keterbaruan Penelitian                                                                                                                                                                                                      |
|-----|-----------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (1) | (2)                         | (3)                                                                                                                                                                                                                      | (4)                                                                                                                                                                                                                         |
|     |                             | hubungan penugasan pengguna maupun validasi <i>workflow</i> .                                                                                                                                                            |                                                                                                                                                                                                                             |
| 11  | Rishi Kumar Sharma (2025)   | Penelitian membahas arsitektur <i>multi-tenant</i> pada tingkat infrastruktur <i>cloud</i> , namun belum mengimplementasikan mekanisme tenant isolation pada proses otorisasi aplikasi.                                  | Mengimplementasikan <i>tenant boundary validation</i> sebagai bagian dari mekanisme <i>Dynamic Context-Aware</i> RBAC untuk menjaga isolasi data antar-divisi.                                                              |
| 12  | Mukul Gupta et al. (2024)   | Penelitian berfokus pada dampak SaaS terhadap skalabilitas bisnis tanpa membahas mekanisme kontrol akses berbasis konteks.                                                                                               | Mengombinasikan arsitektur SaaS dengan mekanisme <i>Dynamic Context-Aware</i> RBAC untuk mendukung keamanan pengelolaan <i>task</i> pada lingkungan <i>multi-tenant</i> .                                                   |
| 13  | Antony R.R. Panguraj (2025) | Penelitian mengusulkan <i>framework security testing</i> pada <i>cloud multi-tenant</i> , namun belum mengimplementasikan mekanisme otorisasi kontekstual pada proses bisnis aplikasi.                                   | Mengintegrasikan <i>Dynamic Context-Aware</i> RBAC, <i>audit trail</i> , dan <i>real-time notification</i> untuk mendukung keamanan serta ketertelusuran aktivitas pada sistem manajemen tugas <i>multi-tenant</i> .        |
| 14  | Ramanan Hariharan (2025)    | Penelitian membahas <i>Zero Trust</i> , <i>tenant-scoped</i> RBAC, dan <i>fine-grained</i> ABAC pada tingkat konseptual, namun belum mengimplementasikan mekanisme tersebut pada aplikasi manajemen tugas berbasis SaaS. | Mengimplementasikan <i>Dynamic Context-Aware</i> RBAC yang mengadopsi prinsip <i>Zero Trust</i> melalui validasi <i>role</i> , <i>ownership</i> , <i>tenant boundary</i> , dan <i>resource state</i> pada backend aplikasi. |

Berdasarkan hasil pemetaan terhadap 14 penelitian terdahulu, dapat disimpulkan bahwa sebagian besar penelitian masih berfokus pada aspek keamanan

tertentu secara parsial, seperti kontrol akses berbasis RBAC atau ABAC, manajemen identitas (*Identity and Access Management*), isolasi data pada arsitektur *multi-tenant*, keamanan *cloud*, maupun penerapan *Zero Trust*. Meskipun beberapa penelitian telah mengembangkan mekanisme otorisasi yang lebih adaptif melalui pendekatan berbasis atribut, *trust-based access control*, maupun integrasi kecerdasan buatan, sebagian besar penelitian tersebut masih berorientasi pada keamanan infrastruktur, tata kelola identitas, atau kontrol akses umum sehingga belum secara spesifik mengakomodasi kebutuhan otorisasi dinamis pada level aplikasi manajemen tugas berbasis *multi-tenant*.

Selain itu, pendekatan RBAC yang digunakan pada penelitian terdahulu umumnya masih menjadikan peran (*role*) sebagai dasar utama dalam pengambilan keputusan akses. Pendekatan tersebut belum mempertimbangkan konteks operasional secara terpadu, seperti hubungan penugasan pengguna terhadap sumber daya (*assignment-based ownership*), batas isolasi antar-tenant (*tenant boundary*), serta status alur kerja (*workflow transition*) yang memengaruhi hak akses selama proses bisnis berlangsung. Kondisi tersebut berpotensi meningkatkan risiko akses lintas tenant maupun pemberian hak akses yang tidak lagi sesuai dengan kondisi operasional aplikasi.

Berdasarkan kesenjangan tersebut, penelitian ini mengembangkan mekanisme *Dynamic Context-Aware Role-Based Access Control* (RBAC) pada sistem manajemen tugas berbasis SaaS *multi-tenant*. Mekanisme yang diusulkan tetap menggunakan RBAC sebagai dasar pengendalian akses, namun melengkapinya dengan validasi *assignment-based ownership*, *workflow transition*, dan *tenant boundary* sehingga keputusan otorisasi dapat menyesuaikan konteks operasional secara dinamis. Dengan demikian, kebaruan penelitian ini terletak pada integrasi keempat komponen tersebut ke dalam mekanisme *Dynamic Context-Aware* RBAC untuk mendukung otorisasi yang lebih kontekstual sekaligus memperkuat isolasi data antar-tenant pada aplikasi manajemen tugas berbasis SaaS.



