

Panduan Final Project (UTS dan UAS)

Sistem Terdistribusi

Prodi : Teknik Informatika – FTTK UMRAH
Nama MK : Sistem Terdistribusi
Semester : V

A. Mekanisme Pengerjaan Final Project

1. Setiap mahasiswa tergabung dalam sebuah kelompok Final Project.
2. Kelompok sesuai dengan kelompok presentasi tugas kuliah
3. Membuat Proposal Final Project.
4. Membuat Laporan Akhir Final Project.
5. Setiap kelompok mengerjakan Final Project sesuai studi kasus yang ditetapkan.
6. Mengunggah source code final project pada laman Github. (Github harus dapat diakses oleh public).
7. Melakukan presentasi akhir final project.

B. Kriteria Penilaian Final Project

- | | |
|---------------------------------------|-------|
| 1. Fitur aplikasi (Tingkat Kerumitan) | : 25% |
| 2. Keberhasilan aplikasi | : 30% |
| 3. Laporan akhir | : 30% |
| 4. Presentasi akhir dan demo | : 15% |

C. Format Laporan Final Project

Abstraksi

- Gambaran umum sistem (singkat)
- Tujuan pembuatan/desain sistem
- Hanya 1 paragraph (maks 250 kata)

#1. Pendahuluan

- Gambaran umum sistem yg dijelaskan secara detail
- Landasan pembuatan system beserta rumusan masalah yang diangkat
- Penjelasan tujuan desain sistem

#2. Literature review

- Penjelasan dasar teori yang digunakan
- Pembahasan sistem terdahulu yg sesuai dengan topik desain (referensi)

#3. Desain sistem

- Penjelasan teknologi yg mendukung sistem (arsitektur sistem)
- Penjelasan bagaimana desain sistem bekerja (Berisikan flowchart dan gambar rancangan sistem)

#4. Implementasi sistem

- Pembahasan implementasi system
- Tampilan dan penjelasan fitur-fitur aplikasi dari sistem
- Pengujian sistem

#5. Kesimpulan

- 1 Paragraf kesimpulan akhir dr desain sistem, dan menjawab rumusan masalah.

Referensi

- Daftar referensi yang digunakan

D. Jadwal Pengerjaan Final Project (Bulan/Tanggal)

- | | |
|----------------------|-------------------------------------|
| 1. Nov/25 - Nov/29 | Progress Report 1 |
| 2. Des/2 - Des/6 | Progress Report 2 |
| 3. Des /9 - Des /13 | Progress Report 3 |
| 4. Des /16 - Des /20 | Progress Report 4 |
| 5. Des /23 - Des /27 | Progress Report 5 |
| 6. Jan/6 - Jan/10 | Final Project Presentation and Demo |

1. Studi Kasus Proyek: Aplikasi Chat Terdistribusi (Multi-server Chat Application)

Deskripsi:

- Buat aplikasi chat dimana pengguna dapat mengirim dan menerima pesan secara real-time.
- Aplikasi ini harus dapat dijalankan di beberapa server, memastikan bahwa pesan disinkronkan di semua server dan dikirimkan ke pengguna tanpa keterlambatan, bahkan jika salah satu server mengalami kegagalan.
- Proyek ini memberikan pengetahuan dasar sistem terdistribusi, termasuk sinkronisasi data, komunikasi server, dan toleransi kesalahan.

Rumusan Masalah:

- Bagaimana aplikasi chat terdistribusi dapat memastikan pengiriman pesan secara real-time dan sinkronisasi di beberapa server
- Bagaimana aplikasi chat terdistribusi dapat menangani kegagalan server tanpa kehilangan data atau gangguan layanan?

