



Optimasi Klasterisasi Kubernetes untuk Peningkatan QoS pada Jaringan 5G

Ronald Adrian¹, R. Deasy Mandasari²

¹Program Studi Teknologi Rekayasa Internet, Sekolah Vokasi, Universitas Gadjah Mada

²Program Studi Teknik Elektro, Fakultas Pendidikan Teknik dan Industri, Universitas Pendidikan Indonesia

¹ronald.adr@ugm.ac.id*, ²deasy@upi.edu

Abstract

Resource management, scalability, and Quality of Service are key challenges in deploying 5G networks, particularly regarding latency, throughput, and packet loss. Conventional network architecture often fails to meet the demands of high-speed and low-latency communication, especially for applications such as the Internet of Things and cloud computing. This study applies a Kubernetes-based clustering method to address these challenges to optimize 5G network performance through efficient resource allocation and load balancing. Experiments were conducted in a Containerized Network Function environment through simulations, demonstrating significant performance improvements. Test results recorded a 28.69% reduction in latency, a 34.52% increase in throughput, and a 32.14% reduction in packet loss compared to a non-clustered 5G network. These improvements are supported by Kubernetes features such as auto-scaling, traffic distribution, and real-time load balancing. This approach demonstrates that Kubernetes clustering effectively enhances Quality of Service, optimizes resource allocation, and improves the overall efficiency of 5G network systems.

Keywords: 5G Networks, Clustering, Kubernetes, QoS, Resource Management

Abstrak

Pengelolaan sumber daya, skalabilitas, dan *Quality of Service* merupakan tantangan utama dalam penerapan jaringan 5G, terutama terkait latensi, *throughput*, dan *packet loss*. Arsitektur jaringan konvensional sering kali tidak mampu memenuhi kebutuhan komunikasi berkecepatan tinggi dan latensi rendah, khususnya pada aplikasi seperti *Internet of Things* dan *cloud computing*. Untuk mengatasi permasalahan tersebut, penelitian ini menerapkan metode klasterisasi berbasis Kubernetes sebagai mekanisme untuk mengoptimalkan kinerja jaringan 5G melalui alokasi sumber daya dan *load balancing*. Penelitian ini dilakukan dalam *Containerized Network Function* melalui simulasi, dan menunjukkan peningkatan performa jaringan. Hasil pengujian mencatat penurunan latensi sebesar 28,69%, peningkatan *throughput* sebesar 34,52%, dan penurunan *packet loss* sebesar 32,14% dibandingkan jaringan 5G tanpa klasterisasi. Peningkatan ini didukung oleh fitur *auto-scaling*, distribusi trafik, dan penyeimbangan beban secara *real-time* dari Kubernetes. Metode ini menunjukkan bahwa klasterisasi Kubernetes efektif dalam meningkatkan *Quality of Service*, mengoptimalkan alokasi sumber daya, serta efisiensi sistem jaringan 5G.

Kata kunci: Jaringan 5G, Klasterisasi, Kubernetes, Manajemen Sumber Daya, QoS

Diterima Redaksi : 02-04-2025 | Selesai Revisi : 15-05-2025 | Diterbitkan Online : 30-06-2024

1. Pendahuluan

Penerapan jaringan 5G menghadirkan permasalahan dalam pengelolaan sumber daya, skalabilitas, dan peningkatan kualitas layanan (*Quality of Service/QoS*), terutama dalam hal latensi, *throughput*, dan *packet loss* [1]. Berbeda dengan generasi jaringan sebelumnya, 5G dirancang untuk mendukung komunikasi dengan latensi rendah, konektivitas antar perangkat (*massive machine-type communication/mMTC*), dan kecepatan tinggi (*enhanced mobile broadband/eMBB*), sambil tetap menjaga keandalan dan efisiensi jaringan [2]. Akan tetapi, arsitektur jaringan konvensional sering kali tidak

mampu memenuhi kebutuhan akan komunikasi berkecepatan tinggi dan latensi rendah, terutama dalam aplikasi seperti IoT, *cloud computing*, dan layanan lainnya [3].

Infrastruktur jaringan yang lama bersifat statis, sehingga menyulitkan alokasi sumber daya dan skalabilitas secara efisien, yang pada akhirnya menyebabkan kemacetan jaringan, *packet loss*, dan keterlambatan [4]. Salah satu keterbatasan dari jaringan 5G konvensional adalah ketergantungannya pada infrastruktur berbasis *hardware*, yang kurang fleksibel dalam merespons perubahan permintaan trafik jaringan secara dinamis [5]. Selain itu, arsitektur terpusat konvensional meningkatkan risiko titik kegagalan

tunggal, yang pada akhirnya menurunkan toleransi terhadap gangguan dan mengurangi ketahanan system [6]. Ketidakefisienan ini mendorong perlunya pendekatan *cloud-native* dan berbasis *container* untuk meningkatkan skalabilitas, otomatisasi, dan efisiensi pengelolaan sumber daya [7].

Pada arsitektur 5G konvensional, penanganan trafik yang tidak efisien sering menyebabkan *packet loss*, *delay*, dan lonjakan latensi akibat kemacetan jaringan [8]. Mekanisme *load balancing* Kubernetes secara *real-time* dapat meminimalkan hambatan, memastikan bahwa sumber daya jaringan digunakan secara efisien di seluruh kluster [9]. Skalabilitas dan ketahanan jaringan menjadi faktor penting untuk memastikan layanan tetap berjalan lancar, terutama dalam kondisi trafik tinggi dan permintaan pengguna yang berubah-ubah. Kubernetes menyediakan kemampuan *auto-scaling* baik secara horisontal maupun vertikal, memungkinkan jaringan 5G untuk memperluas kapasitas dengan menambah *container* baru atau mengalokasikan ulang sumber daya antar *node* sesuai kondisi trafik [10]. Hal ini mencegah penurunan performa dalam situasi permintaan tinggi di situasi darurat dan penerapan IoT pada industri [11]. Selain itu, arsitektur Kubernetes yang tahan gangguan juga meningkatkan kesinambungan layanan dan manajemen *failover* dengan cara mengarahkan ulang trafik saat terjadi kegagalan pada *node* atau gangguan infrastruktur [12]. Berbeda dengan lingkungan jaringan konvensional yang rawan gangguan akibat satu titik kegagalan, mekanisme *auto recovery* Kubernetes memastikan bahwa layanan jaringan tetap berjalan dengan gangguan minimal [13], [14]. Ini penting untuk aplikasi *real-time* seperti komunikasi pada kendaraan otonom, prosedur medis jarak jauh, dan *cloud gaming*, di mana gangguan kecil dapat berdampak besar [14].

