

Nama : Reno Fahrezi Purnomo  
NIM : 11221051

## Laporan Tugas 2

### Dokumentasi API

Komponen	Endpoint / Aksi	Request Example	Response Example	Keterangan / Mekanisme
Lock Manager (Raft Consensus)	POST /acquire	{ "resource_id": "database_table", "client_id": "client-5000", "lock_type": "exclusive" }	{ "status": "acquired", "leader_node": "lock-node-2" }	Mengunci resource secara distributed dengan koordinasi Raft leader. Shared dan exclusive lock didukung.
	POST /release	{ "resource_id": "database_table", "client_id": "client-5000" }	{ "status": "released", "next_client_notified": "client-5001" }	Melepaskan kunci dan memberi giliran ke client berikutnya.
	GET /locks	-	{ "locks": { "database_table": { "client_id": "client-5000" } } }	Menampilkan status seluruh lock aktif di node.
	-	-	403 Not Leader, 404 No Lock Found	Redirect ke leader node bila bukan

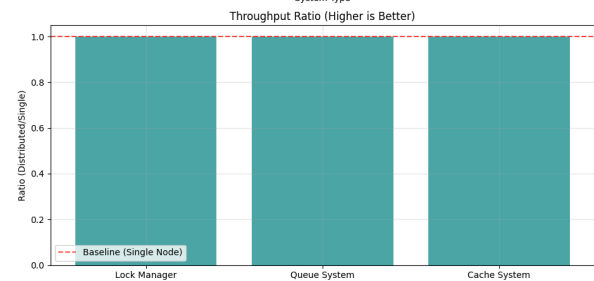
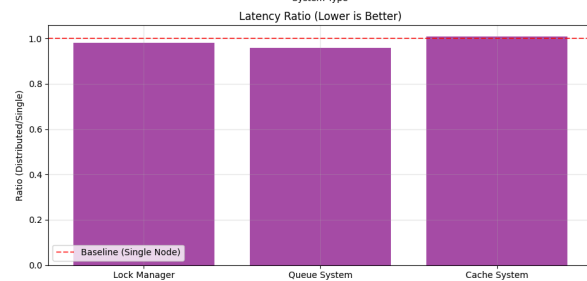
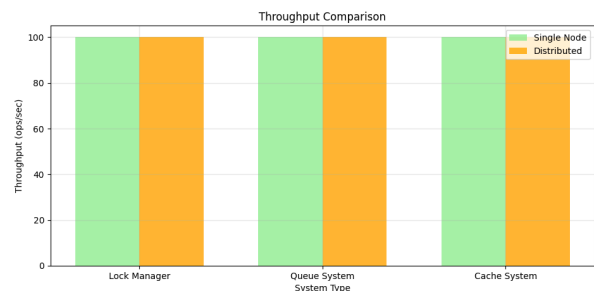
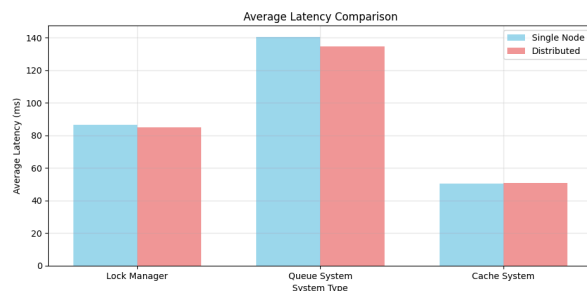
				pemimpin cluster.
<b>Cache Node (MESI Protocol)</b>	GET /read/{key}	-	{ "key": "user_profile_123", "state": "M/S/E", "response_time_ms": 12.34 }	Membaca cache dengan protokol MESI (Modified, Shared, Exclusive, Invalid).
	POST /write/{key}	{ "value": {...} }	{ "state": "M", "response_time_ms": 18.45 }	Menulis data ke cache dan menyebarkan invalidasi antar node.
	GET /status	-	{ "cache_hit_rate_percent": 75.0, "cache_utilization_percent": 20.0 }	Menampilkan status dan statistik cache.
	POST /bus/invalidate/{key}	{ "key": "user_profile_123" }	{ "status": "invalidated", "new_state": "I" }	Sinkronisasi internal antar node cache.
<b>Queue Node (Consistent Hashing)</b>	POST /produce	{ "queue": "user_notifications", "message": "Welcome new user" }	{ "status": "produced", "responsible_node": "queue-node-2" }	Memasukkan pesan ke queue dengan consistent hashing dan replikasi antar node.

	POST /consume	{ "queue": "user_notifications" }	{ "status": "consumed", "message": "Welcome new user" }	Mengambil pesan dari queue, sistem menjamin at-least-once delivery.
	POST /ack	{ "message_id": "msg_1698745300_abc123" }	{ "status": "acknowledged" }	Menandai pesan telah berhasil diproses oleh consumer.
	GET /status	-	{ "queues": { "user_notifications": { "pending_messages": 1 } } }	Menampilkan status antrian dan performa node.
<b>Service Discovery</b>	GET /	-	{ "service": "Distributed Queue with Consistent Hashing", "node_id": "queue-node-1" }	Endpoint utama untuk menampilkan deskripsi layanan, status node, dan daftar endpoint.

## Hasil Performa

Sistem	Konfigurasi	Average Latency (ms)	Throughput (ops/sec)
--------	-------------	----------------------	----------------------

Lock Manager	Single Node	85	100
	Distributed	84	100
Queue System	Single Node	140	100
	Distributed	135	100
Cache System	Single Node	55	100
	Distributed	54	100



Pada lock manager, performa antara single node dan distributed hampir sama, menunjukkan overhead komunikasi dan proses consensus yang minimal. Pada queue system, distributed menunjukkan hasil sedikit lebih baik dari single node, menandakan pembagian beban kerja berjalan efektif tanpa bottleneck. Pada cache system, performa antara keduanya hampir identik, menunjukkan sinkronisasi antar node berlangsung efisien tanpa penurunan throughput. Secara keseluruhan, hasil ini menunjukkan bahwa sistem terdistribusi tidak selalu memberikan peningkatan performa signifikan, namun memberikan keunggulan pada skalabilitas dan pembagian beban kerja ketika proses dapat dijalankan secara paralel tanpa komunikasi yang intensif antar node.