

BAB I

PENDAHULUAN

I.1. Latar Belakang

Pada era *Society 5.0* yang merupakan perkembangan dari era *Industry 4.0*, sosial media sudah menjadi kebutuhan pokok bagi kebanyakan orang. Kebutuhan akan informasi setiap individu menjadi sangat besar karena berkembang pesatnya penggunaan aplikasi sosial media seperti *Facebook*, *WhatsApp*, *Line*, *Messenger*, *Twitter*, dan lainnya, segala macam informasi sudah tersedia di platform-platform tersebut. Namun, dalam perkembangannya, tidak semua pengguna paham betul konsep keamanan informasi/privasi mereka.

Kemudian salah satu kelemahan platform *Enterprise* tersebut adalah keamanan informasi penggunanya yang belum absolut, penyimpanan seluruh informasi baik yang publik maupun privat, terletak di satu tempat yang sangat masif. Dengan seperti itu, dibutuhkan pula pekerja yang melakukan pengamanan, pengawasan dan perawatan/maintenance tempat tersebut dengan biaya yang tidak terbilang murah.

Kemampuan GUN dalam menggantikan ribuan baris kode pada fitur semacam *networking calls*, *storage boilerplate*, *pub/sub routing*, *cache invalidation* dan lain sebagainya, sangat membantu dalam pembuatan aplikasi kompleks menjadi lebih mudah dan cepat, dengan mengabstraksikan algoritma-algoritma kompleks tersebut kedalam *integrated API* sederhana yang berbeda dari sistem tradisional. Karena hal tersebut, *production cost* dan waktu dalam pembuatan aplikasi dapat dipangkas semaksimal mungkin.

Lalu, dengan adanya fitur *SEA (Security, Encryption & Authentication)*, diharapkan dapat membantu dalam aspek keamanan aplikasi dan data yang akan dibuat. Dengan menggunakan salah satunya, metode *Diffie-Hellman key exchange* yang terintegrasi di fitur *SEA*, protokol pertukaran data akan lebih otentik dan aman.

Pada pengerjaan tugas akhir ini, penggunaan library GUN adalah sebagai *storage adapter* atau penyimpanan dan *user authentication*, menggunakan hanya dengan backend code seminimal mungkin, dengan tujuan memaksimalkan penggunaan library tersebut.

Dengan fitur *real-time update* dan *offline-first* milik GUN, aplikasi percakapan sederhana dapat dibuat dengan cukup mudah, dengan menggunakan GUN sebagai *storage* untuk setiap *message instance*, kemudian dengan

memanfaatkan sifat GUN yang merupakan data streaming, pengambilan data dapat dilakukan secara instan.

I.2. Rumusan Masalah

Dengan *booming/tren* cryptocurrency dalam beberapa tahun kebelakang, pembahasan tentang *decentralized apps* sangat banyak beredar. Penulis terinspirasi dengan istilah *decentralized*. Selain itu, dengan banyaknya berita serta artikel tentang maraknya *cybercrime* di sosial media terhadap keamanan data dan privasi pengguna membuat penulis ingin mencoba kombinasi istilah *decentralized app* dan *chat app* sehingga terciptalah tulisan ini sebagai dokumentasi dalam pembuatan aplikasi tersebut. Adapun hal yang harus terjawab untuk penelitian ini adalah:

1. Apa sejarah awal pertukaran pesan (*message services*)?
2. Apa itu *GUN*? Fitur apa saja yang ditawarkan?
3. Bagaimana struktur dan cara kerja database terdesentralisasi?
4. Bagaimana arsitektur aplikasi percakapan dengan database terdesentralisasi?

I.3. Tujuan Penelitian

1. Mengetahui sejarah awal pertukaran pesan atau *message services* diseluruh dunia.
2. Mengetahui secara jelas *library GUN* dengan segala fitur yang tersedia.
3. Mengetahui bentuk struktur dan cara kerja *database* terdesentralisasi.
4. Mengetahui dan dapat membuat arsitektur aplikasi percakapan dengan menggunakan teknologi database terdesentralisasi.

I.4. Pembatasan Masalah

Aplikasi ini dibuat terbatas untuk menangani pengiriman pesan *encrypted end-to-end (p2p)*, dan tidak *end-to-end* pada *group chat*, selain itu:

1. Pengguna *dummy* dan sedikit.
2. Sistem yang masih dalam masa perkembangan dan bukan *production ready (hanya MVP)*.
3. *Resource* yang terbatas dan sedikit menggunakan *backend code*.
4. Berjalan di *localhost*

I.5. Manfaat Penelitian

Dalam bidang industry:

1. Memangkas *production cost* dalam pembuatan aplikasi dengan menggunakan API yang disediakan oleh GUN.
2. Sebagai acuan menggunakan library ini untuk menggantikan fitur yang kompleks dalam pembuatan.
3. Dokumentasi penggunaan teknologi *decentralized* pada aplikasi percakapan yang *mainstream*.