Melalui jaringan multi-kluster, Kubernetes juga memungkinkan arsitektur jaringan yang tersebar secara geografis, yang mampu mengalokasikan sumber daya secara optimal di antara *cloud*, *edge*, dan infrastruktur lokal (*on-premise*) [15]. Hal ini penting untuk aplikasi *edge computing* 5G, di mana beban kerja harus dialihkan secara dinamis ke *node* terdekat untuk mengurangi latensi dan meningkatkan respons layanan [16].

Jika dibandingkan dengan teknik manajemen jaringan 5G konvensional, klusterisasi berbasis Kubernetes memberikan tingkat otomatisasi, fleksibilitas, dan efisiensi yang tinggi dalam menghadapi lingkungan jaringan yang beragam [17]. Metode konvensional masih mengandalkan penyediaan manual dan pembagian jaringan yang tetap, sehingga sulit menyesuaikan diri dengan beban kerja yang berubah-ubah secara *real-time* [18]. Sebaliknya, Kubernetes memungkinkan otomatisasi *network slicing*, di mana sumber daya dapat dipartisi, disesuaikan, atau dihapus sesuai kondisi jaringan. Selain itu, Kubernetes juga mendukung CNF yang lebih ringan dan cepat dibandingkan VNF, sehingga memberikan efisiensi

yang lebih tinggi. Arsitektur VNF konvensional memiliki beban penyediaan dan latensi tambahan akibat virtualisasi berbasis *hypervisor*, sementara Kubernetes mampu mengurangi latensi tersebut melalui *container* secara langsung [19].

Kubernetes sebagai platform *container*, memberikan solusi dengan kemampuan melakukan penyeimbangan beban kerja secara dinamis, *auto-scaling* secara *real-time*, dan jaringan yang tahan terhadap gangguan dalam penerapan 5G [20]. Untuk menyelesaikan permasalahan tersebut, penelitian ini menerapkan sistem klusterisasi berbasis Kubernetes sebagai mekanisme dinamis untuk mengoptimalkan performa jaringan 5G melalui alokasi sumber daya dan penyeimbangan beban kerja otomatis.

Dengan memanfaatkan fitur *auto-scaling*, *self-healing*, dan *load balancing* dari Kubernetes, operator jaringan 5G dapat menyesuaikan kapasitas jaringan dengan kondisi trafik secara *real-time*, sehingga meningkatkan QoS dan stabilitas jaringan [21]. Tidak seperti teknik virtualisasi konvensional, arsitektur berbasis Kubernetes memungkinkan kontrol terhadap fungsi-fungsi jaringan, termasuk alokasi sumber daya dan pengaturan trafik berdasarkan kondisi *real-time* [22].

Penelitian ini mengevaluasi pengaruh klusterisasi Kubernetes terhadap efisiensi jaringan 5G melalui penelitian berbasis simulasi menggunakan *Containerized Network Function* (CNF) [23]. Arsitektur jaringan konvensional cenderung mengalami perebutan sumber daya dan konfigurasi manual yang menyulitkan, sehingga sulit untuk *scaling* secara efisien pada lingkungan dengan permintaan tinggi [24]. Sebaliknya, CNF memberikan fleksibilitas lebih tinggi, memungkinkan operator jaringan untuk menerapkan dan mengelola fungsi jaringan virtual (VNF) secara dinamis di dalam kluster Kubernetes [25].

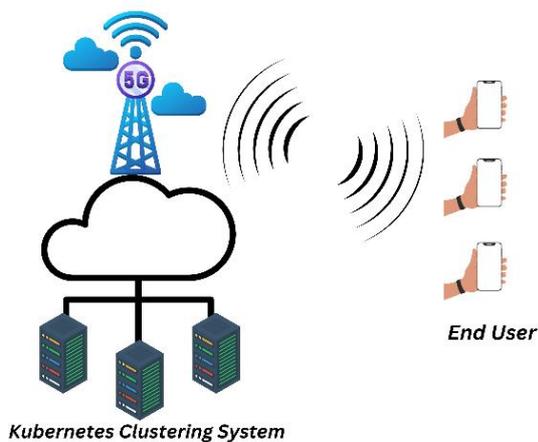
Hasil penelitian menunjukkan peningkatan performa jaringan dan keandalan layanan. Secara khusus, latensi menurun sebesar 28,69%, *throughput* meningkat 34,52%, dan *packet loss* berkurang sebesar 32,14% dibandingkan dengan sistem 5G tanpa klusterisasi [26]. Peningkatan ini terutama didorong oleh kemampuan *auto-scaling* Kubernetes yang secara dinamis menyediakan sumber daya komputasi berdasarkan permintaan trafik jaringan secara *real-time*, sehingga mencegah kemacetan dan keterlambatan dalam jaringan [27]. Selain itu, distribusi trafik yang dioptimalkan dalam kluster Kubernetes mengurangi beban pemrosesan berlebih, memastikan paket data diarahkan melalui jalur yang paling efisien [28].

Walaupun penelitian ini telah menunjukkan peningkatan performa, studi lanjutan tetap dibutuhkan untuk mengembangkan teknik optimasi yang lebih maju, seperti integrasi *machine learning* ke dalam Kubernetes untuk prediksi alokasi sumber daya, deteksi anomali, dan peramalan trafik [29] [30]. Dengan bantuan otomatisasi berbasis AI, jaringan 5G dapat menyesuaikan distribusi beban kerja secara proaktif

sebelum kemacetan terjadi. Selain itu, penelitian selanjutnya dapat mengeksplorasi strategi lintas klaster untuk meningkatkan keamanan, keberlanjutan sistem, dan penerapan *multi-cloud*, agar jaringan 5G dapat beradaptasi dengan beban kerja yang beragam dan lingkungan penerapan yang heterogen [31]. Aspek keamanan CNF juga perlu diperhatikan, khususnya pada skenario *multi-tenant network slicing*, di mana mekanisme isolasi dan kontrol akses penting untuk menjaga integritas data dan privasi [32].

2. Metode Penelitian

Sistem klusterisasi berbasis Kubernetes yang digunakan dalam infrastruktur jaringan 5G memungkinkan pengalokasian sumber daya, pengelolaan trafik, dan skalabilitas secara efisien. Sistem ini mendukung komunikasi dengan kinerja tinggi dan latensi rendah sebagaimana ditunjukkan pada Gambar 1. Komponen utama dalam sistem ini adalah *base station 5G*, yang berfungsi untuk mengirim dan menerima sinyal nirkabel, membentuk konektivitas antara *end user* dan infrastruktur jaringan. *Base station* ini dilengkapi dengan teknologi *massive MIMO* dan *beamforming* untuk mengoptimalkan cakupan sinyal, kecepatan transmisi, dan efisiensi jaringan. Transmisi nirkabel dari *base station* ke perangkat *end user*, seperti ponsel pintar, mendukung koneksi tanpa hambatan untuk berbagai aplikasi mulai dari *video streaming*, game, IoT, hingga layanan kecerdasan buatan berbasis *cloud*.