Gambar 2. 1 Mind Map

## B. Landasan Teori

Landasan teori pada penelitian ini mencakup tinjauan komprehensif mengenai mekanisme kontrol akses, strategi isolasi data pada arsitektur *multi-tenant*, prinsip keamanan *cloud*, serta teknologi pengembangan sistem yang mendukung operasional sistem manajemen tugas.

### a) Mekanisme Kontrol Akses (*Access Control*)

Kontrol akses merupakan mekanisme keamanan informasi yang berfungsi untuk mengatur izin penggunaan sistem melalui proses verifikasi identitas dan otorisasi hak akses. Dalam ekosistem *multi-tenant*, mekanisme ini berperan dalam mencegah akses ilegal antardivisi yang berbagi infrastruktur yang sama.

#### i. *Role-Based Access Control* (RBAC)

RBAC adalah model otorisasi yang menetapkan hak akses berdasarkan peran (*role*) pengguna dalam organisasi. Secara konseptual, RBAC menyederhanakan administrasi akses dengan memetakan hak akses ke dalam peran tertentu, sehingga setiap pengguna memperoleh akses yang sesuai dengan fungsinya [6]. Namun, RBAC memiliki keterbatasan intrinsik dalam menangani kondisi dinamis, di mana hak akses tidak dapat menyesuaikan secara otomatis berdasarkan status objek data [14].

#### ii. *Dynamic Context-Aware RBAC*

*Dynamic Context-Aware RBAC* merupakan pengembangan dari mekanisme RBAC yang mengintegrasikan informasi kontekstual ke dalam proses pengambilan keputusan otorisasi akses. Berbeda dengan RBAC statis yang hanya mendasarkan keputusan akses pada peran (*role*) pengguna, pendekatan ini juga memanfaatkan atribut dan hubungan (*relationships*) yang relevan sehingga keputusan akses dapat disesuaikan secara dinamis berdasarkan kondisi operasional sistem [15]. Dengan demikian, mekanisme ini mampu menghasilkan kontrol akses yang lebih granular, adaptif, serta mendukung penerapan prinsip *least privilege* pada lingkungan yang dinamis.

Selain memanfaatkan informasi kontekstual, pengambilan keputusan otorisasi juga dapat mempertimbangkan atribut yang berkaitan dengan pengguna (*subject*), sumber daya (*object*), maupun lingkungan (*environment*) sehingga keputusan akses dapat dilakukan secara *real-time* sesuai karakteristik permintaan akses yang diterima [16]. Pendekatan tersebut menjadikan *Dynamic Context-Aware RBAC* lebih sesuai diterapkan pada sistem berbasis *multi-tenant* yang memerlukan mekanisme kontrol akses adaptif terhadap perubahan kondisi operasional.

Dalam penelitian ini, konsep *Dynamic Context-Aware RBAC* diadopsi sebagai landasan dalam pengembangan mekanisme otorisasi pada sistem

manajemen tugas berbasis *multi-tenant*. Implementasi mekanisme tersebut disesuaikan dengan kebutuhan sistem melalui empat konteks utama, yaitu *role*, *ownership*, *resource state*, dan *tenant boundary*. Keempat konteks tersebut digunakan sebagai dasar dalam proses pengambilan keputusan otorisasi agar hak akses tidak hanya ditentukan berdasarkan peran pengguna, tetapi juga mempertimbangkan hubungan pengguna terhadap sumber daya, kondisi sumber daya, serta batas akses antar-*tenant*. Penjelasan mengenai implementasi keempat konteks tersebut dijabarkan lebih lanjut pada Bab III sebagai bagian dari perancangan mekanisme otorisasi penelitian

#### b) Arsitektur Multi-Tenancy dan Isolasi Data

*Multi-Tenancy* merupakan arsitektur perangkat lunak di mana satu *instance* aplikasi melayani banyak penyewa (*tenant*) secara simultan. Dalam penelitian ini, 10 divisi pada PT Bharata direpresentasikan sebagai penyewa yang memerlukan privasi data.

##### i. Isolasi pada Infrastruktur Cloud

Tantangan utama pada lingkungan *multi-tenant* adalah penggunaan infrastruktur bersama (*shared infrastructure*) yang berpotensi memicu kebocoran data antar-penyewa (*cross-tenant data leakage*) [5]. Oleh karena itu, diperlukan teknik isolasi pada layer komputasi untuk menjamin keamanan informasi serta mendukung skalabilitas sistem tanpa mengorbankan integritas data [4].

