

Proyek Akhir

Mata Kuliah Struktur Data Semester Genap 2025-2026

A. Prinsip Umum Proyek Akhir

- Dikerjakan berkelompok (4-5 mahasiswa)
- Bahasa pemrograman: C++ (diperbolehkan menggunakan STL C++)
- Wajib menggunakan penyimpanan file (read/write)
- Fokus utama:
 - Pemilihan dan implementasi struktur data yang tepat
 - Perbandingan performa (waktu & ruang) antar struktur data untuk data dalam jumlah besar. Data dapat diambil dari public database dengan mencantumkan sumber, atau generate data secara mandiri.
 - Analisis berbasis eksperimen, bukan hanya teori

Setiap proyek harus bisa membuktikan secara empiris mengapa struktur data tertentu lebih sesuai dibanding alternatif lain. Antar muka bersifat opsional, digunakan hanya untuk mempermudah operasi insert, search, update dan delete, bukan menjadi yang utama.

B. Alur Kerja Proyek yang Disarankan

1. Analisis Masalah & Studi Literatur (Minggu 2–3) → dilakukan mandiri di luar waktu kuliah dan praktikum
2. Desain Struktur Data & Algoritma (Minggu 3–4) → dilakukan mandiri di luar waktu kuliah dan praktikum
3. Implementasi Bertahap (Minggu 4–6) → dilakukan mandiri di luar waktu kuliah dan praktikum
4. Eksperimen & Pengujian Awal (Minggu 6) → dilakukan mandiri di luar waktu kuliah dan praktikum
5. Laporan Progress (Minggu 7) → dilakukan saat praktikum minggu ke-7, laporan progress dikumpulkan
6. Refinement & Eksperimen Lanjutan (Minggu 8–12) → dilakukan mandiri di luar waktu kuliah dan praktikum
7. Finalisasi Produk & Laporan (Minggu 13) → dilakukan mandiri di luar waktu kuliah dan praktikum
8. Pengumpulan seluruh artefak proyek akhir (5 hari sebelum Minggu 14)
9. Workshop Proyek Akhir (Minggu 14)

C. Daftar Topik Proyek Akhir

Topik 1. Sistem Manajemen Inventori Gudang

Konteks industri: perusahaan logistik / e-commerce

Kasus ini menggambarkan kondisi operasional nyata di organisasi berskala menengah hingga besar. Sistem yang awalnya dirancang untuk volume data kecil mulai mengalami degradasi performa seiring pertumbuhan pengguna, data, dan kebutuhan operasional. Proses yang semula dapat ditangani secara manual atau dengan pendekatan sederhana menjadi tidak lagi

memadai ketika frekuensi transaksi meningkat, variasi data bertambah, dan tuntutan respons cepat muncul dari pengguna maupun manajemen. Kondisi ini memaksa tim teknis untuk mengevaluasi ulang pendekatan pengelolaan data dan memilih struktur data yang lebih tepat agar sistem tetap stabil, efisien, dan dapat diskalakan.

Permasalahan di lapangan

Pencarian produk dan update stok menjadi lambat seiring pertumbuhan data.

Tujuan proyek

Membangun sistem inventori yang tetap cepat meski data besar dan membuktikan struktur data paling efisien.

Domain data (minimal)

ID produk, nama, kategori, stok, harga, status

Aktivitas sistem wajib

Insert, search, update stok, delete

Fokus analisis

Analisis perbandingan setiap struktur data pada performa aktivitas sistem dan dampak pertumbuhan data

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario penggunaan
- Implementasi minimal **1 struktur data**
- Dataset awal (dummy / simulasi)
- Pengujian awal insert & search

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu & memori
- Demo aplikasi + laporan analisis

Topik 2. Sistem Antrian Layanan Publik Terintegrasi

Konteks industri: rumah sakit

Kasus ini merepresentasikan kondisi operasional nyata di rumah sakit skala menengah hingga besar yang melayani banyak pasien setiap hari. Sistem antrian yang awalnya sederhana (manual, spreadsheet, atau satu loket) mulai tidak mampu menangani peningkatan jumlah pasien, variasi layanan, serta kebutuhan integrasi antar unit (pendaftaran, poli, laboratorium, farmasi). Akibatnya, waktu tunggu menjadi panjang, pencarian data pasien melambat, dan prioritas layanan (misalnya pasien darurat atau lansia) tidak tertangani secara optimal. Kondisi ini mendorong perlunya perancangan sistem antrian berbasis struktur data yang efisien, adil, dan responsif terhadap dinamika layanan rumah sakit.

Permasalahan di lapangan

- Pengelolaan antrian menjadi tidak efisien saat jumlah pasien meningkat.
- Sulit menangani prioritas pasien (darurat, lansia, BPJS, dll.).
- Pencarian dan pemanggilan antrian pasien menjadi lambat jika data menumpuk.

Tujuan proyek

Membangun sistem antrian layanan rumah sakit yang efisien, adil, dan cepat, serta menganalisis struktur data yang paling sesuai untuk menangani antrian dengan volume dan variasi tinggi.

