Rencana Penelitian Grafika Komputer

Topik: Temporal Code City untuk Visualisasi Evolusi Microservices Berbasis Git History

Anggota:

- Christopher Gijoh (01082240011)
- Evan Laluan (01082240015)
- Christian Oroh (01082240019)

3. Research Methodology

3.1. Research Design

- Jenis Penelitian: Experimental Implementation (Implementasi Eksperimental).
- Pendekatan: Constructive Research — Membangun artefak visualisasi baru untuk memecahkan masalah representasi data kompleks.
- Fokus Utama: Pengembangan teknik rendering real-time untuk data historis berskala besar dan optimasi performa grafis berbasis web.

3.2. Research Object & Scope

- Objek Penelitian:
    - Data riwayat versi (Git Commit History) dari sistem Microservices (Multi-repository).
    - Struktur hierarkis kode (Folder/File) yang dipetakan menjadi geometri 3D prosedural (Distrik/Gedung).
- Ruang Lingkup:
    - Grafis: Implementasi teknik Instanced Rendering untuk menangani ribuan objek gedung tanpa lag.
    - Temporal: Visualisasi perubahan struktural (evolusi) pasca-commit (post-mortem analysis), bukan real-time monitoring server aktif.
    - Platform: Lingkungan berbasis browser (WebGL) untuk aksesibilitas tinggi.

3.3. Research Environment

- Platform & Library: WebGL menggunakan Three.js (High-level API) dan React Three Fiber.
- Bahasa Pemrograman: JavaScript/TypeScript (Front-end Logic) & GLSL (Custom Shaders untuk efek visual khusus).
- Perangkat Keras Uji: Laptop standar mahasiswa (GPU terintegrasi/diskrit kelas menengah) untuk membuktikan efisiensi algoritma.

- Dataset:
    - Dataset Sintetis (Dibuat script untuk simulasi ribuan commit).
    - Dataset Riil (Repo Open Source Microservices, misal: Google Online Boutique).

3.4. Method Development (Graphics Pipeline)

- Data Parsing & Aggregation:
    - Mengembangkan algoritma untuk menyatukan log dari banyak repositori Git menjadi satu linimasa global (Global Timeline).
- Procedural City Generation:
    - Implementasi algoritma tata letak (Treemap atau Force-Directed) yang stabil (posisi gedung tidak melompat acak saat data berubah).
    - Pemetaan atribut metrik kode ke atribut visual (Lines of Code → Tinggi Gedung, Recency → Emisi Warna).
- Rendering Optimization:
    - Penerapan Instanced Mesh untuk merender ribuan gedung dalam satu draw call.
    - Penerapan Delta Updates (hanya memperbarui properti objek yang berubah, bukan merender ulang seluruh scene per frame).

3.5. Data Collection

- Input Data: File JSON hasil ekstraksi git log yang memuat metadata: hash, author, timestamp, file_changed, insertions, deletions.
- Output Metrics (Pengukuran):
    - Performa: Frame Per Second (FPS), Memory Usage (MB), dan Draw Calls Count.
    - Skabilitas: Batas jumlah maksimum commits atau files yang dapat direnderer sebelum FPS turun di bawah 30.

3.6. Data Analysis Technique

- Analisis Kuantitatif (Benchmarking):
    - Stress Test: Mengukur penurunan FPS seiring bertambahnya jumlah gedung (objek 3D) dari 100 hingga 10.000 objek.
    - Membandingkan performa metode Naïve Rendering (tiap gedung satu mesh) vs Instanced Rendering (metode usulan).
- Analisis Kualitatif (Visual Inspection):

    - Validasi visual terhadap Glitches atau Z-fighting saat gedung tumbuh/menyusut.
    - Evaluasi kemudahan identifikasi anomali (apakah God Class atau Hotspot terlihat jelas secara visual?).

3.7. Validation Method

- Proof of Concept (PoC): Mendemonstrasikan skenario kasus penggunaan nyata:
    - Skenario 1: Deteksi Microservice yang membengkak (Scalability visual check).
    - Skenario 2: Pelacakan bug melalui visualisasi area kode yang memerah (sering berubah) dalam waktu singkat.

3.8. Research Output

- Purwarupa Perangkat Lunak: Aplikasi berbasis WebGL yang mampu memvisualisasikan evolusi repositori Git secara interaktif.
- Laporan Analisis Performa: Grafik perbandingan efisiensi rendering teknik yang digunakan.
- Dokumentasi Visual: Video demo evolusi kota dari fase awal proyek hingga fase akhir.