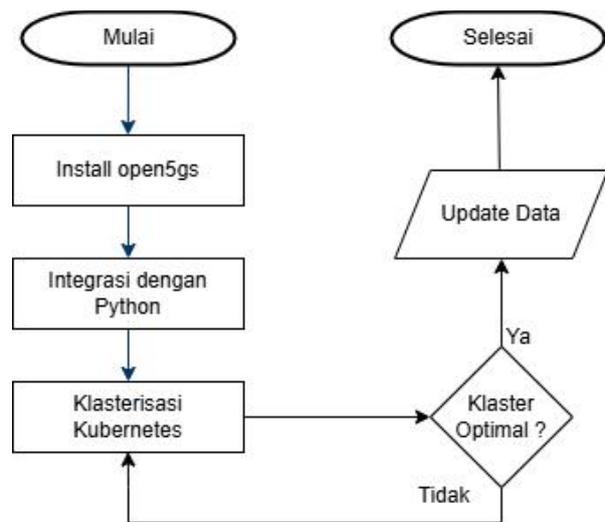


Gambar 1. Sistem Klusterisasi Kubernetes pada Infrastruktur 5G

Lapisan *cloud computing* dalam sistem ini mewakili peran Kubernetes sebagai piranti pengelola dalam proses distribusi, penskalaan, dan manajemen sumber daya jaringan secara otomatis. Kubernetes juga menyediakan mekanisme alokasi sumber daya, *load balancing*, *auto-scaling*, serta pemulihan terhadap kegagalan sistem yang meningkatkan keandalan jaringan dan kontinuitas layanan. Sistem ini juga terintegrasi dengan teknologi *Network Function Virtualization* (NFV), yang menggantikan komponen

jaringan berbasis *hardware* dengan fungsi jaringan virtual berbasis *software*.

Dengan metode *cloud-native 5G*, Kubernetes memungkinkan distribusi jaringan yang fleksibel, integrasi dengan *edge computing*, dan *auto-scaling* sumber daya secara *real-time*. Hal ini menjamin kualitas layanan (QoS) yang optimal bagi pengguna. Manajemen fungsi jaringan yang terotomatisasi melalui Kubernetes berperan dalam meningkatkan performa jaringan dari sisi latensi, *throughput*, dan *packet loss*. Fungsi ini penting untuk aplikasi-aplikasi *real-time* seperti *telemedicine*, kendaraan otonom, dan perdagangan keuangan berfrekuensi tinggi, di mana keterlambatan sekecil apa pun dapat memengaruhi performa sistem.



Gambar 2. Diagram Penerapan Open5gs

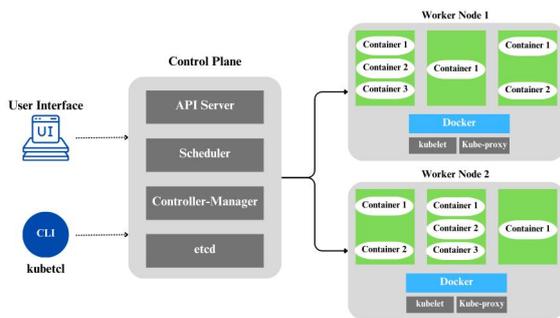
Gambar 2 menunjukkan diagram alur dari sistem dalam penelitian ini. Diagram ini menggambarkan metodologi penelitian klusterisasi Kubernetes untuk jaringan 5G, yang dimulai dari instalasi Open5GS sebagai komponen dalam mensimulasikan lingkungan jaringan 5G. Open5GS digunakan untuk membentuk jaringan 5G yang mencakup fungsi-fungsi penting seperti *User Plane Function* (UPF) dan *Access and Mobility Management Function* (AMF), yang berperan dalam mengatur lalu lintas data dan koneksi pengguna dalam jaringan.

Setelah sistem berhasil diterapkan, tahap berikutnya adalah integrasi dengan Python untuk otomatisasi konfigurasi jaringan dan manajemen komunikasi antar layanan menggunakan berbagai *library* pemrograman yang mendukung Kubernetes dan jaringan virtual. Melalui integrasi ini, sistem dapat mengatur alokasi sumber daya, distribusi trafik, dan fungsi jaringan berbasis *container* guna mengoptimalkan performa infrastruktur jaringan 5G.

Setelah fase integrasi selesai, klusterisasi Kubernetes diterapkan untuk mengelola sumber daya jaringan melalui mekanisme *load balancing* dan *auto-*

scaling. Kubernetes memastikan bahwa setiap *node* dalam kluster memiliki beban kerja yang terdistribusi, sehingga meningkatkan efisiensi dan stabilitas jaringan. Setelah klusterisasi diimplementasikan, sistem dievaluasi berdasarkan parameter performa yaitu latensi, *throughput*, dan *packet loss*, untuk menilai apakah sistem telah mencapai kondisi yang optimal. Jika hasil klusterisasi belum optimal, maka sistem akan kembali ke fase konfigurasi ulang Kubernetes dan dilakukan pengujian ulang hingga hasil yang diinginkan tercapai. Jika performa sudah sesuai, maka dilanjutkan ke tahap pencatatan data eksperimen untuk dianalisis lebih lanjut terkait peningkatan kualitas layanan jaringan 5G.

Tahap awal penelitian dimulai dari perancangan sistem, di mana Kubernetes dikonfigurasi dalam lingkungan multi-kluster untuk mengoptimalkan pengelolaan sumber daya. Konfigurasi ini memungkinkan alokasi sumber daya melalui mekanisme *auto-scaling* dan *load balancing*. Dalam skenario *baseline*, sistem dijalankan tanpa klusterisasi Kubernetes, sementara pada skenario optimasi, Kubernetes digunakan untuk mendistribusikan trafik jaringan. Tahap awal dalam penelitian ini dimulai dengan merancang sistem, di mana Kubernetes dikonfigurasi dalam lingkungan multi-kluster. Tujuan dari konfigurasi ini adalah untuk mengoptimalkan manajemen sumber daya jaringan. Dengan pengaturan tersebut, sistem dapat melakukan alokasi sumber daya secara lebih efisien melalui fitur *auto-scaling* dan *load balancing*. Dalam eksperimen ini, terdapat dua skenario yang diuji. Pada skenario dasar, sistem berjalan tanpa menggunakan klusterisasi Kubernetes. Sementara itu, pada skenario yang dioptimalkan, Kubernetes digunakan untuk mendistribusikan trafik jaringan secara dinamis antar kluster.