Domain data (minimal)

ID pasien, nama pasien, jenis layanan, tingkat prioritas, nomor antrian, waktu kedatangan, status layanan

Aktivitas sistem wajib

Insert pasien ke antrian,
Pemanggilan antrian berikutnya,
Search data pasien/antrian,
Update status layanan,
Delete antrian (selesai/batal)

Fokus analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan pemanggilan antrian
- Penanganan prioritas pasien
- Efisiensi pencarian data pasien
- Dampak pertumbuhan jumlah pasien terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario layanan rumah sakit
- Implementasi minimal 1 struktur data (misalnya Queue atau Priority Queue)
- Dataset awal pasien (dummy / simulasi)
- Pengujian awal insert & pemanggilan antrian

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi sistem antrian terintegrasi
- Laporan analisis performa & rekomendasi struktur data

Topik 3. Sistem Navigasi & Rekomendasi Rute Distribusi

Konteks industri: logistik, transportasi, supply chain

Kasus ini menggambarkan sistem internal perusahaan logistik yang menyimpan dan mengelola data rute distribusi dari satu atau beberapa gudang ke lokasi tujuan. Pada tahap awal, jumlah lokasi dan rute masih sedikit sehingga dapat dikelola secara sederhana. Namun, seiring bertambahnya jumlah lokasi tujuan dan rute distribusi, proses pencarian, pembaruan, dan penghapusan data rute menjadi semakin lambat dan tidak efisien. Oleh karena itu, diperlukan pemilihan struktur data yang tepat agar sistem tetap stabil dan responsif meskipun volume data meningkat. Pada topik ini, hubungan antar lokasi dimodelkan sebagai data relasi (asal–tujuan). Representasi graf digunakan hanya sebagai cara menyimpan hubungan data, bukan untuk implementasi algoritma pencarian rute terpendek atau optimasi rute.

Permasalahan di lapangan

- Pencarian data rute antar lokasi menjadi lambat saat jumlah rute bertambah.
- Pengelolaan data rute sulit ketika sering terjadi penambahan atau penghapusan lokasi.
- Update jarak atau waktu tempuh rute membutuhkan proses pencarian data yang efisien.
- Struktur penyimpanan sederhana tidak lagi memadai untuk data rute berskala besar.

Tujuan proyek

Membangun sistem manajemen rute distribusi yang efisien dan mudah dikelola, serta membandingkan performa beberapa struktur data dalam menyimpan dan mencari data rute distribusi.

Domain data (minimal)

ID lokasi

Nama lokasi

Tipe lokasi (gudang / tujuan)

ID rute

Lokasi asal

Lokasi tujuan

Jarak atau waktu tempuh rute

Aktivitas sistem wajib

Insert data lokasi dan rute

Search data rute berdasarkan lokasi asal dan tujuan

Update jarak atau waktu tempuh rute

Delete data rute atau lokasi

Catatan: Sistem tidak diwajibkan menentukan rute terpendek atau optimal secara otomatis.

Representasi hubungan antar lokasi dapat disimpan dalam bentuk:

- Adjacency Matrix (array dua dimensi), atau
- Adjacency List (daftar rute per lokasi)

Fokus analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan pencarian data rute
- Kemudahan insert, update, dan delete
- Efisiensi penggunaan memori
- Dampak pertumbuhan jumlah rute terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario distribusi sederhana
- Implementasi minimal 1 struktur data
- Dataset awal lokasi dan rute (dummy / simulasi)
- Pengujian awal insert dan search data rute

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi dan penggunaan memori
- Demo aplikasi manajemen rute distribusi
- Laporan analisis performa dan rekomendasi struktur data

Mahasiswa tidak diwajibkan mengimplementasikan algoritma graf seperti Dijkstra, BFS, DFS, atau pencarian rute terpendek.

Topik 4. Sistem Kompresi Data Sederhana

Konteks industri: penyimpanan & transmisi data

Kasus ini menggambarkan kebutuhan perusahaan dalam mengelola penyimpanan dan transmisi data teks yang volumenya terus meningkat, seperti log aktivitas sistem, data transaksi, atau pesan komunikasi internal. Pada tahap awal, ukuran data masih relatif kecil sehingga penyimpanan dan pengiriman data dapat dilakukan tanpa optimasi khusus. Namun, seiring pertumbuhan data, kebutuhan ruang penyimpanan meningkat dan waktu transmisi menjadi lebih lama. Oleh karena itu, diperlukan sistem kompresi data sederhana yang efisien dan mudah dikelola, dengan fokus pada pemilihan struktur data yang tepat. Pada topik ini, kompresi

data dipahami sebagai proses pengkodean data agar ukurannya lebih kecil. Mahasiswa tidak diwajibkan mengimplementasikan algoritma kompresi kompleks atau standar industri (misalnya ZIP, LZW, atau Huffman penuh). Fokus utama adalah bagaimana struktur data digunakan untuk menyimpan, mengakses, dan mengelola informasi frekuensi atau kode karakter secara efisien.

