

Perancangan Aplikasi *Chatting* Symmetric Encryption dengan Metode SDLC

Andi*¹, Jesson, Kevin¹, Melvin Lie¹, Andik Yulianto²

¹Prodi Teknologi Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Jalan Gajah Mada, Baloi Sei Ladi, Batam, Indonesia

²Prodi Sistem Informasi, Fakultas Ilmu Komputer, Universitas Internasional Batam, Jalan Gajah Mada, Baloi Sei Ladi, Batam, Indonesia

ARTICLE INFO

Keywords:

SDLC, Chat Application, Waterfall model, Symmetric Encryption, Cryptography

Received: January 9, 2023

Revised: May 25, 2023

Accepted: June 30, 2023

Corresponding author:

E-mail: 2132065.andi@uib.edu (Andi)

DOI: <http://dx.doi.org/10.37253/telcomatics.v8i1.7385>

ABSTRACT

Chat Application is an application that allows people to communicate with each other without face-to-face meetings or time issues. To design a well-functioning chat application, this study was conducted using the Software Development Life Cycle (SDLC) method for designing chat applications. Apart from that, this study will be using the Python programming language, the PyMySQL module used to connect python to the database, and the Tkinter module is used as a tool in this study to create a Graphical User Interface (GUI) for chat application, and not to mention this study will be using Symmetry Encryption Cryptography to perform encryption and decryption of user chat data.

I. PENDAHULUAN

Aplikasi *Chatting* telah ada sejak tahun 1983, namun pada awalnya manusia yang menggunakan aplikasi *chatting* untuk melakukan kegiatan komunikasi masih jarang, dikarenakan teknologi yang masih kurang canggih dan manusia yang belum terlalu paham dengan teknologi, ditambah juga bahwa manusia sudah memiliki alat untuk melakukan komunikasi yaitu mulut dan melalui gerakan anggota tubuh. Namun seiring berjalannya waktu, teknologi dan pengetahuan manusia terhadap teknologi pun ikut berkembang, kini telah banyak manusia yang telah menggunakan alat virtual untuk dijadikan sebagai alat berkomunikasi antar satu dengan yang lain.

Dengan menggunakan alat virtual ini kita dapat melakukan kegiatan komunikasi satu antar lain secara virtual, yang artinya, kita dapat mengirimkan sebuah informasi tanpa bertemu langsung dengan lawan berbicara, sehingga manusia dapat berkomunikasi dengan siapa saja dan tanpa harus bertemu sehingga kita dapat berkomunikasi pada waktu kapan pun.

Chatting merupakan kegiatan berkomunikasi antara dua orang atau lebih secara *realtime* melalui komputer ataupun telepon genggam dengan koneksi jaringan. Beberapa contoh layanan *chatting* di internet antara lain, WhatsApp, Line, dan Messenger. Dengan adanya layanan *chatting* ini memungkinkan manusia untuk berkomunikasi melalui virtual dengan orang siapa pun dan di mana pun.

Namun alat virtual ini, memiliki bahayanya sendiri, salah satu contohnya adalah bahaya dalam sisi *privacy* ketika data-data dari *chatting* tersebut bocor atau dicuri oleh orang lain.

Dikarenakan ini dalam penelitian ini akan merancang sebuah aplikasi *chatting* dengan fitur enkripsi data-data dari *chatting* pengguna aplikasi tersebut. Berdasarkan latar belakang tersebut, penelitian ini membangun sebuah aplikasi *chatting* dengan GUI yang bagus namun sederhana, dan dapat mengenkripsi data-data *chatting* dari user yang dapat berjalan dengan baik.

II. KAJIAN PUSTAKA

A. Penelitian sebelumnya

Penelitian tentang perancangan aplikasi chat dengan penerapan keamanan data telah dilakukan oleh (Bardis, 2008). Pada penelitian ini kriptografi AES diterapkan untuk keamanan pengiriman data untuk keperluan unit militer. Enkripsi data dilakukan pada beberapa fitur aplikasi diantaranya pembaharuan *password user*, penggantian kunci (*key*) personal, penggantian *key* untuk komunikasi data. Aplikasi ini dikembangkan menggunakan Visual Basic.

Perancangan arsitektur aplikasi percakapan untuk institusi pendidikan juga telah dilakukan pada (Yulianto et al., 2015). Pada penelitian ini aplikasi *chatting* diimplementasikan berbasis web dan Android menggunakan protokol XMPP (Extensible Messaging and Presence Protocol). Aplikasi yang dikembangkan mengirimkan pesan dengan baik, namun demikian belum menerapkan enkripsi untuk keamanan datanya.