I.6. Metode Penelitian dan Perancangan

Metode yang digunakan pada penelitian proyek akhir ini adalah dengan metode SDLC (*Software Development Life Cycle*), yang terdiri dari beberapa langkah yaitu:

1. Identifikasi Masalah

Dalam tahap ini, masukan dari calon pengguna aplikasi atau menganalisis *market demand*, dan mempelajari apa saja kelebihan dan kekurangan dari aplikasi yang dirancang dan melakukan peningkatan dari sistem/aplikasi yang sudah ada

2. Perencanaan (Planning)

Pada tahap ini, dikumpulkan seluruh informasi yang diperlukan dalam pengembangan aplikasi, seperti dokumen rancangan aplikasi, apps phrase, font, layout dan lain sebagainya. Setelah seluruh informasi yang dibutuhkan terkumpul, hal selanjutnya yang dilakukan adalah melakukan perencanaan struktur aplikasi, seperti *database relationship diagram (ERD)*, desain *UI/UX*, fitur-fitur apa saja yang diimplementasikan.

3. Rancangan Aplikasi dan Struktur Pendukung

Di tahap ini, dilakukan perubahan spesifikasi sebuah *software* ke dalam *design plan* yang disebut dengan Desain Dokumen Spesifikasi (DDS), kemudian merancang *UI* hingga *prototyping*.

4. Membangun Aplikasi

Setelah perancangan selesai dilakukan, seluruh catatan dan rencana yang didapatkan akan diimplementasikan dengan membuat aplikasi menggunakan bahasa pemrograman, tahapan ini memakan waktu yang paling lama dari seluruh langkah

5. Pengujian Aplikasi (Testing)

Ketika tahap pembuatan aplikasi selesai, setelahnya akan dilakukan pengetesan terhadap cara kerja aplikasi serta semua fitur yang ada, pada tahap ini segala hal harus berjalan semestinya jika ada bug/error pada aplikasi, biasanya akan mundur kelangkah sebelumnya.

6. Releasing (Deployment)

Setelah dipastikan tidak ada *error/bug*, aplikasi siap di publish untuk publik/*customer*. Pada fase ini, *feedback* pengguna harus diperhatikan, oleh karena itu ada langkah selanjutnya, yaitu *maintenance*

7. Maintenance

Aplikasi yang sudah di *release* ke publik harus dilakukan pemeliharaan/*maintenance*, agar aplikasi dapat terus berjalan dengan baik

Dengan langkah-langkah di atas, digunakanlah model Metodologi *Waterfall* sebagai model yang cocok dalam pengembangan aplikasi ini.

I.6.1. Metodologi *Waterfall*

Perancangan aplikasi dengan cara menyelesaikan tugas secara linier sesuai perencanaan awal serta pembuatan dokumentasi yang sedetail mungkin, dimulai dari perencanaan rancangan *UI wireframe*, implementasi *coding* dan evaluasi atau *debugging*.

I.7. Sistematika Penulisan

Sistematika penulisan yang digunakan pada proyek akhir ini adalah sebagai berikut:

BAB I PENDAHULUAN

Pada bab ini berisi segala informasi dasar yang dibutuhkan dalam penelitian, seperti latar belakang, rumusan masalah, maksud dan tujuan penelitian, pembatasan masalah, metode penelitian dan sistematika penulisan yang digunakan.

BAB II LANDASAN TEORI

Pada bab ini menjelaskan pengertian tentang segala teori yang berhubungan dengan masalah pada proyek akhir ini.

BAB III ANALISIS DAN PERANCANGAN SISTEM

Pada bab ini menjelaskan tentang persiapan dan metode apa saja yang dibutuhkan untuk pengambilan, pengumpulan,

penggunaan dan analisis untuk perancangan sistem.

BAB IV RANCANG BANGUN APLIKASI DAN UJI COBA SERTA ANALISA HASIL

Pada bab ini menjelaskan tata cara dalam pembuatan aplikasi lengkap dengan segala teori dan analisa data yang didapatkan pada bab sebelumnya. Kemudian dilanjut dengan analisa dan hasil, serta pengujian sistem yang dibuat untuk mencari kekurangan-kekurangan dan uji coba terhadap pengguna

BAB V PENUTUP

Pada bab ini berisi kesimpulan atas penelitian yang dilakukan dan hasil akhir dari aplikasi yang dibuat dan beberapa saran untuk penelitian kedepannya.