

PERANCANGAN DAN PENGEMBANGAN SISTEM MONITORING DAN APPROVAL ANGGARAN PURCHASE REQUEST BERBASIS ARSITEKTUR MICROSERVICE TERINTEGRASI ERP CUSTOM

Khusnul Fajri Rhomadon
Magister Ilmu Komputer, Universitas Esa Unggul
Arjuna Utara no.9, Kebon Jeruk, Jakarta 11510
khusnul@student.esaunggul.ac.id

ABSTRAK

Sistem Enterprise Resource Planning (ERP) yang banyak diadopsi perusahaan kerap menunjukkan keterbatasan signifikan, khususnya dalam modul budgeting dan mekanisme persetujuan (approval) untuk Purchase Request (PR). Sistem ERP konvensional yang masih mengandalkan struktur tunggal seringkali mengalami kesulitan dalam pembaruan data secara real-time, kurangnya visibilitas, serta proses persetujuan yang lambat dan rentan kesalahan. Kekurangan ini menjadi kendala utama dalam pengelolaan anggaran serta monitoring pengeluaran yang efektif, terutama pada proses PR yang memerlukan transparansi dan akurasi tinggi. Penelitian ini mengajukan desain inovatif alur persetujuan dan pemantauan anggaran berbasis arsitektur microservice, sebagai solusi untuk meningkatkan fleksibilitas, skalabilitas, serta efisiensi operasional sistem secara menyeluruh. Dengan pendekatan mikroservis, fungsi-fungsi kritikal dipecah menjadi layanan-layanan independen yang mampu bekerja secara terdistribusi dan real-time, seperti layanan PR, budgeting, approval, dan reporting. Hal ini memungkinkan pengelolaan anggaran yang lebih dinamis dan responsif terhadap perubahan kebijakan, notifikasi persetujuan yang otomatis, serta pemantauan konsumsi anggaran yang transparan dan terintegrasi, menyediakan laporan analitis akurat untuk pengambilan keputusan strategis. Implementasi arsitektur ini diharapkan mampu mengatasi keterbatasan sistem ERP konvensional, menyediakan mekanisme approval yang terpadu dan adaptif, serta memberikan dampak positif signifikan terhadap pengendalian biaya dan tata kelola keuangan perusahaan.

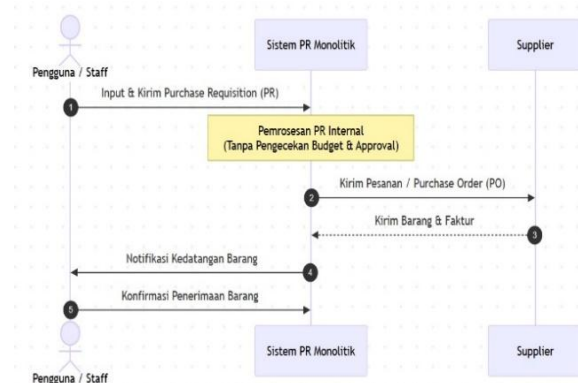
Kata kunci : *Microservice, Purchase Request, Budgeting, Approval, ERP, Monitoring Anggaran*

1. PENDAHULUAN

Sistem Enterprise Resource Planning (ERP) merupakan tulang punggung operasional berbagai perusahaan dalam mengelola sumber daya dan proses bisnis secara terintegrasi[1]. Namun, banyak implementasi ERP yang masih mengandalkan sistem monolitik tradisional yang memiliki keterbatasan dalam hal fleksibilitas skalabilitas[2], [3], [4], terutama dalam modul budgeting dan approval pada proses Purchase Request (PR). Kekurangan ini menyebabkan pengelolaan anggaran menjadi kurang optimal[2], [4], [5], sulit untuk melakukan monitoring pengeluaran secara real-time, serta menghambat transparansi dan kepatuhan terhadap anggaran yang telah ditetapkan [6], [7]. Keterbatasan ini menjadi kendala utama dalam pengelolaan anggaran serta monitoring pengeluaran yang efektif, terutama pada proses Purchase Request (PR) yang memerlukan transparansi dan akurasi tinggi.

Penelitian ini bertujuan untuk merancang dan mengembangkan sistem monitoring dan approval anggaran Purchase Request (PR) yang lebih fleksibel, skalabel, dan efisien. Untuk mencapai tujuan tersebut, rencana penyelesaian masalah yang diajukan adalah mengimplementasikan desain alur persetujuan dan pemantauan anggaran berbasis arsitektur microservice, sehingga memungkinkan pengembangan, pengujian, deployment, dan pemeliharaan modul secara independen[7], [8]. Penerapan microservice pada modul budgeting dan

approval dapat merancang alur proses yang lebih modular dan responsif, seperti pemisahan layanan penyimpanan anggaran, validasi pengeluaran, notifikasi persetujuan, dan pelaporan konsumsi anggaran yang berjalan secara paralel dan terkomunikasi secara efisien[6], [9], [10].



Gambar 1. Sistem pr di custom erp

Desain alur sistem monitoring anggaran dan approval PR berbasis microservice ini mencakup langkah-langkah:

a. Input dan Verifikasi PR:

Service PR menerima permintaan pembelian dan melakukan validasi awal terhadap data yang dimasukkan.