Permasalahan di lapangan

- Ukuran data teks terus bertambah sehingga membutuhkan ruang penyimpanan lebih besar.
- Transmisi data menjadi lebih lambat karena ukuran file yang besar.
- Pengelolaan data tanpa kompresi sederhana tidak lagi efisien.
- Pemilihan struktur data yang kurang tepat menyebabkan proses kompresi/dekompresi lambat.

Tujuan proyek

Membangun sistem kompresi data teks sederhana dan menganalisis struktur data yang paling efisien untuk proses pengkodean dan pengelolaan data kompresi.

Domain data (minimal)

Data teks (string)

Karakter

Frekuensi kemunculan karakter

Kode hasil kompresi

Ukuran data sebelum dan sesudah kompresi

Aktivitas sistem wajib

Insert / input data teks

Analisis frekuensi karakter

Generate kode kompresi sederhana

Search kode karakter

Dekompresi data teks

Fokus analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan analisis frekuensi karakter
- Kecepatan proses encoding dan decoding
- Efisiensi penggunaan memori
- Dampak pertumbuhan ukuran data terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario penggunaan data teks
- Implementasi minimal 1 struktur data
- Dataset awal (teks dummy / simulasi)
- Pengujian awal analisis frekuensi & encoding sederhana

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi kompresi dan dekompresi data sederhana
- Laporan analisis performa & rekomendasi struktur data

Catatan:

Mahasiswa tidak diwajibkan mengimplementasikan algoritma kompresi optimal, cukup yang sederhana saja. Penilaian difokuskan pada penggunaan dan analisis struktur data dalam sistem kompresi data sederhana.

Topik 5. Sistem Manajemen Jadwal & Konflik Ruang

Konteks industri: perkantoran, kampus, coworking space

Kasus ini menggambarkan sistem pengelolaan jadwal penggunaan ruang (ruang rapat, ruang kelas, atau ruang kerja bersama) pada organisasi dengan banyak pengguna. Pada awalnya, pengaturan jadwal dilakukan secara manual atau menggunakan tabel sederhana sehingga masih dapat dikendalikan. Namun, seiring meningkatnya jumlah pengguna, variasi kegiatan, dan keterbatasan jumlah ruang, konflik jadwal sering terjadi dan sulit dideteksi secara cepat. Sistem membutuhkan pendekatan struktur data yang efisien agar pencatatan jadwal, pencarian ketersediaan ruang, dan pendekripsi konflik dapat dilakukan secara cepat meskipun data jadwal bertambah. Pada topik ini, sistem tidak diwajibkan melakukan optimasi penjadwalan otomatis atau algoritma kompleks. Fokus utama adalah pada pengelolaan data jadwal dan deteksi konflik sederhana menggunakan struktur data yang tepat.

Permasalahan di lapangan

- Konflik penggunaan ruang sulit dideteksi ketika jumlah jadwal meningkat.
- Pencarian jadwal ruang tertentu menjadi lambat jika data disimpan secara sederhana.
- Update dan pembatalan jadwal membutuhkan proses pencarian yang efisien.
- Sistem tidak mampu menangani pertumbuhan data jadwal secara stabil.

Tujuan proyek

Membangun sistem manajemen jadwal dan deteksi konflik ruang yang efisien, serta membandingkan performa struktur data yang digunakan dalam pengelolaan data jadwal.

Domain data (minimal)

ID ruang

Nama ruang

ID jadwal

Tanggal

Waktu mulai

Waktu selesai

Nama kegiatan

Status jadwal

Aktivitas sistem wajib

Insert data jadwal

Search jadwal berdasarkan ruang atau waktu

Update data jadwal

Delete jadwal

Deteksi konflik jadwal sederhana (waktu tumpang tindih pada ruang yang sama)

Catatan: Deteksi konflik cukup menggunakan pengecekan interval waktu, tanpa optimasi penjadwalan.

Fokus analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan pencarian jadwal
- Efisiensi deteksi konflik jadwal
- Kemudahan insert, update, dan delete
- Dampak pertumbuhan jumlah jadwal terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario penggunaan ruang
- Implementasi minimal 1 struktur data
- Dataset awal jadwal (dummy / simulasi)
- Pengujian awal insert dan pencarian jadwal

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi manajemen jadwal & konflik ruang
- Laporan analisis performa & rekomendasi struktur data

Catatan:

Mahasiswa tidak diwajibkan mengimplementasikan algoritma penjadwalan atau optimasi konflik tingkat lanjut. Fokus penilaian adalah pada penggunaan struktur data dan analisis performanya dalam sistem manajemen jadwal.