##### ii. Strategi *Single Database – Shared Schema*

Pendekatan *single-database – shared schema* merupakan strategi arsitektur *multi-tenancy* di mana seluruh penyewa (*tenant*) ditampung dalam satu basis data dan skema yang sama. Dalam penelitian ini, penyewa direpresentasikan oleh 10 divisi organisasi. Pendekatan ini lazim digunakan dalam pengembangan layanan *Software as a Service* (SaaS) karena menawarkan efisiensi dalam pemanfaatan sumber daya server, kemudahan pemeliharaan (*maintenance*), serta skalabilitas operasional yang optimal [4].

Dalam skema ini, pemisahan data antar-divisi dilakukan secara logis melalui penggunaan atribut khusus, seperti *tenant\_id* atau *division\_id*, yang disematkan pada setiap dokumen dalam koleksi MongoDB untuk memastikan isolasi data pada tingkat aplikasi.

Namun, penggunaan infrastruktur yang berbagi ruang penyimpanan (*shared storage*) secara inheren menimbulkan risiko keamanan signifikan, khususnya terkait potensi terjadinya kebocoran data antarpenyewa (*cross-*

*tenant data leakage*). Karakteristik arsitektur yang terpusat ini menyebabkan mekanisme keamanan sangat bergantung pada lapisan aplikasi. Jika mekanisme otorisasi hanya mengandalkan *Role-Based Access Control* (RBAC) yang bersifat statis, maka risiko akses ilegal lintas divisi akan meningkat, terutama ketika terjadi kegagalan validasi pada parameter identitas.

Oleh karena itu, implementasi *Dynamic Context-Aware* RBAC menjadi komponen krusial dalam penelitian ini. Pendekatan ini berperan sebagai mekanisme keamanan tambahan yang tidak hanya memverifikasi identitas dan peran pengguna, tetapi juga melakukan validasi berkelanjutan terhadap konteks permintaan akses, sehingga mampu memitigasi risiko kebocoran data pada arsitektur *single database shared* secara lebih komprehensif [5].

c) Paradigma Keamanan Zero Trust

i. Prinsip Never Trust, Always Verify

Paradigma Zero Trust menekankan penerapan autentikasi berkelanjutan (*continuous authentication*) terhadap setiap permintaan akses. Proses verifikasi didasarkan pada identitas pengguna dan konteks permintaan akses tersebut [8]. Pendekatan ini penting untuk memitigasi risiko ancaman internal (*insider threats*) di mana pengguna dalam satu divisi berupaya melampaui hak aksesnya terhadap data divisi lain [17].

d) Teknologi Pengembangan Sistem

Efisiensi operasional dari sistem otorisasi dinamis memerlukan dukungan teknologi yang memiliki kemampuan menangani proses secara bersamaan serta fleksibilitas dalam pengelolaan struktur data.

i. Bahasa Pemrograman Golang

Golang dipilih karena efisiensinya dalam menangani banyak proses secara paralel melalui fitur *goroutines*. Karakteristik ini sangat krusial saat sistem harus melakukan pengecekan konteks dinamis secara cepat dan berulang pada setiap permintaan akses, tanpa menurunkan performa aplikasi secara keseluruhan.

ii. Basis Data NoSQL MongoDB

MongoDB menyediakan fleksibilitas melalui skema data yang bersifat *schema-less*. Karakteristik ini memungkinkan penyimpanan data tugas dengan atribut konteks yang bervariasi antardivisi tanpa memerlukan perubahan skema yang kompleks. Hal ini selaras dengan kebutuhan sistem manajemen proyek berbasis *Software as a Service* (SaaS) yang menuntut fleksibilitas dan adaptivitas tinggi [13].

e) *Clean Architecture*

Pengembangan sistem berbasis Golang dalam penelitian ini menerapkan prinsip *clean architecture* untuk memastikan kode program memiliki struktur yang modular serta meminimalkan ketergantungan terhadap *framework* eksternal.

i. Pemisahan Lapisan Sistem

*Clean Architecture* mengorganisasi kode perangkat lunak ke dalam lapisan konsentris yang memiliki tanggung jawab masing-masing. Disesuaikan dengan alur kerja sistem otorisasi, implementasi ini membagi arsitektur sistem ke dalam empat lapisan utama :

- 1) Lapisan *Delivery/Handler*, yang bertanggung jawab dalam menangani dan memformat komunikasi protocol HTTP.
- 2) Lapisan *Use Case/Service*, yang berperan sebagai pusat eksekusi logika bisnis, termasuk algoritma *Dynamic Context-Aware* RBAC.
- 3) Lapisan *Repository* yang bertanggung jawab dalam mengelola interaksi dan kueri data ke basis data MongoDB.
- 4) Lapisan *Entity/Domain* yang mendefinisikan struktur dan aturan data dalam sistem [18].