- b. Pengecekan Anggaran:
Service budgeting secara real-time memeriksa ketersediaan anggaran terkait dan memperbarui saldo anggaran setelah PR masuk.
- c. Workflow Approval:
Service approval mengelola proses persetujuan berjenjang sesuai dengan aturan bisnis, mengirimkan notifikasi kepada pengguna yang berwenang, serta menangani keputusan approve atau reject [10], [11], [12].
- d. Monitoring dan Pelaporan:
Service reporting mengakumulasi data konsumsi anggaran dan status PR untuk disajikan dalam dashboard yang memudahkan pengawasan dan audit internal secara transparan[12].

Dengan desain alur berbasis microservice tersebut, sistem ERP diharapkan dapat meningkatkan kecepatan proses, mengurangi *coupling* antar modul, dan memberikan kemudahan pengembangan fitur baru tanpa mengganggu keseluruhan sistem [8], [9], [13], sekaligus mendukung skalabilitas fungsional dan menjaga kestabilan sistem di masa depan[6], [9]. Desain ini menyediakan kerangka kerja yang lebih adaptif dan responsif, memungkinkan pengelolaan anggaran yang lebih efektif serta meningkatkan transparansi dan akuntabilitas dalam proses pengadaan di lingkungan ERP.

2. TINJAUAN PUSTAKA

Bab ini menyajikan tinjauan pustaka mengenai konsep-konsep kunci yang mendasari penelitian ini, termasuk sistem Enterprise Resource Planning (ERP), arsitektur Microservice, serta relevansinya dalam pengelolaan Purchase Request (PR), monitoring anggaran, dan alur persetujuan.

2.1. Enterprise Resource Planning (ERP)

Sistem Enterprise Resource Planning (ERP) merupakan tulang punggung operasional bagi banyak perusahaan untuk mengelola sumber daya dan proses bisnis secara terintegrasi. ERP dirancang untuk mengintegrasikan berbagai fungsi bisnis, seperti keuangan, sumber daya manusia, produksi, dan rantai pasok, ke dalam satu sistem terpadu [9]. Namun, implementasi ERP kerap menunjukkan keterbatasan signifikan, terutama dalam modul budgeting dan mekanisme persetujuan (approval) yang belum dioptimalkan. Sistem ERP tradisional seringkali mengandalkan arsitektur monolitik, yang memiliki kendala dalam hal fleksibilitas dan skalabilitas, menghambat pengelolaan anggaran yang optimal, pemantauan pengeluaran real-time, transparansi, dan kepatuhan terhadap anggaran. Literatur lain juga menyoroti pentingnya faktor keberhasilan kritis dalam implementasi pasca-ERP [1], serta strategi manajemen perubahan untuk implementasi ERP yang sukses [2].

2.2. Arsitektur Microservice

Arsitektur microservice adalah pendekatan modern yang efektif untuk mengatasi keterbatasan

sistem monolitik konvensional, [14]. Pendekatan ini memecah fungsi-fungsi sistem menjadi layanan-layanan kecil yang mandiri, berjalan independen, dan berkomunikasi melalui Application Programming Interface (API), [8]. Keunggulan utama arsitektur microservice meliputi peningkatan fleksibilitas, skalabilitas, serta kemudahan dalam pengembangan, pengujian, dan pemeliharaan tanpa mengganggu sistem secara keseluruhan, [15]. Dengan microservice, setiap layanan dapat dikembangkan, diuji, dan di-deploy secara independen, memungkinkan evolusi sistem sesuai kebutuhan bisnis. Penelitian ini mengusulkan desain inovatif alur persetujuan dan pemantauan anggaran berbasis arsitektur microservice sebagai solusi untuk meningkatkan fleksibilitas, skalabilitas, dan efisiensi operasional sistem ERP.

2.3. Purchase Request (PR) dan Pengelolaan Anggaran

Proses Purchase Request (PR) merupakan bagian krusial dalam siklus pengadaan perusahaan yang memerlukan transparansi dan akurasi tinggi. Dalam konteks ERP, pengelolaan PR yang terintegrasi dengan modul budgeting dan approval sangat penting untuk pengendalian biaya dan tata kelola keuangan yang efektif. Keterbatasan dalam modul budgeting dan approval pada sistem ERP konvensional seringkali menyebabkan pengelolaan anggaran menjadi kurang optimal dan sulit untuk melakukan monitoring pengeluaran secara real-time. Sebuah sistem berbasis microservice dirancang untuk memiliki fungsi-fungsi seperti Input dan Verifikasi Purchase Request, Budget Availability Check secara real-time untuk mencegah overbudget, serta Monitoring and Reporting untuk menyediakan dashboard transparan.

2.4. Alur Persetujuan (Approval Workflow)

Alur persetujuan atau approval workflow adalah mekanisme penting dalam banyak proses bisnis, termasuk PR dan pengelolaan anggaran. Dalam sistem yang kompleks, alur persetujuan seringkali memerlukan jenjang (tiered approval) dan notifikasi otomatis. Keterbatasan sistem ERP tradisional seringkali terletak pada mekanisme approval yang belum terpadu dan adaptif. Dengan arsitektur microservice, layanan approval dapat diatur untuk mengelola proses persetujuan berjenjang, memberikan notifikasi kepada pengguna berwenang, serta mengelola keputusan approve atau reject sesuai aturan bisnis. Modul Workflow Persetujuan Monitoring Budget dapat mengelola alur persetujuan secara otomatis, mencakup mekanisme approval maupun rejection, yang dilengkapi dengan pengiriman notifikasi melalui email secara real-time.