Topik 6. Sistem Rekomendasi Produk Berbasis Riwayat Transaksi

Konteks industri: e-commerce

Kasus ini menggambarkan sistem rekomendasi sederhana pada platform e-commerce yang memanfaatkan riwayat transaksi pelanggan. Pada tahap awal, rekomendasi produk dapat dilakukan secara manual atau berbasis aturan sederhana, seperti menampilkan produk terlaris atau produk yang sering dibeli secara bersamaan. Namun, seiring bertambahnya jumlah transaksi, produk, dan pengguna, pencarian pola pembelian dan penyajian rekomendasi menjadi semakin lambat jika data dikelola secara sederhana. Oleh karena itu, diperlukan pemilihan struktur data yang tepat agar pengelolaan riwayat transaksi dan penyusunan rekomendasi tetap efisien meskipun volume data meningkat. Pada topik ini, sistem rekomendasi tidak berbasis machine learning. Rekomendasi dibangun menggunakan aturan sederhana (rule-based) yang dapat dijelaskan secara eksplisit melalui perhitungan frekuensi dan pencarian data. Bentuk rekomendasi yang dikembangkan dibatasi pada rekomendasi produk paling sering dibeli (Top-N produk), rekomendasi produk yang sering dibeli bersama dalam transaksi yang sama (*frequently bought together*), serta rekomendasi berbasis riwayat pelanggan, yaitu menyarankan produk yang sering muncul pada transaksi pelanggan lain dengan pola pembelian serupa. Seluruh rekomendasi yang dihasilkan harus dapat dijelaskan secara logis menggunakan data transaksi yang tersedia, tanpa melibatkan model prediktif atau proses pembelajaran otomatis.

Permasalahan di lapangan

- Riwayat transaksi semakin besar dan sulit dicari secara cepat.

- Penyusunan rekomendasi produk menjadi lambat jika data tidak terstruktur dengan baik.
- Update data transaksi dan produk memengaruhi performa sistem.
- Sistem sederhana tidak mampu menangani pertumbuhan data transaksi.

Tujuan proyek

Membangun sistem rekomendasi produk sederhana berbasis riwayat transaksi, serta membandingkan performa struktur data yang digunakan untuk pengelolaan data transaksi dan penyusunan rekomendasi.

Domain data (minimal)

- ID transaksi
- ID pelanggan
- ID produk
- Nama produk
- Kategori produk
- Jumlah pembelian
- Tanggal transaksi

Aktivitas sistem wajib

- Insert data transaksi
- Search riwayat transaksi berdasarkan pelanggan atau produk
- Analisis sederhana pola pembelian
(misalnya: produk paling sering dibeli, atau produk yang sering muncul dalam transaksi yang sama)
- Generate rekomendasi produk sederhana
- Delete atau update data transaksi

Fokus analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan pencarian riwayat transaksi
- Efisiensi penyusunan rekomendasi produk berbasis frekuensi
- Kemudahan insert, update, dan delete data transaksi
- Dampak pertumbuhan jumlah transaksi terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario transaksi e-commerce
- Implementasi minimal 1 struktur data
- Dataset awal transaksi (dummy / simulasi)
- Pengujian awal insert dan search transaksi

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi sistem rekomendasi produk sederhana
- Laporan analisis performa & rekomendasi struktur data

Catatan

Mahasiswa tidak diwajibkan dan tidak dinilai berdasarkan implementasi:

- Machine learning
- Collaborative filtering
- Content-based filtering
- Similarity matrix atau model prediktif lainnya

Fokus penilaian adalah pada penggunaan struktur data, pengelolaan data transaksi, dan analisis performa sistem dalam menghasilkan rekomendasi sederhana berbasis riwayat transaksi. Analisis pola pembelian bersifat sederhana dan berbasis hitungan frekuensi (counting), bukan berbasis prediksi, pembelajaran model, atau perhitungan similarity.

Topik 7. Sistem Deteksi Duplikasi Data

Konteks industri: data warehouse, arsip digital

Pada lingkungan arsip digital, data dari berbagai sumber sering dikumpulkan secara berkala. Dalam proses ini, sangat mungkin terjadi duplikasi data, baik berupa file yang sama, record dengan isi identik, maupun data dengan metadata yang sama. Jika tidak ditangani, duplikasi akan menyebabkan pemborosan penyimpanan, memperlambat pencarian data, dan menurunkan kualitas informasi. Pada tahap awal, pendekripsi duplikasi dapat dilakukan secara manual atau dengan pencarian sederhana. Namun, seiring bertambahnya volume data, proses pengecekan duplikasi menjadi semakin lambat dan tidak efisien jika struktur data yang digunakan tidak tepat. Oleh karena itu, diperlukan pemilihan struktur data yang sesuai agar deteksi, pencarian, dan pengelolaan data duplikat dapat dilakukan secara efisien. Pada topik ini, sistem deteksi duplikasi tidak menggunakan teknik kompleks, fingerprinting, atau machine learning. Fokus utama adalah bagaimana struktur data menyimpan dan membandingkan data untuk mendeteksi duplikasi secara sederhana dan efisien.