Langkah Implementasi:

1. Kajian Pustaka (Literature Review)
Lakukan kajian Pustaka untuk mencari referensi mengenai pengembangan aplikasi chat terdistribusi.
Pahami konsep real-time application development, basis data terdistribusi, pengembangan aplikasi pada multi-server, dan fault tolerance.
2. Desain Arsitektur Aplikasi:
Tentukan komponen: antarmuka pengguna, konsep basis data terdistribusi, host server, dll
Tentukan protokol komunikasi untuk interaksi server-to-server dan server-to-klien.
3. Siapkan Beberapa Server:
Pasang aplikasi chat di beberapa server (node).
Implementasikan mekanisme pendaftaran untuk pengguna mendaftar dan masuk.
4. Sinkronisasi Pesan:
Ketika pesan dikirim dari pengguna, pesan tersebut harus disimpan di basis data terdistribusi.
Implementasikan mekanisme untuk menyinkronkan pesan di semua server, memastikan semua pengguna melihat pesan yang sama secara real-time.
5. Tangani Kegagalan Server:
Implementasikan mekanisme deteksi kegagalan server.

Jika server gagal, beban kerjanya harus didistribusikan ke server lain, dan pengguna yang terhubung ke server yang gagal harus terhubung kembali ke server lain tanpa menyadari gangguan tersebut.

6. Pengujian:

Uji aplikasi dengan beberapa pengguna mengirim dan menerima pesan.

Simulasikan kegagalan server untuk memastikan bahwa sistem dapat menanganinya tanpa kehilangan data atau gangguan layanan.

2. Studi Kasus Proyek: Sistem Pemungutan Suara Terdistribusi

Deskripsi:

- Buat aplikasi voting (pemungutan suara) dimana pengguna dapat memberikan suara pada berbagai jajak pendapat, dan hasilnya diagregasikan secara real-time di beberapa server.
- Aplikasi ini harus dapat dijalankan di beberapa server, memastikan bahwa hasil voting disinkronkan di semua server, bahkan jika salah satu server mengalami kegagalan.
- Proyek ini memberikan pengetahuan dasar sistem terdistribusi, termasuk sinkronisasi data, komunikasi server, dan toleransi kesalahan.

Rumusan Masalah:

- Bagaimana sistem pemungutan suara terdistribusi dapat memastikan agregasi suara yang akurat dan real-time di beberapa server sambil mencegah pemungutan suara ganda dan memastikan integritas data?

Langkah Implementasi:

1. Kajian Pustaka (Literature Review)
Lakukan kajian Pustaka untuk mencari referensi mengenai pengembangan aplikasi. Pahami konsep real-time application development, basis data terdistribusi, pengembangan aplikasi pada multi-server.
2. Desain Sistem:
Tentukan komponen: antarmuka pengguna, basis data pemungutan suara, node server, dll.
3. Pemungutan Suara:
Pengguna memberikan suaranya, yang disimpan di basis data terdistribusi. Implementasikan mekanisme untuk mencegah pemungutan suara ganda.
4. Agregasi Suara:
Agregasikan suara dari semua server secara real-time untuk menampilkan hasil voting.
5. Integritas Data:
Pastikan bahwa suara disimpan dengan aman dan proses agregasi mempertahankan integritas data.
6. Pengujian:
Lakukan beberapa voting dan berikan suara untuk menguji akurasi dan agregasi real-time sistem.
Uji mekanisme keamanan sistem untuk mencegah pemungutan suara ganda dan memastikan integritas data.

3. Studi Kasus Proyek: Sistem Penyimpanan Berkas Terdistribusi

Deskripsi:

- Kembangkan sistem penyimpanan berkas terdistribusi di mana pengguna dapat mengunggah, mengunduh, dan berbagi berkas.
- Sistem harus mendistribusikan berkas di beberapa server untuk memastikan redundansi data dan ketersediaan.
- Proyek ini memberikan pengetahuan dasar sistem terdistribusi, termasuk sinkronisasi data, komunikasi server, dan toleransi kesalahan.

Rumusan Masalah:

- Bagaimana sistem penyimpanan berkas terdistribusi dapat mendistribusikan dan mengambil berkas secara efisien di beberapa server sambil memastikan redundansi data, toleransi kesalahan, dan skalabilitas?