2.5. Persyaratan Fungsional dan Non-Fungsional dalam Sistem Microservice

Pengembangan sistem berbasis microservice untuk monitoring dan approval anggaran PR memerlukan identifikasi persyaratan yang jelas:

- a. Persyaratan Fungsional
Mencakup Purchase Request Input and Verification, Budget Availability Check, Tiered Approval Workflow, serta Monitoring and Reporting. Modul utama yang mendukung ini meliputi Input dan Verifikasi Purchase Request, Workflow Persetujuan Monitoring Budget, serta Konsumsi Anggaran dan Pelaporan.
- b. Persyaratan Non-Fungsional
Meliputi Security (otentikasi dan otorisasi ketat), Performance (waktu respons rendah, stabilitas tinggi), Maintainability and Extensibility (kemudahan pemeliharaan dan pengembangan fitur baru), dan Scalability (kemampuan menyesuaikan kapasitas layanan secara elastis). Implementasi ketersediaan tinggi (high availability) juga merupakan aspek krusial, didukung oleh redundansi, load balancing, strategi database yang handal, monitoring dan recovery otomatis, serta isolasi kesalahan (fault tolerance).

2.6. Teknologi Pendukung

Dalam implementasi arsitektur microservice, pemilihan teknologi sangat penting. Penelitian ini menggunakan .NET Core sebagai platform pengembangan microservice karena sifatnya yang modular, ringan, dan cross-platform, mendukung pembuatan layanan independen yang responsif dan mudah diintegrasikan. Untuk penyimpanan data, Firebird dipilih sebagai database relasional open-source dengan konsumsi sumber daya rendah, performa cepat, ringan, serta dukungan integrasi yang baik dengan .NET Core, menjamin keamanan dan konsistensi transaksi data anggaran.

Secara keseluruhan, tinjauan pustaka ini menunjukkan bahwa adopsi arsitektur microservice dapat secara signifikan mengatasi tantangan yang dihadapi oleh sistem ERP monolitik dalam mengelola Purchase Request, monitoring anggaran, dan alur persetujuan, sehingga meningkatkan efisiensi operasional dan tata kelola keuangan perusahaan.

3. METODE

3.1. Persyaratan Bisnis

Sistem Enterprise Resource Planning (ERP) berperan penting sebagai tulang punggung integrasi sumber daya dan proses bisnis di berbagai perusahaan[16]. Namun, sistem ERP tradisional yang masih bersifat monolitik sering mengalami keterbatasan dalam hal fleksibilitas, skalabilitas, serta responsivitas, khususnya dalam modul budgeting dan approval pada proses Purchase Request (PR). Pengelolaan anggaran yang kurang optimal dan minimnya kemampuan monitoring real-time menyebabkan hambatan dalam transparansi dan kepatuhan terhadap anggaran yang telah ditetapkan, sehingga berdampak pada efisiensi proses bisnis dan pengambilan keputusan.

Beberapa kebutuhan utama dari sistem Purchase Request dan budgeting dengan metode microservice yang diidentifikasi meliputi:

- a. Real-time Data Update
Pengguna membutuhkan pembaruan data anggaran dan status PR secara instan untuk memastikan keputusan dapat dibuat tepat waktu.
- b. Email Notification
Sistem harus mampu mengirimkan notifikasi otomatis melalui email kepada pengguna terkait status pengajuan, persetujuan, dan perubahan penting lainnya agar informasi dapat diterima tepat waktu, terutama bagi pengguna yang tidak selalu aktif mengakses sistem secara langsung.
- c. User Interface Friendliness
Desain antarmuka yang intuitif dan mudah digunakan meningkatkan pengalaman pengguna serta produktivitas[17].
- d. Scalability and Modularity
Sistem harus mampu beradaptasi dengan perubahan kebutuhan bisnis dan dapat dikembangkan secara modular tanpa mengganggu operasi layanan lain[11], [17].

3.2. Persyaratan Fungsional

Berlandaskan pendekatan arsitektur microservice, sistem diharapkan memiliki fungsi-fungsi sebagai berikut:

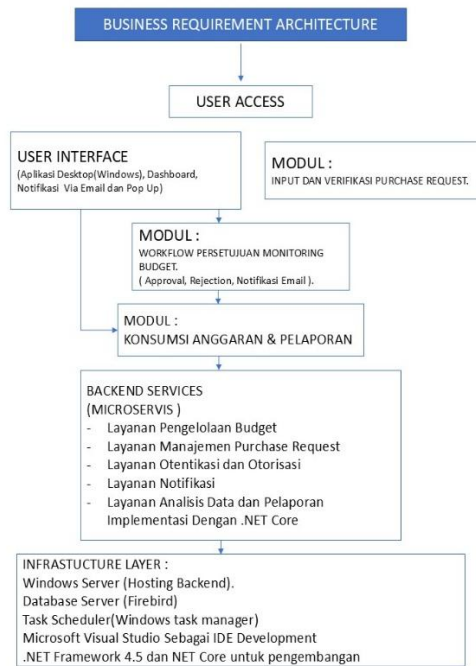
- a. Purchase Request Input and Verification
Modul PR harus mampu menerima serta melakukan validasi data awal secara mandiri, memastikan informasi yang diterima akurat dan lengkap sebelum diproses ke tahap selanjutnya.
- b. Budget Availability Check
Layanan budgeting harus memeriksa dan memperbarui saldo anggaran secara real-time ketika PR diajukan, mencegah overbudget dan meningkatkan pengendalian keuangan[13].
- c. Tiered Approval Workflow
Layanan approval mengatur proses persetujuan berjenjang, memberikan notifikasi kepada pengguna berwenang, serta mengelola keputusan approve atau reject sesuai aturan bisnis[15].
- d. Monitoring and Reporting
Service reporting harus mengumpulkan data konsumsi anggaran dan status PR, menyediakan dashboard transparan untuk memudahkan pengawasan dan audit internal.