Permasalahan di Lapangan

- Volume data terus bertambah dan sulit diperiksa secara manual
- Pencarian data duplikat menjadi lambat jika dilakukan secara linear
- Penyimpanan data duplikat meningkatkan penggunaan memori
- Update dan penghapusan data memengaruhi konsistensi sistem
- Sistem sederhana tidak mampu menangani pertumbuhan data dalam jangka panjang

Tujuan Proyek

Membangun sistem deteksi duplikasi data sederhana serta menganalisis performa berbagai struktur data dalam menyimpan, mencari, dan mendeteksi data duplikat.

Domain Data (Minimal)

- ID data / ID dokumen
- Nama data / nama file
- Ukuran data
- Tanggal unggah
- Sumber data
- Konten sederhana (misalnya string atau representasi ringkas data)

Aktivitas Sistem Wajib

- Insert data baru
- Search data berdasarkan ID atau nama
- Deteksi data duplikat berdasarkan:
 - isi data
 - atau metadata (misalnya nama + ukuran)
- Menampilkan daftar data duplikat

- Delete atau update data
- Statistik sederhana (jumlah data unik vs duplikat)

Fokus Analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan deteksi data duplikat
- Efisiensi pencarian data
- Kemudahan insert, update, dan delete
- Dampak pertumbuhan jumlah data terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario arsip data
- Implementasi minimal 1 struktur data
- Dataset dummy data arsip
- Pengujian awal insert dan deteksi duplikasi

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi sistem deteksi duplikasi data sederhana
- Laporan analisis performa & rekomendasi struktur data

Catatan

Mahasiswa tidak diwajibkan mengimplementasikan algoritma hashing kriptografi, checksum kompleks, machine learning, atau teknik deduplikasi lanjutan. Deteksi duplikasi dilakukan menggunakan perbandingan data dan perhitungan frekuensi sederhana, dengan fokus utama pada penggunaan dan analisis struktur data.

Topik 8. Sistem Manajemen Versi Dokumen Sederhana

Konteks industri: software development, manajemen dokumen

Dalam lingkungan pengembangan perangkat lunak maupun manajemen dokumen, satu dokumen sering mengalami banyak perubahan dari waktu ke waktu. Tanpa sistem manajemen versi yang baik, pengguna akan kesulitan melacak perubahan, mengakses versi sebelumnya, dan menghindari konflik akibat pembaruan dokumen yang tidak terkelola. Pada tahap awal, pengelolaan versi sering dilakukan secara manual, misalnya dengan menyalin file dan memberi nama berbeda. Namun, seiring bertambahnya jumlah dokumen dan versi, pendekatan ini menjadi tidak efisien dan rawan kesalahan. Oleh karena itu, diperlukan sistem manajemen versi dokumen sederhana yang mampu menyimpan, mencari, dan mengelola versi dokumen secara efisien dengan memanfaatkan struktur data yang tepat. Struktur data juga harus dapat menyimpan versi dokumen dan memungkinkan rollback ke versi sebelumnya.

Pada topik ini, sistem manajemen versi tidak mengimplementasikan sistem version control kompleks seperti Git, branching lanjutan, merging otomatis, atau diff berbasis algoritma teks kompleks. Fokus utama adalah pengelolaan data versi dokumen menggunakan struktur data dasar.

Permasalahan di Lapangan

- Banyak versi dokumen sulit dilacak secara manual
- Pencarian versi tertentu menjadi lambat saat jumlah versi bertambah
- Penyimpanan versi dokumen tidak terstruktur dengan baik
- Update versi dokumen memengaruhi konsistensi data

- Sistem sederhana tidak mampu menangani pertumbuhan jumlah dokumen dan versi

Tujuan Proyek

Membangun sistem manajemen versi dokumen sederhana serta menganalisis performa berbagai struktur data dalam menyimpan, mengelola, dan mencari versi dokumen.

Domain Data (Minimal)

- ID dokumen
- Nama dokumen
- Nomor versi
- Tanggal perubahan
- Penulis / editor
- Konten dokumen (string sederhana atau ringkasan)

Aktivitas Sistem Wajib

- Insert dokumen baru
- Menambahkan versi baru dokumen
- Search dokumen berdasarkan nama atau ID
- Menampilkan seluruh versi suatu dokumen
- Mengakses versi tertentu (misalnya versi ke-3)
- Delete atau rollback ke versi sebelumnya

Fokus Analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan pencarian dokumen dan versi
- Efisiensi penambahan versi baru
- Kemudahan rollback dan delete versi
- Dampak pertumbuhan jumlah dokumen dan versi terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario pengelolaan dokumen
- Implementasi minimal 1 struktur data
- Dataset dummy dokumen dan versi
- Pengujian awal insert dan search versi

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi sistem manajemen versi dokumen sederhana
- Laporan analisis performa & rekomendasi struktur data

Catatan

Mahasiswa tidak diwajibkan mengimplementasikan sistem version control kompleks seperti Git, branching, merging, conflict resolution otomatis, atau algoritma perbandingan teks (diff) tingkat lanjut. Sistem versi dibangun secara linear dan sederhana, dengan fokus pada penggunaan struktur data dan analisis performanya.