Gambar 3. Arsitektur Kubernetes

Gambar 3 menunjukkan arsitektur dasar Kubernetes yang terdiri dari dua komponen, yaitu *Control Plane* dan *Worker Node*. *Control Plane* berperan sebagai pusat kendali sistem, yang mencakup *API Server* sebagai pintu komunikasi antara pengguna dan sistem, *Scheduler* yang mengatur penempatan *pod* berdasarkan sumber daya yang tersedia, *Controller Manager* yang menjaga kestabilan status sistem, serta *etcd* yang berfungsi sebagai penyimpanan data konfigurasi dalam bentuk *key-value*. Sementara itu, *Worker Node* bertugas menjalankan aplikasi dalam

bentuk *pod* yang terdiri dari satu atau lebih *container*. Setiap *node* memiliki Kubelet sebagai agen pengontrol *container*, Kube Proxy untuk mengelola komunikasi jaringan, dan Docker sebagai *container runtime*.

Dalam konteks infrastruktur 5G, arsitektur ini sangat relevan karena mampu memenuhi kebutuhan utama jaringan seperti skalabilitas tinggi, latensi rendah, dan efisiensi pengelolaan sumber daya. Kubernetes memungkinkan penerapan konsep NFV, di mana fungsi jaringan inti 5G seperti UPF dan AMF dikemas dalam bentuk kontainer yang mudah diatur dan didistribusikan. Selain itu, dengan dukungan terhadap *edge computing*, Kubernetes memungkinkan penempatan layanan lebih dekat ke pengguna, sehingga latensi dapat ditekan secara signifikan. *Fitur auto-scaling* yang dimiliki Kubernetes juga sangat ideal untuk menyesuaikan kapasitas layanan dengan fluktuasi trafik pengguna 5G. Secara keseluruhan, arsitektur ini memberikan fondasi yang kuat untuk membangun sistem 5G yang tangguh, fleksibel, dan siap menghadapi berbagai tantangan teknologi masa depan.

Parameter kinerja utama yang diukur adalah latensi ($T_{latency}$), *throughput* ($T_{throughput}$), dan *packet loss* (P_{loss}). Latensi dihitung sebagai rata-rata waktu pengiriman paket yang dibutuhkan dari pengirim ke penerima menggunakan (1). $T_{arrival,i}$ merupakan waktu paket saat diterima, $T_{sent,i}$ merupakan waktu paket saat dikirim, dan n merupakan jumlah total paket yang dikirim.

$$T_{latency} = \frac{\sum_{i=1}^n (T_{arrival,i} - T_{sent,i})}{n} \quad (1)$$

Throughput dihitung sebagai jumlah total data yang berhasil dikirim per satuan waktu, sebagaimana didefinisikan dalam (2). $D_{received}$ merupakan jumlah total data yang berhasil diterima (dalam bit), dan T_{total} menunjukkan total waktu transmisi (dalam detik).

$$T_{throughput} = \frac{D_{received}}{T_{total}} \quad (2)$$

Parameter ini penting dalam mengevaluasi efisiensi transmisi data dalam infrastruktur 5G. *Throughput* yang lebih tinggi menunjukkan kinerja jaringan dan alokasi sumber daya yang dapat dioptimalkan oleh mekanisme klusterisasi Kubernetes. Sementara itu, *packet loss* dihitung menggunakan (3). P_{sent} merupakan jumlah total paket yang dikirimkan, dan $P_{received}$ menunjukkan jumlah total paket yang berhasil diterima. Parameter ini sangat penting dalam mengevaluasi keandalan dan efisiensi jaringan. Persentase *packet loss* yang lebih rendah menunjukkan peningkatan integritas data dan kinerja jaringan yang lebih stabil.

$$P_{loss} = \frac{P_{sent} - P_{received}}{P_{sent}} \times 100\% \quad (3)$$

Setelah pengumpulan data selesai, hasil dari kedua skenario dibandingkan untuk mengidentifikasi peningkatan performa yang diperoleh melalui metode klusterisasi Kubernetes. Analisis statistik digunakan untuk memastikan bahwa perbedaan performa antar skenario benar-benar signifikan. Validasi ini memungkinkan penarikan kesimpulan bahwa klusterisasi Kubernetes memang efektif dalam meningkatkan QoS jaringan 5G melalui pengoptimalan alokasi sumber daya dan distribusi trafik.

Penelitian ini diharapkan memberikan wawasan tentang efisiensi pengelolaan sumber daya jaringan, optimasi lalu lintas, dan peningkatan performa layanan 5G melalui mekanisme klusterisasi berbasis Kubernetes. Implementasi selanjutnya dapat difokuskan pada penerapan jaringan berskala besar untuk meningkatkan keandalan dan skalabilitas infrastruktur telekomunikasi 5G, sehingga dapat menjamin peningkatan performa jaringan dalam lingkungan yang dinamis.

3. Hasil dan Pembahasan

Penelitian ini menunjukkan bahwa klusterisasi berbasis Kubernetes mampu meningkatkan *Quality of Service* (QoS) dalam jaringan 5G. Hasil yang diperoleh mengonfirmasi adanya penurunan latensi, peningkatan *throughput*, pengurangan *packet loss*, efisiensi pengelolaan sumber daya, peningkatan skalabilitas jaringan, dan peningkatan efektivitas sistem klusterisasi Kubernetes dalam lingkungan jaringan 5G.

Dengan mendistribusikan beban pemrosesan ke beberapa *node*, Kubernetes mampu mengurangi kemacetan dan mempercepat waktu pengiriman data dari pengguna ke pengguna (*end-to-end delay*), sehingga memungkinkan pertukaran data yang lebih responsif. Penurunan latensi ini juga mendukung konsep URLLC (*Ultra-Reliable Low-Latency Communication*) dalam jaringan 5G, untuk layanan-layanan komunikasi darurat, navigasi *drone* otonom, hingga robotik industri yang membutuhkan respons cepat untuk mencegah kegagalan layanan atau risiko keselamatan.