3.3. Persyaratan Non-Fungsional

Selain fungsi utama, sistem juga harus memenuhi sejumlah kebutuhan non-fungsional berikut:

- a. Security
Sistem harus memastikan keamanan data dan akses dengan mekanisme autentikasi dan otorisasi yang ketat guna melindungi kerahasiaan dan integritas informasi anggaran dan transaksi.
- b. Performance
Kemampuan sistem untuk menangani permintaan dengan waktu respons rendah dan stabilitas tinggi

- saat volume transaksi yang tinggi atau kegagalan modul tertentu terjadi [14]
- Maintainability and Extensibility**
Arsitektur microservice yang digunakan harus memudahkan pemeliharaan dan pengembangan fitur baru tanpa mengganggu layanan lain, memungkinkan evolusi sistem sesuai kebutuhan bisnis.
 - Scalability**
Sistem harus mampu menyesuaikan kapasitas pelayanan sesuai dengan jumlah pengguna dan volume transaksi secara elastis[18], [19]



Gambar 2. Arsitektur persyaratan bisnis

3.4. Business Requirement Architecture

Bagan ini menggambarkan kerangka kebutuhan fungsional dan teknis dari sebuah sistem informasi yang dirancang untuk mengoptimalkan pengelolaan proses permintaan pembelian (purchase request) secara komprehensif. Sistem tersebut mencakup siklus penuh mulai dari tahap input data dan verifikasi permintaan, mekanisme alur persetujuan yang terstruktur, hingga pengawasan penggunaan anggaran serta penyajian pelaporan yang akurat dan real-time. Pendekatan ini bertujuan untuk memastikan transparansi, efisiensi, serta akuntabilitas dalam manajemen anggaran, sekaligus mendukung pengambilan keputusan berbasis data secara tepat waktu. Adopsi arsitektur microservices dan teknologi .NET Core memperkuat fleksibilitas dan skalabilitas sistem dalam menghadapi dinamika kebutuhan bisnis yang kompleks.

3.5. User Access dan User Interface

Pengguna sistem diberikan kemudahan akses melalui berbagai saluran interaksi yang dirancang

untuk menunjang fleksibilitas dan efektivitas pengguna dalam menjalankan tugasnya.

- Akses dapat dilakukan melalui aplikasi desktop yang dikembangkan berbasis platform Windows, yang menawarkan antarmuka pengguna yang responsif dan integrasi mendalam dengan lingkungan lokal.
- Dashboard berbasis web yang menyediakan informasi terkini secara real-time serta fitur interaktif yang mendukung pemantauan dan pengelolaan proses bisnis secara terpadu.
- Sistem juga menerapkan mekanisme pemberitahuan melalui email dan notifikasi pop-up dalam aplikasi, yang berfungsi sebagai media komunikasi proaktif untuk memastikan pengguna mendapatkan informasi penting secara cepat dan tepat waktu, sehingga mendukung kelancaran alur kerja dan pengambilan Keputusan.

3.6. Modul-Modul Utama

Modul-modul utama dalam sistem ini dirancang untuk mengakomodasi kebutuhan fungsional secara komprehensif dan terstruktur.

- Modul Input dan Verifikasi Purchase Request memberikan kemampuan bagi pengguna untuk melakukan penginputan data permintaan pembelian secara sistematis serta melakukan validasi atas keberlangsungan dan keakuratan informasi tersebut.
- Modul Workflow Persetujuan Monitoring Budget berperan dalam mengelola alur persetujuan secara otomatis, mencakup mekanisme approval maupun rejection, yang dilengkapi dengan pengiriman notifikasi melalui email guna memberikan informasi terkini terkait status permohonan kepada pemangku kepentingan secara real-time.
- Modul Konsumsi Anggaran dan Pelaporan bertanggung jawab dalam memantau serta mengendalikan penggunaan anggaran, sekaligus menyediakan laporan analitis yang komprehensif sebagai basis pengambilan keputusan strategis.

3.7. Layanan backend

Dalam arsitektur sistem ini diimplementasikan dengan pendekatan microservices yang memungkinkan pemisahan fungsi-fungsi inti ke dalam komponen-komponen layanan yang bersifat mandiri dan terdistribusi. Pendekatan ini menawarkan keunggulan berupa kemampuan skalabilitas yang tinggi, isolasi kesalahan yang efektif, dan fleksibilitas dalam pengembangan serta pemeliharaan sistem. Adapun fungsi-fungsi kritikal yang dikelola melalui microservices meliputi:

- pengelolaan anggaran secara terstruktur dan akuntabel,
- manajemen permintaan pembelian dengan mekanisme yang dinamis dan terintegrasi,

- c. layanan autentikasi dan otorisasi pengguna yang menjamin keamanan dan kontrol akses berdasarkan peran,
- d. pengiriman notifikasi sebagai sarana komunikasi proaktif antar subsistem maupun pengguna,
- e. layanan analisis data dan penyusunan laporan yang berfungsi sebagai basis informasi strategis bagi pengambilan Keputusan

3.8. Aplikasi Arsitektur Microservice

Arsitektur microservice adalah solusi modern yang efektif untuk mengatasi keterbatasan sistem monolitik konvensional, terutama dalam mengelola proses bisnis kompleks seperti monitoring dan approval anggaran Purchase Request (PR). Dengan membagi fungsi sistem menjadi layanan mandiri yang berjalan independen dan berkomunikasi melalui API, pendekatan ini meningkatkan fleksibilitas, skalabilitas, serta mempercepat pengembangan, pengujian, dan pemeliharaan tanpa mengganggu sistem secara keseluruhan.