Topik 9. Sistem Monitoring & Log Aktivitas Aplikasi

Konteks industri: backend system, DevOps

Pada sistem backend dan aplikasi skala menengah, setiap aktivitas sistem—seperti login pengguna, permintaan API, error, dan proses internal—akan menghasilkan log. Log ini

berfungsi sebagai sumber utama untuk pemantauan kondisi aplikasi, pelacakan error, dan analisis performa sistem.

Pada tahap awal, log sering disimpan secara sederhana dalam bentuk file atau daftar tanpa struktur yang jelas. Namun, seiring meningkatnya jumlah pengguna dan aktivitas aplikasi, volume log bertambah sangat cepat. Tanpa pengelolaan data yang baik, pencarian log tertentu, analisis kejadian, dan pemantauan sistem menjadi lambat dan tidak efisien. Oleh karena itu, diperlukan sistem monitoring dan log aktivitas sederhana yang mampu menyimpan, mencari, dan menganalisis log secara efisien dengan memanfaatkan struktur data yang tepat. Pada topik ini, sistem monitoring tidak mengimplementasikan distributed tracing, real-time alerting kompleks, atau analitik log berbasis machine learning. Fokus utama adalah pengelolaan dan analisis log menggunakan struktur data dasar.

Permasalahan di Lapangan

- Volume log meningkat sangat cepat
- Pencarian log berdasarkan waktu atau jenis kejadian menjadi lambat
- Log error sulit dipisahkan dari log normal
- Penyimpanan log tidak terstruktur
- Sistem sederhana tidak mampu menangani pertumbuhan data log

Tujuan Proyek

Membangun sistem monitoring dan log aktivitas aplikasi sederhana serta menganalisis performa berbagai struktur data dalam menyimpan, mencari, dan mengelola data log.

Domain Data (Minimal)

- ID log
- Timestamp
- Level log (INFO, WARNING, ERROR)
- Sumber modul / service
- Pesan log

Aktivitas Sistem Wajib

- Insert log baru
- Search log berdasarkan:
 - waktu
 - level log
 - sumber modul
- Menampilkan log error
- Delete log lama (misalnya berdasarkan waktu)
- Statistik sederhana (jumlah log per level)

Fokus Analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan insert log
- Kecepatan pencarian log
- Efisiensi penghapusan log lama
- Dampak pertumbuhan volume log terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario aktivitas aplikasi
- Implementasi minimal 1 struktur data
- Dataset log dummy (simulasi aktivitas aplikasi)

- Pengujian awal insert dan search log

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi sistem monitoring & log aktivitas sederhana
- Laporan analisis performa & rekomendasi struktur data

Catatan

Mahasiswa tidak diwajibkan mengimplementasikan sistem monitoring kompleks seperti real-time alerting, distributed logging, dashboard visual lanjutan, atau analisis log berbasis machine learning. Sistem log dibangun secara sederhana dan terpusat, dengan fokus pada penggunaan struktur data dan analisis performanya.

Topik 10. Sistem Manajemen Kategori & Hierarki Data

Konteks industri: e-commerce, arsip digital, manajemen konten

Pada berbagai sistem informasi seperti platform e-commerce, arsip digital, dan sistem manajemen konten, data sering disusun dalam bentuk kategori dan subkategori yang bersifat hierarkis. Contohnya adalah kategori produk, klasifikasi dokumen, atau struktur menu aplikasi. Struktur hierarki ini memudahkan pengguna dalam melakukan pengelompokan, pencarian, dan navigasi data.

Pada tahap awal, kategori dan relasinya sering dikelola secara sederhana, misalnya menggunakan daftar datar (flat list) dengan penanda induk-anak. Pendekatan ini masih dapat digunakan ketika jumlah kategori sedikit. Namun, seiring bertambahnya jumlah kategori, kedalaman hierarki, dan data yang terhubung, pengelolaan kategori menjadi semakin kompleks. Operasi pencarian kategori tertentu, penelusuran subkategori, serta penghapusan kategori beserta turunannya menjadi lambat dan rawan kesalahan jika tidak menggunakan struktur data yang tepat.

Oleh karena itu, diperlukan sistem manajemen kategori dan hierarki data sederhana yang mampu menyimpan, menelusuri, dan mengelola data bertingkat secara efisien. Pada topik ini, sistem tidak membahas optimasi navigasi berbasis AI, pencarian semantik, atau rekomendasi kategori otomatis. Fokus utama adalah pemilihan dan analisis struktur data untuk mengelola hierarki data dan dampaknya terhadap performa sistem.

Permasalahan di Lapangan

- Jumlah kategori dan subkategori terus bertambah
- Struktur data datar menyulitkan penelusuran hierarki
- Pencarian kategori dan subkategori menjadi lambat
- Penghapusan kategori yang memiliki turunan berisiko inkonsistensi data
- Sistem sederhana tidak mampu menangani pertumbuhan dan kedalaman hierarki data

Tujuan Proyek

Membangun sistem manajemen kategori dan hierarki data sederhana serta menganalisis performa berbagai struktur data dalam menyimpan, menelusuri, dan mengelola data bertingkat.

