Program Studi Teknik Informatika
Institut Teknologi Sumatera

Kelompok: **Kelompok 3** Tugas: **Tugas Besar**
Mata Kuliah: **Komputasi Awan** Tanggal: 20 Desember 2025

Arsitektur Cloud Sistem Manajemen Surat (Simas) pada Google Cloud Platform

Anggota Kelompok:

1. Luthfiandri Ardanie - 122140089
2. Muhammad Fatih Hanbali - 122140112
3. Falih Dzakwan Zuhdi - 122140132
4. Hamka Putra Andiyan - 122140121
5. Bayu Ega Ferdana - 122140129
6. Anjes Bermana - 122140190

1 Pendahuluan

Sistem Manajemen Surat (Simas) merupakan aplikasi berbasis web untuk pengelolaan surat masuk dan keluar yang di-deploy menggunakan arsitektur *cloud-native* pada Google Cloud Platform (GCP). Sistem ini dirancang dengan pendekatan *serverless* untuk mencapai skalabilitas tinggi, keamanan berlapis, dan efisiensi biaya operasional.

1.1 Cloud Provider dan Deployment Region

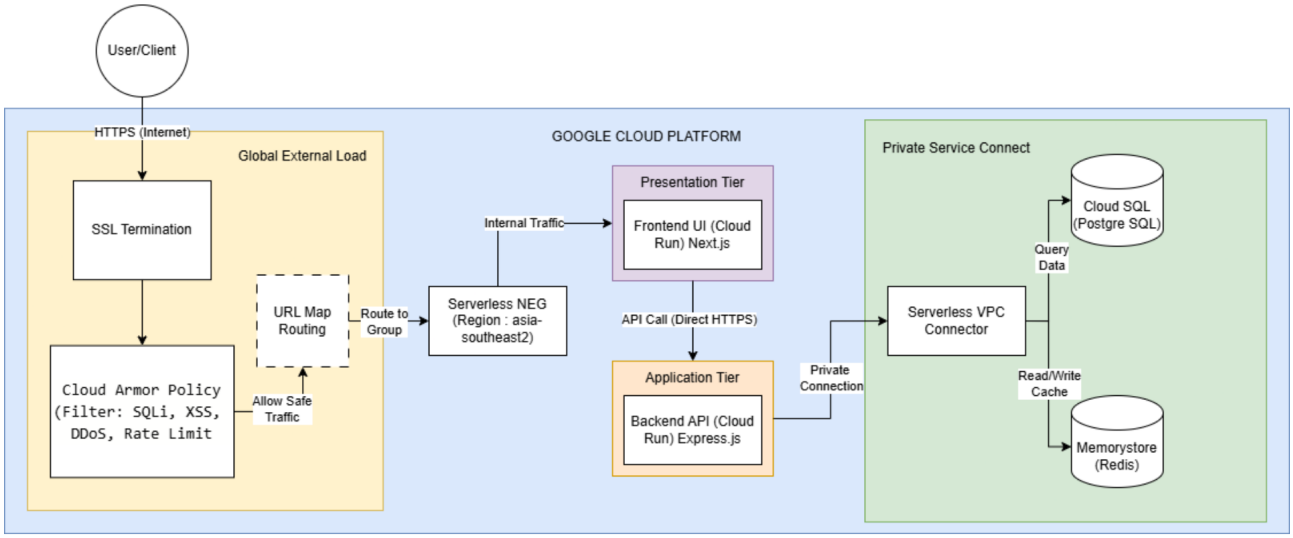
Simas menggunakan **Google Cloud Platform (GCP)** sebagai cloud provider utama dengan deployment di dua region menggunakan Cloud Run [1]:

- **asia-southeast2** (Jakarta, Indonesia) - Backend services, database, dan cache
- **asia-southeast1** (Singapore) - Frontend services untuk latensi rendah ke pengguna

Pemilihan region ini didasarkan pada pertimbangan latensi rendah untuk pengguna Indonesia dan kepatuhan terhadap regulasi data sovereignty.

1.2 Arsitektur Sistem

Gambar 1 menunjukkan arsitektur lengkap sistem Simas yang terdiri dari komponen frontend (Next.js), backend (Express.js), database (Cloud SQL PostgreSQL), cache layer (Redis), load balancer dengan SSL termination, dan security layer (Cloud Armor WAF).



Gambar 1: Arsitektur sistem Simas di Google Cloud Platform

2 Infrastruktur Utama

Infrastruktur Simas dibangun menggunakan managed services dari GCP untuk meminimalkan operational overhead dan memaksimalkan reliability.