Pada bab ini, akan dibahas secara detail penerapan arsitektur microservice dalam membangun aplikasi yang mendukung siklus penuh proses PR mulai dari input dan verifikasi data, pengecekan ketersediaan anggaran secara real-time, pengaturan workflow approval berjenjang, hingga pemantauan dan pelaporan konsumsi anggaran secara transparan. Sistem ini dirancang untuk meningkatkan efisiensi operasional serta menjaga keamanan dan integritas data. Selain itu, sistem menyediakan akses yang mudah melalui berbagai antarmuka pengguna yang responsif dan interaktif. Dengan pendekatan arsitektur microservice, sistem ini mampu menjadi sumber informasi strategis yang mendukung pengambilan keputusan secara cepat dan akurat dalam lingkungan ERP.

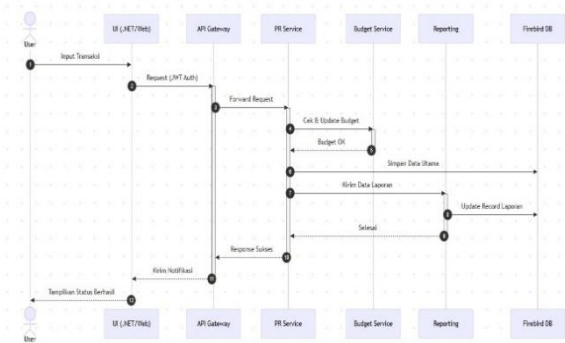
3.9. Seleksi Teknologi Arsitektur

Sistem monitoring dan approval anggaran Purchase Request ini menggunakan .NET Core sebagai platform pengembangan microservice karena kemampuannya yang modular, ringan, dan cross-platform, serta mendukung pembuatan layanan independen yang responsif dan mudah diintegrasikan.

Untuk penyimpanan data, dipilih Firebird, database relasional open-source konsumsi sumber daya yang sangat rendah, serta sifatnya yang ringan dan cepat. Ia juga mendukung banyak platform, memiliki komunitas yang aktif, dan merupakan solusi gratis serta open source yang hemat biaya. mampu mengelola transaksi secara konsisten. Firebird memiliki dukungan baik untuk integrasi dengan .NET, sehingga memastikan performa optimal dan keamanan data anggaran.

Kombinasi .NET Core dan Firebird mendukung skalabilitas sistem, keamanan data melalui autentikasi dan otorisasi ketat, serta kemudahan pengembangan dan pemeliharaan layanan secara terpisah tanpa mengganggu sistem keseluruhan.

Dengan arsitektur ini, sistem diharapkan mampu memberikan fleksibilitas, efisiensi, dan transparansi pengelolaan anggaran yang tinggi dalam ERP perusahaan.



Gambar 3. UML Arsitektur Sistem Purchase Request

3.10. Penentuan Teknologi Arsitektur

Dalam pengembangan sistem monitoring dan approval anggaran Purchase Request berbasis microservice ini, pemilihan teknologi menjadi aspek krusial untuk menjamin performa, skalabilitas, serta kemudahan pemeliharaan sistem secara keseluruhan. Sistem dirancang dengan teknologi-teknologi yang mendukung pendekatan microservice serta memenuhi kebutuhan akan fleksibilitas dan efisiensi operasional dalam lingkungan ERP.

Untuk platform pengembangan microservice, dipilih .NET Core karena sifatnya yang modular, ringan, dan cross-platform. .NET Core memungkinkan pembangunan layanan independen yang responsif serta mudah diintegrasikan dengan komponen lainnya, memberikan kemudahan dalam pengembangan, pengujian, dan deployment layanan secara terpisah tanpa mengganggu keseluruhan sistem.

Dalam hal penyimpanan data, Firebird dipilih sebagai database relasional open-source yang memiliki konsumsi sumber daya rendah serta performa yang cepat dan ringan. Selain itu, Firebird mendukung banyak platform dan memiliki komunitas aktif sehingga menjadi solusi hemat biaya namun tetap andal. Integrasi Firebird dengan .NET Core juga berjalan optimal sehingga menjamin keamanan dan konsistensi transaksi data anggaran yang disimpan.

Kombinasi teknologi .NET Core dan Firebird tidak hanya mendukung skalabilitas sistem, tetapi juga menjaga keamanan data melalui mekanisme autentikasi dan otorisasi yang ketat. Pendekatan ini memudahkan pengembangan dan pemeliharaan layanan secara mandiri, meningkatkan fleksibilitas serta transparansi pengelolaan anggaran dalam ERP perusahaan.

Dengan pemilihan teknologi tersebut, sistem diharapkan dapat memberikan solusi yang efisien dan Sistem dirancang untuk mampu beradaptasi dengan perubahan kebutuhan bisnis yang kompleks, sekaligus memastikan kestabilan operasional dan isolasi kesalahan antar layanan yang berjalan secara terdistribusi.