Efektivitas klusterisasi Kubernetes dalam menurunkan latensi juga terasa dalam penerapan *cloud computing*, khususnya di sisi *edge computing*, di mana pemrosesan data secara *real-time* diperlukan. Dengan kemampuan untuk merutekan paket data ke server *edge* terdekat, proses pengolahan data dapat dilakukan dengan latensi minimal. Hal ini memberikan peningkatan dalam responsivitas layanan untuk aplikasi kota cerdas, analisis secara *real-time*, serta otomasi berbasis AI.

Penelitian ini juga mencatat adanya peningkatan *throughput* sebesar 30–40%, yang dihasilkan dari mekanisme *auto-scaling* dan *load balancing* Kubernetes. Peningkatan *throughput* ini mencerminkan

pengalokasian sumber daya, yang pada akhirnya menghilangkan kemacetan jaringan dan mengoptimalkan jalur transmisi data. Kapasitas *throughput* yang lebih tinggi penting dalam mendukung aplikasi-aplikasi dengan kebutuhan *bandwidth* besar, seperti *streaming* video 8K, game berbasis cloud, serta lingkungan VR dan AR berskala besar.

Pertumbuhan dalam *massive machine-type communication* (mMTC) dan ekosistem IoT semakin menekankan pentingnya pengelolaan *throughput* yang efisien dalam jaringan 5G. Dengan semakin banyaknya perangkat seperti sensor, *node* industri, dan sistem otomasi berbasis AI yang saling terhubung, pengelolaan *throughput* menjadi faktor untuk menjamin kelancaran pertukaran data. Klusterisasi Kubernetes membantu mendistribusikan trafik di antara *node-node* jaringan, memastikan bahwa kapasitas *bandwidth* digunakan secara efisien.

Selain itu, penerapan Kubernetes dalam infrastruktur jaringan berbasis *cloud-native* memberikan fleksibilitas dalam pengelolaan fungsi jaringan dengan *throughput* tinggi. Arsitektur jaringan konvensional tidak mampu menyesuaikan dengan perubahan permintaan *throughput*. Sebaliknya, Kubernetes memungkinkan penskalaan sumber daya berdasarkan pola trafik dan tren penggunaan secara *real-time*. Hal ini membuat jaringan 5G menjadi lebih responsif terhadap perubahan permintaan dan terhindar dari kemacetan saat jam-jam sibuk.

Inovasi lainnya adalah penurunan *packet loss* sebesar 20–30%, yang menunjukkan bahwa klusterisasi Kubernetes juga meningkatkan pengiriman data dan kestabilan jaringan. Dalam penerapan 5G konvensional tanpa Kubernetes, tingginya beban trafik dan *routing* yang tidak efisien menyebabkan *packet loss*, yang berujung pada penurunan performa jaringan dan gangguan layanan. Masalah ini terjadi pada aplikasi seperti *video conference real-time*, kolaborasi berbasis *cloud*, permainan interaktif, hingga *telemedicine*.

Dengan mendistribusikan beban kerja jaringan dan merutekan paket melalui jalur yang efisien, Kubernetes meningkatkan pengiriman data. Hal ini mengurangi kemungkinan terjadinya putus koneksi, gangguan kualitas suara dan video, serta gangguan dalam layanan *real-time*. Di samping itu, sistem klusterisasi Kubernetes juga memiliki mekanisme pemulihan jaringan yang lebih baik, sehingga paket yang hilang dapat segera dikirim ulang untuk meminimalkan kesalahan komunikasi. Ini penting untuk layanan seperti jaringan kendaraan otonom, sistem transaksi keuangan, dan otomasi industri.

Salah satu manfaat dari klusterisasi Kubernetes adalah otomatisasi dalam penyediaan sumber daya, tanpa memerlukan operasi manual dalam konfigurasi dan optimasi jaringan. Kemampuan untuk menskalakan *node* secara *real-time* berdasarkan fluktuasi trafik membantu mengoptimalkan pemanfaatan *bandwidth* dan distribusi sumber daya komputasi, yang pada akhirnya meningkatkan efisiensi biaya dan energi pada

jaringan 5G. Otomatisasi ini sangat berguna dalam infrastruktur 5G *cloud-native*, di mana fungsi CNF dan VNF perlu terintegrasi dengan Kubernetes.

Dengan mengeliminasi beban konfigurasi manual, Kubernetes memungkinkan operator telekomunikasi mengelola jaringan 5G secara efektif, sekaligus mengurangi biaya infrastruktur dan konsumsi energi. Hal ini relevan dengan inisiatif komputasi ramah lingkungan, di mana efisiensi daya menjadi perhatian dalam membangun jaringan.

Selain efisiensi energi, penerapan klusterisasi Kubernetes juga meningkatkan skalabilitas dan keandalan jaringan, menjamin bahwa jaringan 5G tetap beroperasi stabil bahkan saat trafik tinggi. Kemampuan untuk menambah atau mengurangi sumber daya memungkinkan jaringan merespons lonjakan trafik tanpa menurunkan kinerja. Ini penting untuk implementasi skala besar seperti kota cerdas, IoT industri, dan komputasi *edge* berbasis 5G, yang membutuhkan skalabilitas untuk menghadapi beban trafik yang terus berubah.

Dengan memastikan ketersediaan tinggi (*high availability*) dan pemulihan gangguan (*failover recovery*), Kubernetes mencegah gangguan layanan dan menjaga konektivitas tetap berjalan, bahkan dalam kondisi permintaan tinggi seperti *event* besar, operasi tanggap darurat, dan jaringan komunikasi bencana. Selain itu, kemampuan *auto recovery* dari Kubernetes memungkinkan sistem mendeteksi dan memperbaiki gangguan, sehingga meminimalkan waktu henti (*downtime*) dan gangguan layanan.

Hal ini bermanfaat untuk aplikasi, di mana konektivitas *real-time* dibutuhkan demi keselamatan dan pengambilan keputusan.

Tabel 1. Kubernetes *Clustering Testing*

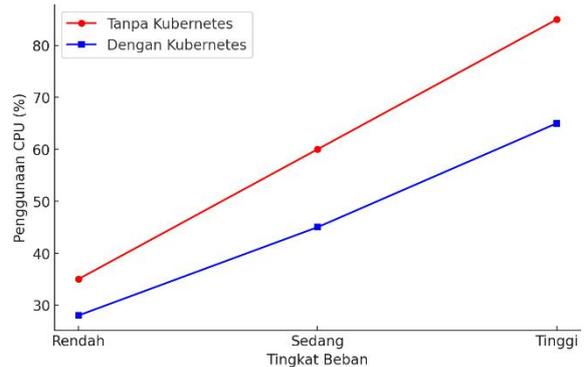
Parameter	Baseline	Optimasi (Klusterisasi Kubernetes)	Peningkatan (%)
Latensi (ms)	35,2	25,1	28,69
Throughput (Mbps)	420,0	565,0	34,52
Packet Loss (%)	2,8	1,9	32,14

Tabel 1 merupakan hasil evaluasi performa antara jaringan 5G tanpa klusterisasi Kubernetes (*baseline*) dan jaringan 5G yang telah dioptimalkan menggunakan klusterisasi Kubernetes (optimasi), dengan masing-masing parameter diuji sebanyak 10 iterasi. Parameter yang diukur meliputi latensi (dalam milidetik), *throughput* (dalam megabit per detik), dan *packet loss* (dalam persen) sebagai indikator kualitas layanan (*Quality of Service/QoS*).