2.1 Cloud Run Services

Cloud Run dipilih sebagai compute platform karena sifatnya yang serverless, auto-scaling, dan pay-per-use. Sistem memiliki dua Cloud Run services:

2.1.1 Frontend Service

- **Teknologi:** Next.js 15 dengan React 19 (Server-Side Rendering)
- **Container:** Node.js 18 Alpine dengan standalone build
- **Resources:** 1 vCPU, 2GB RAM
- **Scaling:** 0-20 instances (scale to zero untuk cost optimization)
- **Concurrency:** 80 requests per instance

2.1.2 Backend Service

Backend service dikonfigurasi dengan minimum 1 instance untuk menghindari cold start dan memastikan response time yang konsisten. Kode 1 menunjukkan konfigurasi Terraform [2] untuk backend service.

```
resource "google_cloud_run_service" "backend" {
  name     = "simasbe"
  location = var.region_be
```

```
4
5   template {
6     metadata {
7       annotations = {
8         "autoscaling.knative.dev/maxScale" = "20"
9         "run.googleapis.com/vpc-access-connector" =
10            var.vpc_connector_id
11        "run.googleapis.com/vpc-access-egress" =
12            "private-ranges-only"
13      }
14    }
15
16    spec {
17      service_account_name  = "compute@developer.gserviceaccount.com"
18      container_concurrency = 80
19      timeout_seconds       = 300
20
21      containers {
22        image = "asia-southeast2-docker.pkg.dev/..."
23
24        resources {
25          limits = {
26            cpu    = "1000m"
27            memory = "512Mi"
28          }
29        }
30      }
31    }
32  }
33 }
```

Kode 1: Konfigurasi Cloud Run Backend Service

Spesifikasi lengkap Cloud Run services dapat dilihat pada Tabel 1.

Tabel 1: Spesifikasi Cloud Run Services

Spesifikasi	Frontend	Backend
CPU	1000m (1 vCPU)	1000m (1 vCPU)
Memory	2Gi	512Mi
Min Instances	0	1
Max Instances	20	20
Concurrency	80	80
Timeout	300s	300s
Region	asia-southeast1	asia-southeast2

2.2 Database dan Cache Layer

2.2.1 Cloud SQL PostgreSQL

Database menggunakan Cloud SQL PostgreSQL [3] dengan konfigurasi:

- **Version:** PostgreSQL 14
- **Tier:** db-f1-micro (0.6GB RAM, shared vCPU)
- **Storage:** 10GB SSD dengan auto-increase (max 100GB)
- **Network:** Private IP only (tidak ada public access)

- **Backup:** Automated daily backup dengan 7-day retention

2.2.2 Memorystore for Redis

Redis [4] digunakan untuk session storage, caching, dan rate limiting dengan spesifikasi:

- **Version:** Redis 7.0
- **Tier:** Basic (single zone)
- **Capacity:** 1GB memory
- **Network:** Private IP dalam VPC
- **Eviction Policy:** allkeys-lru

3 Networking dan Load Balancing

3.1 HTTPS Load Balancer

Sistem menggunakan Global HTTPS Load Balancer dengan komponen:

- **Frontend NEG:** Serverless Network Endpoint Group ke Cloud Run
- **Backend Service:** Routing traffic ke frontend NEG
- **SSL Termination:** HTTPS proxy dengan managed SSL certificate
- **Static IP:** Global anycast IP untuk low latency
- **HTTP Redirect:** Automatic redirect dari HTTP ke HTTPS

3.2 SSL Certificate

SSL certificate menggunakan Google-managed certificate untuk domain `komasimas.web.id` dengan auto-renewal dan TLS 1.2+ minimum version.

3.3 VPC dan Private Networking

Backend service terhubung ke Cloud SQL dan Redis melalui VPC Connector dengan konfigurasi:

- **VPC Connector:** Private access ke internal services
- **Egress Setting:** `private-ranges-only` untuk security
- **Subnet Ranges:**
  - Connector: 10.12.0.0/28
  - Cloud SQL: 10.10.0.0/24
  - Redis: 10.11.0.0/24