Domain Data (Minimal)

- ID kategori
- Nama kategori
- ID induk (parent)

- Level / kedalaman kategori
- Status kategori (aktif / nonaktif)

Aktivitas Sistem Wajib

- Insert kategori baru
- Menambahkan subkategori ke kategori tertentu
- Search kategori berdasarkan ID atau nama
- Menampilkan seluruh subkategori dari suatu kategori
- Menampilkan struktur hierarki kategori
- Delete kategori (beserta subkategorinya)

Fokus Analisis

Analisis perbandingan struktur data terhadap:

- Kecepatan pencarian kategori
- Efisiensi penelusuran hierarki data
- Kemudahan insert dan delete kategori
- Dampak pertumbuhan jumlah dan kedalaman kategori terhadap performa sistem

Target Minggu ke-7 (Progress)

- Spesifikasi sistem & skenario penggunaan kategori
- Implementasi minimal 1 struktur data
- Dataset kategori dummy (simulasi hierarki data)
- Pengujian awal insert dan search kategori

Target Minggu ke-14 (Final)

- Implementasi lengkap ≥ 2 struktur data
- Grafik perbandingan waktu eksekusi & penggunaan memori
- Demo aplikasi sistem manajemen kategori & hierarki data
- Laporan analisis performa & rekomendasi struktur data

Catatan

Mahasiswa tidak diwajibkan mengimplementasikan sistem manajemen hierarki tingkat lanjut seperti optimasi navigasi otomatis, pencarian berbasis AI, atau pengelolaan hak akses kompleks. Sistem dibangun secara sederhana dengan fokus pada penggunaan struktur data dan analisis performanya dalam menangani data bertingkat.

D. Dataset Publik yang Bisa Digunakan untuk Mendukung Simulasi dan Percobaan

Perbandingan Struktur Data

1. Inventori & Transaksi

- [Online Retail II – UCI Machine Learning Repository \(transaksi ritel nyata\)](#)
 - Data transaksi retail online (1+ juta entri) cocok untuk simulasi inventori, **rekomendasi produk, deteksi duplikasi, dan log aktivitas**.
- [Daily Transactions Dataset – Kaggle \(transaksi harian\)](#)
 - Dataset transaksi dummy yang baik untuk latihan operasional struktur data (hash, search).
- [Financial Transactions Dataset – Kaggle \(rekaman transaksi finansial\)](#)
 - Cocok sebagai data input untuk sistem **log aktivitas** atau **analisis anomalai/duplikasi**.

- [Transaction Records Dataset \(Fintech\) – Zenodo](#)
 - Dataset transaksi e-wallet nyata (Excel). Ideal untuk sistem **rekonsiliasi, hashing**, dan **performa query berdasarkan waktu**.

2. Rute & Graph

- [Stanford Large Network Dataset Collection \(graf\)](#)
 - Koleksi graf besar (social, komunikasi, web, produk, dsb.) bagus untuk eksperimen **graph traversal, rute**, dan **network analysis**.

3. Pengiriman / Logistik

- [LaDe – Last Mile Delivery Dataset \(logistik nyata\)](#)
 - Dataset rute pengiriman paket besar dari industri logistik (CSV). Dapat digunakan untuk **routing algoritma** dan **optimasi jadwal/konflik**.

4. Autocomplete / Teks

- [AmazonQAC – Query Autocomplete Dataset \(autocomplete kata\)](#)
 - Dataset besar autocomplete kata/kueri untuk eksperimen dengan struktur data *Trie* dan perbandingan dengan *hash table*.

5. Data Umum & Open Data Portal

Mahasiswa bebas menggunakan portal-portal lain untuk mencari dataset publik yang sesuai dengan kebutuhan, seperti Kaggle, Data.gov, dsb.

E. Format Laporan Kemajuan (Minggu ke-7)

Laporan kemajuan bertujuan memastikan proyek **berjalan, terarah, dan sudah mengandung eksperimen awal**, bukan sekadar rencana.

Ketentuan umum

- Panjang: **5–8 halaman** (di luar cover & daftar pustaka)
- Format: PDF, spasi 1.5, font 11–12
- Wajib menyertakan **hasil eksperimen awal**

Struktur Laporan Kemajuan (format sitasi dan segala hal terkait format yang tidak dijelaskan di sini, mengikuti ketentuan di PPKI IPB)

1. Halaman Judul

Judul proyek, nama anggota & NIM, kelas/praktikum, topik yang dipilih.