4. HASIL DAN PEMBAHASAN

4.1. Pembagian Microservice (Microservice Splitting)

Penerapan arsitektur microservice pada sistem monitoring dan approval anggaran Purchase Request (PR) menuntut pemecahan fungsi-fungsi utama sistem

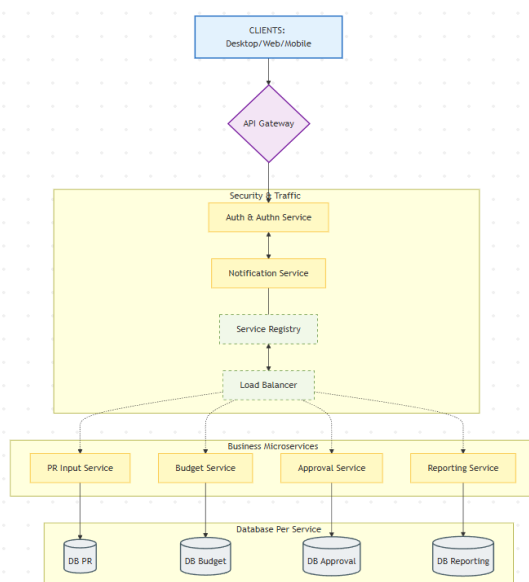
ke dalam layanan-layanan terpisah yang mandiri namun saling berinteraksi. Pendekatan ini memungkinkan setiap layanan dapat dikembangkan, diuji, dan di-deploy secara independen sehingga meningkatkan fleksibilitas dan skalabilitas sistem secara keseluruhan.

Tabel 1. Microservice Splitting

Kategori Layanan	Nama Layanan	Fungsi Layanan
Input & Verification	Purchase Request Input Service	Menerima dan memvalidasi data permintaan pembelian (PR) secara sistematis untuk memastikan keakuratan informasi.
Budgeting	Budget Availability Service	Memeriksa dan memperbarui saldo anggaran secara real-time untuk mencegah pengeluaran melebihi anggaran.
Approval Workflow	Tiered Approval Service	Mengelola proses persetujuan berjenjang, mengirim notifikasi otomatis, serta menangani keputusan approve/reject.
Monitoring & Reporting	Budget Monitoring Service	Memantau konsumsi anggaran dan menyediakan laporan analitis serta dashboard transparan untuk audit internal.
Notification	Notification Service	Mengirimkan email dan notifikasi pop-up sebagai komunikasi proaktif terkait status permohonan kepada pengguna.
Security	Authentication & Authorization Service	Menjamin keamanan data dan kontrol akses berbasis peran melalui mekanisme autentikasi dan otorisasi.
Data Analysis	Data Analysis & Reporting Service	Menyusun laporan dan analisis data sebagai dasar pengambilan keputusan strategis.

4.2. Arsitektur Keseluruhan

Mengingat kompleksitas kebutuhan bisnis perusahaan, sistem dipecah dan dibagi fungsi-fungsinya berdasarkan konsep desain arsitektur microservice. Sistem ini dibangun menggunakan .NET Core sebagai platform microservice yang modular, ringan, dan dapat dijalankan lintas platform. Firebird dipilih sebagai database relasional yang ringan, cepat, dan open source untuk penyimpanan data dengan performa yang optimal.



Gambar 4. Overall System Architecture

Arsitektur ini dirancang untuk menyediakan sistem yang scalable, fleksibel, dan menjamin keamanan data anggaran melalui autentikasi dan otorisasi yang ketat. Dengan pembagian layanan yang

mandiri namun saling berinteraksi, sistem dapat menangani beban kerja tinggi dan perubahan kebutuhan bisnis secara dinamis. Secara garis besar, arsitektur dibagi menjadi modul-modul utama sebagai berikut.

- Manajemen Layanan (Service Management)**
Sistem menggunakan mekanisme pendaftaran dan penemuan layanan (service registry and discovery) untuk mengelola microservice yang berjalan secara independen. Hal ini memungkinkan tiap layanan dapat terdaftar, ditemukan, dan dikonfigurasi secara dinamis tanpa perlu mengganggu layanan lain. Pengelolaan konfigurasi layanan secara terpusat dan dinamis meningkatkan efisiensi operasional dan kemudahan pemeliharaan sistem.
- Gateway Routing** Sebagai pintu masuk tunggal ke aplikasi, gateway berfungsi untuk memisahkan akses pengguna dengan layanan internal. Gateway mengatur rute permintaan API secara dinamis, mengelola autentikasi, pengontrolan lalu lintas (flow control), serta konversi protokol. Hal ini memungkinkan keamanan dan performa sistem terjaga sekaligus memudahkan pengelolaan akses pengguna.
- Modul Logika Bisnis Fungsi utama sistem monitoring dan approval anggaran** dipecah menjadi layanan microservice yang mandiri dan dapat dikembangkan terpisah. Misalnya:
 - Layanan PR untuk menerima dan memverifikasi permintaan pembelian.
 - Layanan Budgeting yang memeriksa ketersediaan anggaran secara real-time dan memperbarui saldo anggaran.
 - Layanan Approval yang mengatur alur persetujuan berjenjang sesuai aturan bisnis dan mengirim notifikasi persetujuan.

- Layanan Pelaporan yang mengakumulasi data dan menampilkan konsumsi anggaran melalui dashboard yang transparan.