4 Keamanan Multi-Layer

4.1 Cloud Armor Web Application Firewall

Cloud Armor WAF [5] melindungi aplikasi dari serangan web dengan tiga layer protection yang ditunjukkan pada Kode 2.

```
resource "google_compute_security_policy" "simas_waf" {
  name = "simas-security-policy-v2"
  type = "CLOUD_ARMOR"

  # Rule 1: Block SQL Injection
  rule {
    action   = "deny(403)"
    priority = "1000"
    match {
      expr {
        expression = "evaluatePreconfiguredExpr('sqli-v33-stable')"
      }
    }
    description = "Block SQL Injection attacks"
  }

  # Rule 2: Block XSS
  rule {
    action   = "deny(403)"
    priority = "1100"
    match {
      expr {
        expression = "evaluatePreconfiguredExpr('xss-v33-stable')"
      }
    }
    description = "Block XSS attacks"
  }

  # Rule 3: Rate Limiting
  rule {
    action   = "rate_based_ban"
    priority = "2000"
    rate_limit_options {
      conform_action = "allow"
      exceed_action  = "deny(429)"
      enforce_on_key = "IP"
      rate_limit_threshold {
        count        = 200
        interval_sec = 60
      }
      ban_duration_sec = 300
    }
    description = "Rate Limit: 200 req/min"
  }
}
```

Kode 2: Konfigurasi Cloud Armor WAF Rules

Tabel 2 merangkum security rules yang diimplementasikan.

4.2 Network Security

- **Private IP Only:** Cloud SQL dan Redis tidak memiliki public access
- **VPC Isolation:** Database hanya accessible melalui VPC Connector

Tabel 2: Cloud Armor WAF Security Rules

Priority	Rule	Action	Threshold
1000	SQL Injection	Deny 403	-
1100	XSS Attack	Deny 403	-
2000	Rate Limiting	Ban 300s	200 req/min
Default	Allow All	Allow	-

- **SSL/TLS:** Semua komunikasi dienkripsi (TLS 1.3)
- **IAM:** Least privilege access dengan service accounts

4.3 Application Security

- **Authentication:** Token-based authentication dengan UUID
- **Password Hashing:** bcrypt dengan salt rounds
- **Authorization:** Role-based access control (admin/user)
- **Input Validation:** Zod schema validation
- **SQL Injection Prevention:** Prisma ORM dengan parameterized queries

5 Deployment dan Containerization

5.1 Docker Multi-Stage Build

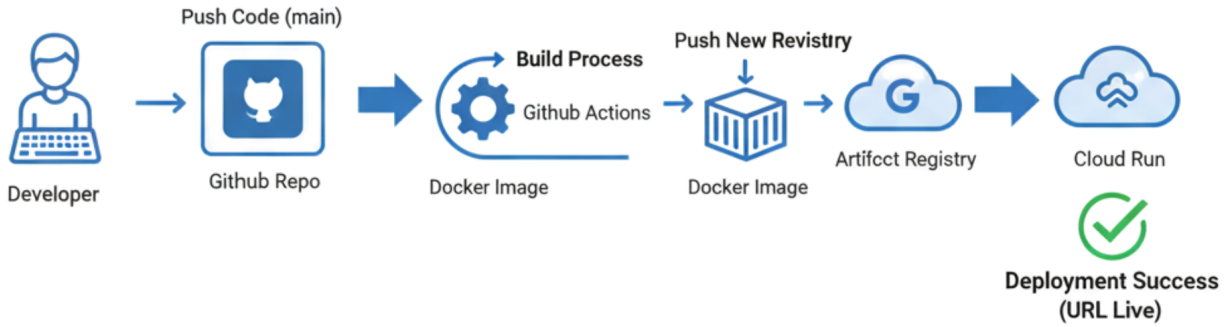
Aplikasi menggunakan Docker multi-stage build [6] untuk optimasi ukuran image dan security. Kode 3 menunjukkan Dockerfile backend.

```
FROM node:18-alpine

WORKDIR /app

# Copy package files and install dependencies
COPY package*.json ./
RUN npm ci

# Copy source code
COPY . .

# Build the TypeScript code
RUN npm run build

# Expose port 8080 (Cloud Run standard)
ENV PORT=8080
EXPOSE 8080

# Start the application
CMD ["node", "dist/main.js"]
```

Kode 3: Backend Dockerfile Multi-Stage Build

5.2 CI/CD Pipeline

Gambar 2 menunjukkan pipeline deployment yang mencakup build, test, push ke Artifact Registry, dan deploy ke Cloud Run. Pipeline deployment menggunakan strategi rolling update dengan health checks dan automatic rollback pada kegagalan.



Gambar 2: Pipeline CI/CD deployment sistem Simas

6 Skalabilitas dan Estimasi Cost

6.1 Auto-Scaling Configuration

Cloud Run secara otomatis melakukan scaling berdasarkan:

- Concurrent requests per instance (max 80)
- CPU utilization
- Request queue depth

Frontend dapat scale dari 0 hingga 20 instances (scale to zero untuk cost savings), sedangkan backend maintain minimum 1 instance untuk menghindari cold start dengan maksimum 20 instances untuk high load scenarios.