Pemisahan lapisan ini memastikan bahwa logika otorisasi pada sistem *multi-tenant* tetap terpisah dari mekanisme akses data, sehingga tidak bergantung langsung pada implementasi database.

ii. Kemudahan Pemeliharaan dan Kolaborasi.

Penerapan arsitektur ini penting dalam pengembangan sistem berskala besar yang melibatkan lebih dari satu pengembang. Struktur lapisan yang jelas tidak hanya mempermudah pemahaman alur kerja sistem, tetapi juga mendukung proses pemeliharaan (*maintenance*) serta pengembangan fitur di masa mendatang tanpa mengganggu stabilitas sistem secara keseluruhan [18].

f) *JSON Web Token (JWT)* pada Skema *Zero Trust*

Untuk mendukung paradigma *Zero Trust* yang meminimalkan kepercayaan otomatis, sistem menggunakan *JSON Web Token (JWT)* sebagai mekanisme autentikasi dan otorisasi yang bekerja secara *stateless*.

i. Validasi Identitas *Stateless*

JWT digunakan karena kemampuannya dalam memverifikasi identitas pengguna pada setiap permintaan akses tanpa memerlukan penyimpanan data sesi di sisi server. Sifat *stateless* ini selaras dengan karakteristik performa Golang yang efisien dalam menangani beban kerja tinggi, di

mana validasi dilakukan secara berkelanjutan melalui *Header HTTP Authorization:Bearer* [8][18].

ii. Optimasi Payload untuk Isolasi Tenant

JWT memungkinkan penyisipan klaim identitas penting seperti *user\_id*, *role*, dan terutama *tenant\_id*. Informasi ini memungkinkan algoritma *Dynamic Context-Aware* RBAC untuk memverifikasi hak akses terhadap sumber daya divisi secara cepat pada setiap permintaan akses, tanpa memerlukan kueri basis data secara berulang [19]. Pendekatan ini meningkatkan keamanan isolasi data lintas penyewa serta efisiensi proses otorisasi.

h. Teknologi Kontainerisasi (Docker)

Docker merupakan platform kontainerisasi yang mencerminkan pergeseran dari virtualisasi berbasis perangkat keras menuju virtualisasi berbasis sistem operasi [20]. Teknologi ini memungkinkan aplikasi berjalan dalam lingkungan terisolasi dengan memanfaatkan kernel sistem operasi dari mesin induk, sehingga meningkatkan efisiensi penggunaan sumber daya komputasi [20] [21]. Untuk mendukung kinerja, Docker memanfaatkan fitur kernel Linux berupa *namespacing* untuk isolasi proses dan *control groups* untuk pengelolaan alokasi sumber daya secara independen [21]. Karakteristik ini menghasilkan lingkungan yang ringan dan konsisten, sehingga aplikasi dapat berjalan secara seragam di berbagai infrastruktur [20]

i. Arsitektur dan Komponen Inti

Sistem docker bekerja melalui integrasi beberapa komponen utama dalam mengelola siklus hidup aplikasi [20]. Proses pembangunan aplikasi dimulai dari *Docker Image*, yaitu *template read-only (immutable)* yang tersusun atas lapisan (*layers*) dan didefinisikan melalui *Dockerfile* [20] [22]. Image ini kemudian dijalankan sebagai *Docker Container*, yaitu instance aplikasi yang dieksekusi dalam lingkungan terisolasi [22]. Pengelolaan aplikasi dapat dioptimalkan dengan Menyusun instruksi *Dockerfile* secara efisien untuk mengurangi jumlah lapisan serta menghapus data yang tidak diperlukan [22]. Untuk mendukung skalabilitas, alat orkestrasi seperti *Docker Compose* digunakan untuk mengelola aplikasi *multi-kontainer* secara terintegrasi [22].

ii. Mekanisme Keamanan dan Isolasi-Tenant

Dalam sistem *multi-tenant*, Docker menyediakan lapisan isolasi infrastruktur yang penting untuk menjaga stabilitas operasional [22]. Setiap kontainernya berjalan dalam lingkungan terpisah dengan sistem berkas dan *variable* lingkungan yang terisolasi, sehingga mencegah gangguan antar kontainer [22]. Keamanan dapat diperkuat melalui prinsip *Zero Trust*, di mana isolasi jaringan memungkinkan penerapan *mikro-segmentasi* dengan

membatasi komunikasi antar layanan hanya pada jalur yang telah ditentukan [22]. Dengan demikian, kombinasi isolasi pada tingkat infrastruktur dan mekanisme otorisasi pada tingkat aplikasi mampu meningkatkan keamanan sistem secara menyeluruh [20], [22].