Pada skenario *baseline*, rata-rata latensi tercatat sebesar 35,2 ms, *throughput* sebesar 420 Mbps, dan *packet loss* sebesar 2,8%. Sementara pada skenario optimisasi dengan klusterisasi Kubernetes, terjadi peningkatan: latensi turun menjadi 25,1 ms, *throughput*

naik menjadi 565 Mbps, dan *packet loss* berkurang menjadi 1,9%.

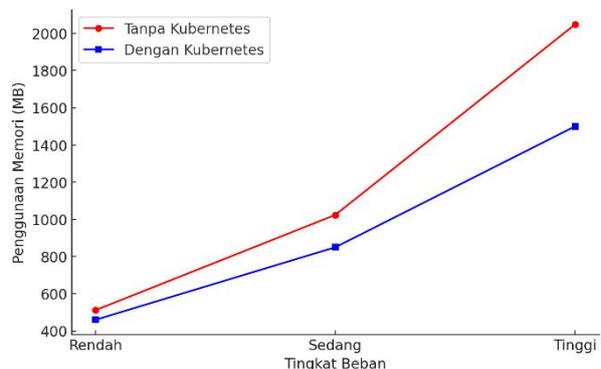
Peningkatan tersebut menunjukkan bahwa klusterisasi Kubernetes berperan dalam pengalokasian sumber daya, penyeimbangan beban trafik jaringan, dan peningkatan efisiensi transmisi data. Penurunan latensi mendukung respons sistem yang lebih cepat, *throughput* memungkinkan aplikasi *bandwidth* besar berjalan lebih baik, sementara rendahnya *packet loss* mendukung koneksi yang lebih stabil untuk komunikasi *real-time*.



Gambar 4. Perbandingan Penggunaan CPU

Gambar 4 memperlihatkan adanya penurunan signifikan pada beban pemrosesan sistem setelah implementasi klusterisasi menggunakan Kubernetes. Dalam kondisi trafik tinggi, sistem *baseline* tanpa klusterisasi menunjukkan penggunaan CPU yang mencapai 85%, mengindikasikan beban yang tinggi pada unit pemrosesan. Sebaliknya, sistem yang diatur dengan Kubernetes hanya mencatatkan penggunaan CPU sebesar 65% pada kondisi yang sama. Penurunan sebesar 20% ini menunjukkan efisiensi yang dihasilkan dari mekanisme alokasi sumber daya yang lebih merata dan adaptif antar *node* dalam kluster.

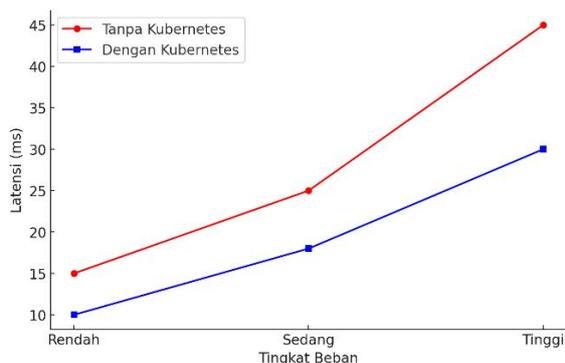
Kubernetes secara otomatis mendistribusikan beban kerja berdasarkan kapasitas dan kondisi masing-masing *node*, sehingga mencegah terjadinya *overloading* pada satu titik tertentu. Dengan demikian, sistem mampu mempertahankan performa yang stabil dan responsif, bahkan saat mengalami lonjakan trafik, serta mengurangi potensi terjadinya *bottleneck* yang dapat mengganggu kualitas layanan secara keseluruhan.



Gambar 5. Perbandingan Penggunaan Memori

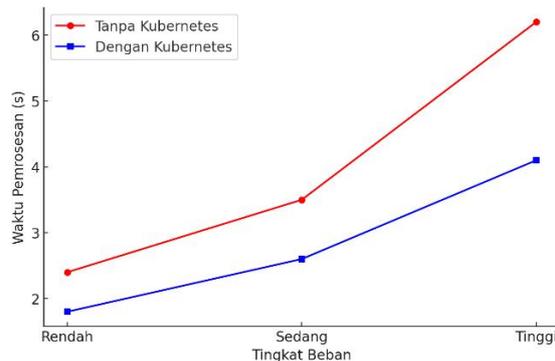
Gambar 5 mengilustrasikan bahwa penggunaan memori menjadi salah satu parameter penting dalam menilai stabilitas dan efisiensi suatu sistem. Pada sistem yang tidak menerapkan klusterisasi, konsumsi memori saat mengalami lonjakan trafik tercatat mencapai 2048 MB. Sebaliknya, pada sistem yang telah dioptimalkan dengan penerapan Kubernetes, penggunaan memori hanya sebesar 1500 MB dalam kondisi yang sama.

Pengurangan sebesar 25% ini tidak hanya mencerminkan efisiensi penggunaan sumber daya, tetapi juga menegaskan kemampuan Kubernetes dalam mengelola alokasi memori secara dinamis. Dengan fitur orkestrasi yang dimilikinya, Kubernetes mampu mengarahkan beban kerja ke *node* yang aktif, sekaligus meminimalisir pemborosan memori pada *node* yang sedang tidak digunakan. Efisiensi ini menjadi indikator kuat bahwa Kubernetes berkontribusi langsung terhadap peningkatan performa dan stabilitas sistem, terutama dalam skenario dengan trafik tinggi dan kebutuhan skalabilitas yang tinggi.



Gambar 6. Perbandingan Latensi Jaringan

Latensi jaringan merupakan salah satu parameter krusial, terutama dalam aplikasi *real-time* seperti *telemedicine*, kendaraan otonom, dan sistem industri otomatisasi yang sangat bergantung pada kecepatan respons data. Gambar 6 memperlihatkan perbandingan latensi antara sistem tanpa klusterisasi dan sistem yang telah dioptimalkan dengan Kubernetes. Dalam kondisi trafik tinggi, sistem tanpa klusterisasi mencatatkan latensi sebesar 45 milidetik, sementara sistem dengan Kubernetes hanya menunjukkan latensi sebesar 30 milidetik. Penurunan sebesar 33% ini menunjukkan peningkatan performa yang signifikan. Efisiensi tersebut terutama dihasilkan dari mekanisme *routing* trafik yang lebih terstruktur serta pengelolaan beban kerja oleh Kubernetes.