6.2 Estimasi Biaya Bulanan

Tabel 3 menunjukkan estimasi biaya operasional bulanan sistem dengan traffic medium (100,000 requests per bulan).
**Konversi dengan rate Rp 15,600 per USD*

Analisis Biaya:

Tabel 3: Estimasi Biaya Operasional Bulanan (Medium Traffic)

Komponen	**USD/Bulan**	**IDR/Bulan***
Cloud Run Frontend	$1.60	Rp 24,960
Cloud Run Backend	$15.00	Rp 234,000
Cloud SQL (db-f1-micro)	$7.00	Rp 109,200
Memorystore Redis (1GB)	$15.00	Rp 234,000
Load Balancer	$5.00	Rp 78,000
Compute Engine (VPC)	$3.00	Rp 46,800
Networking	$2.00	Rp 31,200
Artifact Registry	$0.50	Rp 7,800
Cloud Build	$0.30	Rp 4,680
Cloud Storage	$0.30	Rp 4,680
Cloud Logging	$0.30	Rp 4,680
SSL Certificate	$0.00	Rp 0
Total	**$50.00**	**Rp 780,000**

- Cloud Run Backend (30%) dan Redis (30%) merupakan komponen tertinggi untuk compute dan caching
- Cloud SQL (14%) untuk managed database dengan automated backups
- Load Balancer dan networking infrastructure 14% dari total cost
- Cloud Run Frontend sangat murah ($1.60) dengan scale-to-zero capability
- Total estimasi Rp 780 ribu/bulan ($50) sangat cost-efficient untuk production workload
- Pay-per-use model memastikan biaya tetap rendah untuk traffic medium

6.3 High Availability

- **Multi-zone deployment:** Cloud Run distributed across zones
- **Automated backups:** Daily backup dengan 7-day retention
- **Point-in-Time Recovery:** Database recovery sampai detik tertentu
- **Health checks:** Automatic instance replacement jika unhealthy

7 Kesimpulan

Sistem Manajemen Surat (Simas) mengimplementasikan arsitektur cloud-native yang modern dengan karakteristik:

1. **Serverless Architecture:** Menggunakan Cloud Run untuk compute dengan auto-scaling 0-20 instances, mengurangi operational overhead dan optimasi biaya.
2. **Security Multi-Layer:**
   - Cloud Armor WAF melindungi dari SQL injection, XSS, dan rate limiting (200 req/min)
   - Private networking untuk database dan cache (no public access)
   - SSL/TLS encryption untuk semua komunikasi

- Role-based access control di application layer

3. **High Availability:** Multi-zone deployment dengan automated backups, point-in-time recovery, dan health checks untuk reliability 99.95%.

4. **Cost Efficiency:** Estimasi biaya Rp 780 ribu/bulan ($50) untuk medium traffic dengan pay-per-use model. Cloud Run Backend dan Redis masing-masing 30% dari total cost. Scale-to-zero frontend dan optimized resource allocation menghemat biaya signifikan.

5. **Infrastructure as Code:** Seluruh infrastruktur dikelola menggunakan Terraform untuk reproducibility dan version control.

Arsitektur ini membuktikan bahwa aplikasi web modern dapat di-deploy dengan security, scalability, dan cost-efficiency yang optimal menggunakan managed services dari Google Cloud Platform. Dengan biaya Rp 780 ribu/bulan, sistem production-ready dengan kemampuan scale hingga 20 instances per service.

References

[1] Google Cloud, "Cloud run documentation - serverless container platform," https://cloud.google.com/run/docs, 2024, accessed: 2025-12-20.

[2] HashiCorp, "Terraform google cloud platform provider," https://registry.terraform.io/providers/hashicorp/google/latest/docs, 2024, accessed: 2025-12-20.

[3] Google Cloud, "Cloud SQL for PostgreSQL," https://cloud.google.com/sql/docs/postgres, 2024, accessed: 2025-12-20.

[4] ——, "Memorystore for redis," https://cloud.google.com/memorystore/docs/redis, 2024, accessed: 2025-12-20.

[5] ——, "Cloud armor - DDoS protection and WAF," https://cloud.google.com/armor/docs, 2024, accessed: 2025-12-20.

[6] Docker Inc., "Best practices for writing dockerfiles," https://docs.docker.com/develop/dev-best-practices/, 2024, accessed: 2025-12-20.