Pemecahan ini mengurangi keterkaitan antar modul (low coupling), meningkatkan fleksibilitas pengembangan dan pemeliharaan tanpa perlu deploy ulang seluruh sistem.

- d. Modul Database Penyimpanan data dilakukan menggunakan Firebird sebagai database utama yang ringan dan cepat. Setiap layanan microservice menyimpan data di database yang terpisah, sehingga isolasi data antar layanan terjaga. Penggunaan cache in-memory (bila diperlukan) dapat meningkatkan performa akses data dan mempercepat respon sistem. Pengelolaan koneksi database menggunakan connection pool yang efisien untuk monitoring kinerja dan mencegah query lambat.

Dengan desain arsitektur seperti ini, sistem monitoring dan approval anggaran Purchase Request diharapkan memiliki skalabilitas tinggi, responsif terhadap perubahan kebutuhan bisnis, serta transparansi dan keamanan data yang terjamin.

4.3. Implementasi Ketersediaan Tinggi

Dalam rangka memastikan sistem monitoring dan approval anggaran Purchase Request (PR) berbasis microservice memiliki ketersediaan tinggi (high availability), beberapa strategi dan teknologi diterapkan untuk menjaga sistem tetap responsif dan dapat diakses secara kontinu meskipun terjadi kegagalan pada komponen tertentu. High availability menjadi aspek penting mengingat sistem ini digunakan sebagai fungsi kritikal dalam ERP yang memerlukan keandalan tinggi demi kelancaran proses bisnis.

4.4. Desain Microservice untuk High Availability

Arsitektur microservice yang diadopsi pada sistem ini memungkinkan pemecahan fungsi utama menjadi layanan-layanan terpisah yang independen namun saling berinteraksi. Dengan desain terdistribusi ini, kegagalan pada satu layanan tidak akan langsung berdampak pada layanan lain, sehingga sistem secara keseluruhan dapat tetap beroperasi normal tanpa downtime signifikan.

4.5. Redundansi dan Load Balancing

Setiap layanan microservice di-deploy secara redundan pada beberapa instance yang berjalan pada container atau virtual machine. Penggunaan load balancer di depan layanan memastikan distribusi beban lalu lintas secara merata dan juga menyediakan failover otomatis apabila terjadi gangguan pada salah satu instance layanan. Hal ini meningkatkan ketersediaan layanan dan menjaga performa sistem saat beban pengguna meningkat.

4.6. Database dan Penyimpanan Data

Penggunaan database Firebird dipilih tidak hanya karena kecepatan dan konsumsi sumber daya yang

rendah, tetapi juga karena kemampuannya mendukung transaksi yang konsisten. Strategi backup berkala dan replikasi database dapat diterapkan untuk mengurangi risiko kehilangan data serta memastikan data anggaran yang kritikal selalu tersedia, meskipun terjadi kegagalan server penyimpanan.

4.7. Monitoring dan Recovery Otomatis

Sistem dilengkapi dengan mekanisme monitoring yang memantau kesehatan setiap layanan secara real-time. Jika terdeteksi adanya gangguan atau kegagalan pada microservice tertentu, sistem dapat melakukan restart layanan secara otomatis atau memindahkan beban ke instance lain yang sehat, sehingga mempercepat proses recovery dan meminimalkan downtime.

4.8. Isolasi Kesalahan dan Fault Tolerance

Setiap layanan microservice diisolasi dengan baik sehingga kesalahan yang terjadi pada satu layanan tidak menyebar ke layanan lain (fault isolation). Pendekatan ini menjamin kestabilan sistem secara keseluruhan melalui manajemen kesalahan yang terstruktur, sehingga sistem tahan terhadap dinamika kebutuhan bisnis yang kompleks tanpa mengorbankan ketersediaan layanan.

4.9. Implementasi Ekstensibilitas

Implementasi sistem monitoring dan approval anggaran Purchase Request (PR) berbasis microservice menjadikan extensibility atau kemampuan sistem untuk berkembang dan beradaptasi sebagai salah satu aspek utama dalam desainnya. Dengan kebutuhan bisnis yang terus berubah dan berkembang, sistem dirancang agar mudah diperluas dengan fitur-fitur baru maupun integrasi layanan tambahan tanpa mengganggu fungsionalitas yang sudah berjalan.

5. KESIMPULAN

Implementasi sistem monitoring dan approval anggaran Purchase Request berbasis arsitektur microservice pada ERP perusahaan terbukti memberikan solusi efektif untuk mengatasi keterbatasan sistem monolitik tradisional. Dengan memecah fungsi-fungsi kritikal menjadi layanan-layanan independen yang berjalan secara terdistribusi dan real-time, sistem ini mampu meningkatkan fleksibilitas, skalabilitas, dan transparansi dalam pengelolaan anggaran.

Pemilihan teknologi .NET Core dan Firebird mendukung pengembangan layanan yang modular, ringan, serta efisien dalam hal pemeliharaan dan integrasi. Arsitektur microservice juga memungkinkan proses persetujuan (approval) yang adaptif dan notifikasi otomatis yang responsif, sehingga mendukung pengambilan keputusan yang tepat waktu dan akurat.

Sistem ini tidak hanya meningkatkan efisiensi operasional dan keamanan data, tetapi juga

memberikan kemudahan akses melalui berbagai antarmuka pengguna responsif. Dengan demikian, pendekatan desain ini berkontribusi signifikan dalam memperbaiki pengendalian biaya dan tata kelola keuangan perusahaan secara menyeluruh.