Gambar 7. Perbandingan Waktu Pemrosesan

Gambar 7 menampilkan metrik waktu pemrosesan, yang didefinisikan sebagai durasi yang dibutuhkan sistem untuk memproses dan merutekan paket data dari sumber ke tujuan. Parameter ini sangat penting dalam menilai efisiensi operasional, khususnya saat sistem berada dalam kondisi trafik tinggi. Pada skenario baseline tanpa penerapan klusterisasi, waktu pemrosesan tercatat mencapai 6,2 detik. Sebaliknya, ketika sistem menggunakan klusterisasi dengan Kubernetes, waktu pemrosesan berkurang secara signifikan menjadi 4,1 detik. Pengurangan sebesar 34% ini mencerminkan peningkatan efisiensi yang nyata dalam manajemen trafik dan pemrosesan data. Efisiensi ini diperoleh dari kemampuan Kubernetes dalam mendistribusikan beban kerja secara merata ke seluruh *node* aktif, sehingga menghindari terjadinya penumpukan proses pada satu titik tertentu. Dengan demikian, sistem dapat beroperasi lebih responsif dan stabil, bahkan dalam kondisi trafik yang tinggi sekalipun.

4. Kesimpulan

Penelitian ini mengevaluasi dampak klusterisasi Kubernetes terhadap efisiensi komputasi dan performa infrastruktur jaringan 5G. Hasil yang diperoleh menunjukkan bahwa Kubernetes dapat menurunkan penggunaan CPU hingga 20% melalui distribusi beban kerja, menghemat konsumsi memori hingga 25% dengan manajemen sumber daya, mengurangi latensi jaringan sebesar 33% sehingga lebih responsif untuk aplikasi *real-time*, dan mempercepat waktu pemrosesan hingga 34% berkat *load balancing* dan *auto-scaling*. Integrasi Kubernetes dalam arsitektur jaringan 5G menciptakan sistem yang fleksibel dan adaptif. Kemampuan Kubernetes dalam mengatasi masalah seperti alokasi sumber daya, distribusi trafik, dan manajemen kemacetan membuktikan bahwa metode ini sangat cocok untuk jaringan generasi berikutnya. Selain itu, klusterisasi Kubernetes juga memberikan kontribusi terhadap efisiensi energi dan pengurangan biaya operasional melalui *auto-scaling*. Hal ini menjadikannya solusi untuk implementasi *cloud-native* 5G, khususnya pada lingkungan yang menuntut skalabilitas tinggi, komunikasi latensi rendah, dan mMTC. Implikasi dari penelitian ini relevan bagi

operator jaringan, penyedia layanan *cloud*, serta perusahaan yang mengelola fungsi jaringan berskala besar. Pengembangan ke depan mencakup integrasi berbasis AI, penerapan multi-klaster lintas wilayah, serta kolaborasi dengan teknologi SDN dan NFV untuk mempercepat transisi menuju jaringan 5G dan 6G.