Langkah Implementasi:

1. Kajian Pustaka (Literature Review)
Lakukan kajian Pustaka untuk mencari referensi mengenai pengembangan aplikasi terdistribusi.
Pahami konsep real-time application development, basis data terdistribusi, pengembangan aplikasi pada multi-server, dan fault tolerance.
2. Desain Sistem:
Tentukan komponen: antarmuka pengguna, basis data berkas, node server, dll.
Pilih metode untuk pemecahan dan distribusi berkas.
3. Distribusi Berkas:
Ketika berkas diunggah, bagi menjadi beberapa bagian (chunks).
Distribusikan bagian-bagian ini ke server yang berbeda untuk memastikan redundansi.
4. Pengambilan Berkas:
Ketika berkas diminta, ambil bagian-bagiannya dari berbagai server dan rekonstruksi berkas untuk pengguna.
5. Toleransi Kesalahan:
Implementasikan mekanisme untuk mendeteksi kegagalan server.
Pastikan bagian-bagian berkas direplikasi di beberapa server sehingga sistem masih dapat mengambil berkas meskipun beberapa server gagal.
6. Pengujian:
Uji fungsi unggah, unduh, dan berbagi berkas.
Simulasikan kegagalan server untuk menguji mekanisme pengambilan data dan redundansi.

4. Studi Kasus Proyek: Agregator Feed Berita Terdistribusi

Deskripsi:

- Kembangkan sistem yang mengagregasi artikel berita dari berbagai sumber, memprosesnya, dan menyediakan feed yang disatukan kepada pengguna berdasarkan preferensi dan minat mereka.

Rumusan Masalah:

- Bagaimana sistem terdistribusi dapat mengagregasi, memproses, dan menyampaikan feed berita yang dipersonalisasi kepada pengguna secara real-time dari berbagai sumber?

Langkah Implementasi:

1. Kajian Pustaka (Literature Review)
Lakukan kajian Pustaka untuk mencari referensi mengenai pengembangan aplikasi terdistribusi.
Pahami konsep real-time application development, basis data terdistribusi, pengembangan aplikasi pada multi-server, dan fault tolerance.
2. Identifikasi dan integrasikan berbagai sumber berita menggunakan API atau teknik web scraping.
3. Rancang basis data terdistribusi untuk menyimpan artikel berita, profil pengguna, dan preferensi.
4. Implementasikan penyeimbang beban (Load balancer) untuk mendistribusikan data berita masuk ke beberapa server.
5. Kembangkan mesin rekomendasi untuk menyusun feed berita yang dipersonalisasi untuk pengguna.
6. Implementasikan mekanisme caching untuk mempercepat penyampaian berita.
Pastikan toleransi kesalahan dengan mereplikasi data di beberapa node.
Kembangkan antarmuka pengguna untuk pengguna agar dapat melihat dan menyesuaikan feed berita mereka.
7. Lakukan pengujian sistem untuk memastikan agregasi feed berita diterima dengan baik oleh pengguna.

5. Studi Kasus Proyek: Platform Pengeditan Dokumen Kolaboratif Terdistribusi

Deskripsi:

- Kembangkan sistem dimana beberapa pengguna dapat mengedit dokumen secara kolaboratif secara real-time, dengan perubahan disinkronkan di semua pengguna.

Rumusan Masalah:

- Bagaimana sistem terdistribusi memastikan sinkronisasi real-time dan konsistensi dalam pengeditan dokumen kolaboratif?

Langkah Implementasi:

1. Kajian Pustaka (Literature Review)
Lakukan kajian Pustaka untuk mencari referensi mengenai pengembangan aplikasi terdistribusi.
2. Pahami konsep real-time application development, basis data terdistribusi, pengembangan aplikasi pada multi-server, dan fault tolerance.
3. Rancang basis data terdistribusi untuk menyimpan data dokumen dan suntingan pengguna.
4. Implementasikan Transformasi Operasional (OT) atau Tipe Data Replikasi Bebas Konflik (CRDT) untuk menangani suntingan yang bersamaan.
5. Kembangkan protokol komunikasi real-time untuk menyinkronkan suntingan di antara pengguna.
6. Implementasikan sistem kontrol versi untuk melacak perubahan dan memungkinkan pengembalian.
7. Pastikan redundansi data dan toleransi kesalahan dengan mereplikasi data dokumen di beberapa node.
8. Kembangkan antarmuka pengguna untuk pengeditan dokumen dan kolaborasi.
9. Lakukan pengujian sistem untuk memastikan agregasi feed berita diterima dengan baik oleh pengguna.