Ke depan, sistem ini memiliki potensi besar untuk dikembangkan lebih lanjut dengan fitur tambahan dan integrasi layanan baru, berkat kemampuan extensibility yang inheren dalam arsitektur microservice. Hal ini menjadikan solusi yang diusulkan sebagai fondasi robust untuk inovasi dan pengembangan ERP yang lebih adaptif terhadap dinamika bisnis.

DAFTAR PUSTAKA

- [1] Z. T. Butarbutar, P. W. Handayani, R. R. Suryono, and W. S. Wibowo, "Systematic literature review of Critical success factors on enterprise resource planning post implementation," 2023, *Cogent OA*. doi: 10.1080/23311975.2023.2264001.
- [2] A. M. Aladwani, "Change management strategies for successful ERP implementation," # MCB University Press, 2001. [Online]. Available: <http://www.emerald-library.com/ft>
- [3] Hideyuki. Takagi, *Proceedings : 2010 Second World Congress on Nature and Biologically Inspired Computing : NaBIC 2010, Kitakyushu, Japan, December 15-17, 2010*. IEEE, 2010.
- [4] T. T. Van Hau and J. Kuzic, "Change management strategies for the successful implementation of enterprise resource planning systems," in *Proceedings - 2nd International Conference on Knowledge and Systems Engineering, KSE 2010*, 2010, pp. 178–182. doi: 10.1109/KSE.2010.10.
- [5] N. Aniba, J. Siddiqi, and M. Elmosbahy, *TOWARDS THE CASE FOR INTRODUCING ENTERPRISE RESOURCE PLANNING AT NATIONAL OIL CORPORATION (NOC) OF LIBYA: A CASE STUDY*. 2019.
- [6] W. Tang, L. Wang, and G. Xue, "Design of Information System Architecture of Garment Enterprises Based on Microservices," in *Journal of Physics: Conference Series*, Institute of Physics Publishing, Mar. 2019. doi: 10.1088/1742-6596/1168/3/032128.
- [7] W. Yan and F. Shuai, "Application of Microservice Architecture in Commodity ERP Financial System," *International Journal of Computer Theory and Engineering*, vol. 14, no. 4, pp. 168–173, Nov. 2022, doi: 10.7763/IJCTE.2022.V14.1324.
- [8] V. Velepucha and P. Flores, "A Survey on Microservices Architecture: Principles, Patterns and Migration Challenges," *IEEE Access*, vol. 11, pp. 88339–88358, 2023, doi: 10.1109/ACCESS.2023.3305687.
- [9] Z. Zhang, "Function Expansion and Application Optimization of Enterprise Resource Planning (ERP) System Based on Cloud Computing," in *Procedia Computer Science*, Elsevier B.V., 2025, pp. 287–293. doi: 10.1016/j.procs.2025.04.206.
- [10] C. Lee, H. F. Kim, and B. G. Lee, "A Systematic Literature Review on the Strategic Shift to Cloud ERP: Leveraging Microservice Architecture and MSPs for Resilience and Agility," Jul. 01, 2024, *Multidisciplinary Digital Publishing Institute (MDPI)*. doi: 10.3390/electronics13142885.
- [11] B. S. Chohan, X. Xu, and Y. Lu, "MES Dynamic interoperability for SMEs in the Factory of the Future perspective," in *Procedia CIRP*, Elsevier B.V., 2022, pp. 1329–1335. doi: 10.1016/j.procir.2022.05.153.
- [12] B. Shah and S. Jain, "Hybrid Cloud Architectures for Multi-Modal AI Systems," 2025.
- [13] A. Taghinezhad-Niar, S. Pashazadeh, and J. Taheri, "Workflow scheduling of scientific workflows under simultaneous deadline and budget constraints," *Cluster Comput*, vol. 24, no. 4, pp. 3449–3467, Dec. 2021, doi: 10.1007/s10586-021-03314-3.
- [14] M. A. AL-Shboul, "Towards better understanding of determinants logistical factors in SMEs for cloud ERP adoption in developing economies," *Business Process Management Journal*, vol. 25, no. 5, pp. 887–907, Aug. 2019, doi: 10.1108/BPMJ-01-2018-0004.
- [15] S. Finney, "Stakeholder perspective on internal marketing communication: An ERP implementation case study," *Business Process Management Journal*, vol. 17, no. 2, pp. 311–331, Apr. 2011, doi: 10.1108/1463715111122365.
- [16] U. Sirin, S. Dwarkadas, and A. Ailamaki, "Performance characterization of HTAP workloads," in *Proceedings - International Conference on Data Engineering*, IEEE Computer Society, Apr. 2021, pp. 1829–1834. doi: 10.1109/ICDE51399.2021.00162.
- [17] M. Kaloudis, "Evolving Software Architectures from Monolithic Systems to Resilient Microservices: Best Practices, Challenges and Future Trends," 2024. [Online]. Available: www.ijacsa.thesai.org
- [18] G. R. Rajendiran, "Microservices Architecture for Optimizing Warehouse Management Systems in Global Supply Chains," *Journal of Artificial Intelligence, Machine Learning and Data Science*, vol. 1, no. 1, pp. 1388–1394, Mar. 2023, doi: 10.51219/jaimld/gautham-ram-rajendiran/314.
- [19] J. Lewis and M. Fowler, "Microservices," 2014.