Daftar Rujukan

- [1] S. Wijethilaka and M. Liyanage, "Survey on Network Slicing for Internet of Things Realization in 5G Networks," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 957–994, 2021, doi: 10.1109/COMST.2021.3067807.
- [2] W. Rafique, J. Rani Barai, A. O. Fapojuwo, and D. Krishnamurthy, "A Survey on Beyond 5G Network Slicing for Smart Cities Applications," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 1, pp. 595–628, Feb. 2025, doi: 10.1109/COMST.2024.3410295.
- [3] "Ethernet VPN Provisioning Towards Slicing for 5G Transport Networks | IEEE Conference Publication | IEEE Xplore." Accessed: Mar. 18, 2025. [Online]. Available: <https://ieeexplore.ieee.org/document/10258168>
- [4] H. Li et al., "Slice-Based Service Function Chain Embedding for End-to-End Network Slice Deployment," *IEEE Transactions on Network and Service Management*, vol. 20, no. 3, pp. 3652–3672, Sep. 2023, doi: 10.1109/TNSM.2023.3250719.
- [5] M. Bagaa, T. Taleb, A. Laghrissi, A. Ksentini, and H. Flinck, "Coalitional Game for the Creation of Efficient Virtual Core Network Slices in 5G Mobile Systems," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 469–484, Mar. 2018, doi: 10.1109/JSAC.2018.2815398.
- [6] C. Stan, A. Domeke, C. R. Garcia, I. T. Monroy, and J. J. V. Olmos, "Improving the flexibility of 5G heterogeneous networks by slicing," in 2021 IEEE International Mediterranean Conference on Communications and Networking (MeditCom), IEEE, Sep. 2021, pp. 65–69. doi: 10.1109/MeditCom49071.2021.9647586.
- [7] A. Checko et al., "Cloud RAN for Mobile Networks—A Technology Overview," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 405–426, 2015, doi: 10.1109/COMST.2014.2355255.
- [8] A. Deshpande, O. Obi, E. Stipidis, and P. Charchalakis, "Integrated vtronic survivability: requirements for vtronic survivability strategies," in 6th IET International Conference on System Safety 2011, IET, 2011, pp. A42–A42. doi: 10.1049/cp.2011.0249.
- [9] A. Murmu, P. Kumar, N. Rao Moparthy, S. Namasudra, and P. Lorenz, "Reliable Federated Learning With GAN Model for Robust and Resilient Future Healthcare System," *IEEE Transactions on Network and Service Management*, vol. 21, no. 5, pp. 5335–5346, Oct. 2024, doi: 10.1109/TNSM.2024.3422376.
- [10] M. Gramaglia et al., "Flexible connectivity and QoS/QoE management for 5G Networks: The 5G NORMA view," in 2016 IEEE International Conference on Communications Workshops (ICC), IEEE, May 2016, pp. 373–379. doi: 10.1109/ICCW.2016.7503816.
- [11] C. Yan and S. Sheng, "Sdn+K8s Routing Optimization Strategy in 5G Cloud Edge Collaboration Scenario," *IEEE Access*, vol. 11, pp. 8397–8406, 2023, doi: 10.1109/ACCESS.2023.3237201.
- [12] J. Santos, M. Verkerken, L. D'Hooge, T. Wauters, B. Volckaert, and F. De Turck, "Performance Impact of Queue Sorting in Container-Based Application Scheduling," in 2023 19th International Conference on Network and Service Management (CNSM), IEEE, Oct. 2023, pp. 1–9. doi: 10.23919/CNSM59352.2023.10327792.
- [13] H. Hawilo, M. Jammal, and A. Shami, "Orchestrating network function virtualization platform: Migration or re-instantiation?," in 2017 IEEE 6th International Conference on Cloud Networking (CloudNet), IEEE, Sep. 2017, pp. 1–6. doi: 10.1109/CloudNet.2017.8071528.
- [14] I. Syrigos, N. Makris, and T. Korakis, "Multi-Cluster Orchestration of 5G Experimental Deployments in Kubernetes over High-Speed Fabric," in 2023 IEEE Globecom Workshops (GC Wkshps), IEEE, Dec. 2023, pp. 1764–1769. doi: 10.1109/GCWkshps58843.2023.10465054.
- [15] S. Zhao, "An Evolutionary Game Model of Digital Product Innovation Enabled by 5G," *Proceedings - 2023 International Seminar on Computer Science and Engineering Technology, SCSET 2023*, pp. 515–519, 2023, doi: 10.1109/SCSET58950.2023.00119.
- [16] J. Lee and D. Kim, "A Study on Innovation in University Education: Focusing on 5G Mobile Communication," 2020 IEEE 17th Annual Consumer Communications and Networking Conference, CCNC 2020, Jan. 2020, doi: 10.1109/CCNC46108.2020.9045138.
- [17] G. A. Santos, J. P. J. Da Costa, and A. A. S. Da Silva, "Towards to Beyond 5G Virtual Environment for Cybersecurity Testing in V2X Systems," 2023 Workshop on Communication Networks and Power Systems, WCNPS 2023, 2023, doi: 10.1109/WCNPS60622.2023.10344440.
- [18] T. M. Ho, K.-K. Nguyen, J. D. Vo, A. Larabi, and M. Cheriet, "Energy Efficient Orchestration for O-RAN," *GLOBECOM 2024 - 2024 IEEE Global Communications Conference*, pp. 3316–3321, Dec. 2024, doi: 10.1109/GLOBECOM52923.2024.10901461.
- [19] A. Dutta and E. Hammad, "5G Security Challenges and Opportunities: A System Approach," in 2020 IEEE 3rd 5G World Forum (5GWF), IEEE, Sep. 2020, pp. 109–114. doi: 10.1109/5GWF49715.2020.9221122.
- [20] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-Native Network Slicing Using Software Defined Networking Based Multi-Access Edge Computing: A Survey," *IEEE Access*, vol. 9, no. 3, pp. 10903–10924, 2021, doi: 10.1109/ACCESS.2021.3050155.
- [21] O. Chabbouh, S. Ben Rejeb, N. Agoulmine, and Z. Choukair, "Cloud RAN Architecture Model Based upon Flexible RAN Functionalities Split for 5G Networks," in 2017 31st International Conference on Advanced Information Networking and Applications Workshops (WAINA), IEEE, Mar. 2017, pp. 184–188. doi: 10.1109/WAINA.2017.107.
- [22] Z. Zhang, R. Mo, X. Hou, X. Wei, and S. Guo, "Resource Allocation of Large-Scale Communication Network based on Machine Learning," in 2024 International Conference on Integrated Circuits and Communication Systems (ICICACS), IEEE, Feb. 2024, pp. 1–5. doi: 10.1109/ICICACS60521.2024.10498186.
- [23] S. Rajagopal, J. Baker, and R. Akula, "Testing Open RAN in the Cloud (TORC)," *IEEE Vehicular Technology Conference, 2024*, doi: 10.1109/VTC2024-FALL63153.2024.10757934.
- [24] A. Dutta and E. Hammad, "5G Security Challenges and Opportunities: A System Approach," in 2020 IEEE 3rd 5G World Forum (5GWF), IEEE, Sep. 2020, pp. 109–114. doi: 10.1109/5GWF49715.2020.9221122.
- [25] R. M. Cherifi, M. A. Boudjelli, S. Cherrared, and F. Guillemin, "5G Services Placement in Dynamic Multi-clusters," in 2024 15th International Conference on Network of the Future (NoF), IEEE, Oct. 2024, pp. 204–208. doi: 10.1109/NoF62948.2024.10741415.
- [26] N. P. K. R. Chaganti, S. L. Reddy, E.licherla, S. Guddati, A. Swarna, and P. T. Reddy, "Optimizing Latency and Communication in Federated Edge Computing with LAFeO and Gradient Compression for Real-time Edge Analytics," 2025 6th International Conference on Mobile Computing and Sustainable Informatics (ICMCSI), pp. 608–613, Jan. 2025, doi: 10.1109/ICMCSI64620.2025.10883220.
- [27] Y. Fan, Y. Li, M. Zhan, H. Cui, and Y. Zhang, "IoTDefender: A Federated Transfer Learning Intrusion Detection Framework for 5G IoT," in 2020 IEEE 14th International Conference on Big Data Science and Engineering (BigDataSE), IEEE, Dec. 2020, pp. 88–95. doi: 10.1109/BigDataSE50710.2020.00020.
- [28] L. Wojciechowski et al., "NetMARKS: Network metrics-AwaRe kubernetes scheduler powered by service mesh," *Proceedings - IEEE INFOCOM*, vol. 2021-May, May 2021, doi: 10.1109/INFOCOM42981.2021.9488670.
- [29] P. Rost et al., "Network Slicing to Enable Scalability and Flexibility in 5G Mobile Networks," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 72–79, May 2017, doi: 10.1109/MCOM.2017.1600920.

- [30]M. A. Habibi, M. Nasimi, B. Han, and H. D. Schotten, "A Comprehensive Survey of RAN Architectures Toward 5G Mobile Communication System," *IEEE Access*, vol. 7, no. 2, pp. 70371–70421, 2019, doi: 10.1109/ACCESS.2019.2919657.
- [31]Y. Liu and A. H. Herranz, "Enabling 5G QoS configuration capabilities for IoT applications on container orchestration platform," in 2023 IEEE International Conference on Cloud Computing Technology and Science (CloudCom), IEEE, Dec. 2023, pp. 63–68. doi: 10.1109/CloudCom59040.2023.00023.
- [32]Y. Huo, X. Dong, and W. Xu, "5G Cellular User Equipment: From Theory to Practical Hardware Design," *IEEE Access*, vol. 5, pp. 13992–14010, 2017, doi: 10.1109/ACCESS.2017.2727550.