2. Pendahuluan

- Latar belakang masalah (konteks industri)
- Rumusan masalah
- Tujuan proyek

3. Deskripsi Data & Skenario Uji

- Sumber dataset (tautan & deskripsi singkat)
- Karakteristik data (jumlah record, atribut utama)
- Skenario penggunaan (use case utama)

4. Desain Awal Solusi

- Arsitektur program secara umum

- Alur proses utama (boleh diagram)
 - Struktur data yang **direncanakan** digunakan
5. **Implementasi Sementara**
- Struktur data yang **sudah berhasil diimplementasikan**
 - Penjelasan operasi utama (insert, search, delete)
6. **Eksperimen Awal & Hasil Sementara**
- Setup eksperimen (ukuran data, spesifikasi singkat)
 - Tabel/hasil waktu eksekusi awal
 - Analisis singkat (belum final)
7. **Kendala & Rencana Lanjutan**
- Kendala teknis/non-teknis
 - Rencana pekerjaan hingga minggu ke-14
8. **Daftar Pustaka**

F. Format Laporan Akhir (Minggu ke-14)

Laporan akhir merupakan **dokumen utama penilaian proyek** dan harus mencerminkan proses eksperimen yang lengkap dan analitis.

Ketentuan umum

- Panjang: **15–25 halaman** (di luar cover & lampiran)
- Format: PDF, spasi 1.5, font 11–12
- Wajib menyertakan **analisis perbandingan struktur data**

Struktur Laporan Akhir (format sitasi dan segala hal terkait format yang tidak dijelaskan di sini, mengikuti ketentuan di PPKI IPB)

1. **Halaman Judul**
2. **Abstrak**
Ringkasan masalah, metode, struktur data yang dibandingkan, dan hasil utama (± 150 kata).
3. **Pendahuluan**
 - Latar belakang & urgensi masalah
 - Rumusan masalah
 - Tujuan dan batasan proyek
4. **Landasan Teori & Studi Terkait**
 - Konsep struktur data yang digunakan
 - Kompleksitas waktu & ruang teoritis
 - Studi/kasus sejenis (ringkas)
5. **Desain Sistem & Metodologi**
 - Desain arsitektur program
 - Penjelasan setiap struktur data yang digunakan
 - Algoritma utama (flow/diagram/pseudocode)

6. Implementasi

- Detail implementasi tiap struktur data
- Mekanisme penyimpanan file (read/write)
- Penjelasan modul penting

7. Eksperimen & Pengujian

- Skenario uji (ukuran data, variasi input)
- Metode pengukuran waktu & memori
- Replikasi eksperimen

8. Hasil & Analisis

- Tabel dan grafik perbandingan performa
- Analisis waktu (insert/search/delete)
- Analisis penggunaan memori
- Diskusi trade-off

9. Kesimpulan & Rekomendasi

- Ringkasan temuan utama
- Struktur data paling optimal dan alasannya
- Saran pengembangan lanjutan

10. Disclosure Penggunaan AI (jika ada)

- Bagian laporan yang menggunakan AI
- Penjelasan pemahaman penulis

11. Daftar Pustaka**12. Lampiran (opsional)**

- Potongan kode penting
- Data tambahan eksperimen

G. Rubrik Penilaian Laporan Kemajuan (Minggu ke-7)

Bobot laporan dan presentasi kemajuan adalah 25% dari total nilai proyek akhir.

No	Komponen Penilaian	Deskripsi	Bobot (%)
1	Perumusan Masalah & Tujuan	Kejelasan konteks industri, rumusan masalah, dan tujuan proyek	10
2	Dataset & Skenario Uji	Dataset jelas, relevan, berukuran realistik, dan skenario uji masuk akal	15
3	Desain Awal Sistem	Alur sistem logis, struktur data dipilih dengan alasan yang tepat	15
4	Implementasi Sementara	Sudah ada kode berjalan untuk ≥ 1 struktur data	20
5	Eksperimen Awal	Ada hasil uji awal (tabel/grafik), bukan sekadar rencana	20

6	Analisis Awal	Interpretasi hasil awal, meskipun belum final	10
7	Kerapihan & Sistematika	Format rapi, bahasa baku, alur laporan jelas	10
	Total		100

Catatan penting:

- Laporan tanpa eksperimen **tidak dapat nilai >60**.
- Screenshot tanpa analisis **tidak dianggap eksperimen**.

H. Rubrik Penilaian Laporan Akhir (Minggu ke-14)

No	Komponen Penilaian	Deskripsi	Bobot (%)
1	Pendahuluan & Urgensi	Masalah nyata, urgensi jelas, tujuan terdefinisi	10
2	Landasan Teori	Pemahaman struktur data & kompleksitas, relevan dengan proyek	10
3	Desain & Metodologi	Desain sistem matang, metodologi eksperimen jelas	15
4	Implementasi	Kode modular, benar secara logika, memanfaatkan struktur data dengan tepat	20
5	Eksperimen & Pengujian	Eksperimen sistematis, skenario uji bervariasi	20
6	Analisis & Diskusi	Analisis tajam, mampu menjelaskan trade-off	15
7	Kesimpulan & Rekomendasi	Kesimpulan berbasis data, bukan opini	5
8	Kualitas Laporan	Struktur rapi, bahasa baku, sitasi sesuai PPKI IPB	5
	Total		100

I. Berkas yang dikumpulkan (5 hari sebelum Minggu 14):

- PPT presentasi
- Video presentasi berdurasi maksimal 15 menit.
- Laporan proyek akhir
- Produk (program yang dihasilkan)