Selanjutnya penerapan enkripsi sebagai perangkat keamanan data pada aplikasi *chatting* dilakukan pada

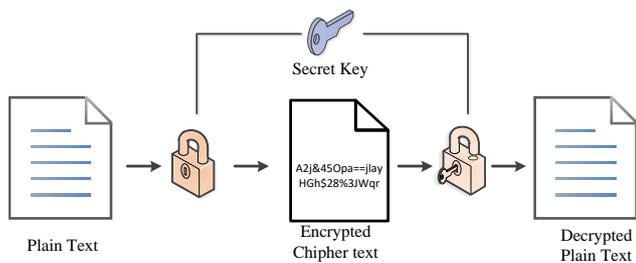
penelitian (Ali & Sagheer, 2017). Pada penelitian ini menerapkan enkripsi AES pada komunikasi pesan teks. Sedangkan generasi pasangan *key* yang digunakan pada enkripsi AES menggunakan algoritma ECDH (Elliptic Curve Diffie Hellman Key Exchange). Untuk enkripsi pesan suara digunakan algoritma RC4 asimetris. Aplikasi telah diuji kecepatannya dalam mengenkripsi data pada beberapa merek *handphone* dengan hasil yang cukup baik.

B. Kriptografi

Kriptografi (*Cryptography*) merupakan sebuah seni mengubah teks yang dapat dibaca oleh manusia menjadi tidak dapat dibaca secara langsung untuk keperluan privasi data atau keamanan data (Chandra et al., 2014). Mengenkripsi dan mendekripsi data menggunakan kode, hal ini memiliki arti bahwa tidak seorang pun kecuali orang yang ditentukan dapat melihat data yang dikirimkan [6].

Terdapat dua jenis kriptografi dasar yaitu 1) *Symmetric Encryption*, dan 2) *Asymmetric Encryption*. Jenis ini didasarkan pada pemakaian kunci (*key*) untuk pada proses sisi enkripsi maupun dekripsi (Chandra et al., 2014). Data akan dienkripsi dengan sebuah *key*, dan data yang terenkripsi tersebut dapat dibaca setelah didekripsi dengan menggunakan *key* tersebut.

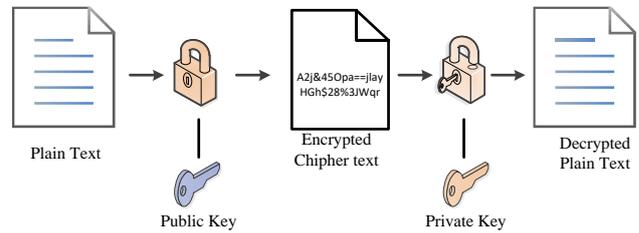
Symmetric Encryption atau Enkripsi Simetris adalah sebuah jenis enkripsi yang menggunakan satu kunci yang sama untuk melakukan enkripsi dan dekripsi pada data. Hal ini menjadikannya lebih efektif dibandingkan dengan *Asymmetric Encryption* (Chandra et al., 2014). Hal ini juga didukung oleh penelitian (Vashi* et al., 2019) yang telah menguji beberapa teknik enkripsi seperti AES, 3DES, Rijndael, RC2 and RSA. Hasilnya *symmetric encryption* lebih cocok dan hemat memory dibandingkan dengan *asymmetric encryption* dalam hal keamanan penyajian data. Gambar 1 menunjukkan alur proses pada *Symmetric Encryption*.



Gambar 1. Symmetric Encryption

Pada *Asymmetric Encryption* menggunakan dua kunci yang berbeda masing-masing untuk enkripsi dan dekripsi. Pada saat proses enkripsi digunakan kunci public (*Public Key*) yang dapat mungkin dapat didistribusikan ketika melibatkan banyak perangkat atau server, sedang kunci pribadi (*Private Key*) digunakan pada saat dekripsi data. Kunci pribadi tidak dibagikan secara publik. *Asymmetric encryption* cocok digunakan untuk enkripsi data dalam jumlah kecil karena membutuhkan waktu yang relatif panjang dibandingkan dengan *Symmetric encryption*. Namun demikian enkripsi jenis ini memiliki tingkat keamanan yang lebih baik dikarenakan menggunakan dua jenis kunci (Santoso et al., 2018). Meskipun kunci publik dapat didistribusikan namun kunci pribadi hanya

dapat diakses oleh satu pengguna yang memperkecil kemungkinan untuk disebar. Contoh *Asymmetric Encryption* antara lain *RSA, DSA, Diffie-Hellman*. Gambar 2 menunjukkan alur proses pada *Asymmetric Encryption*.



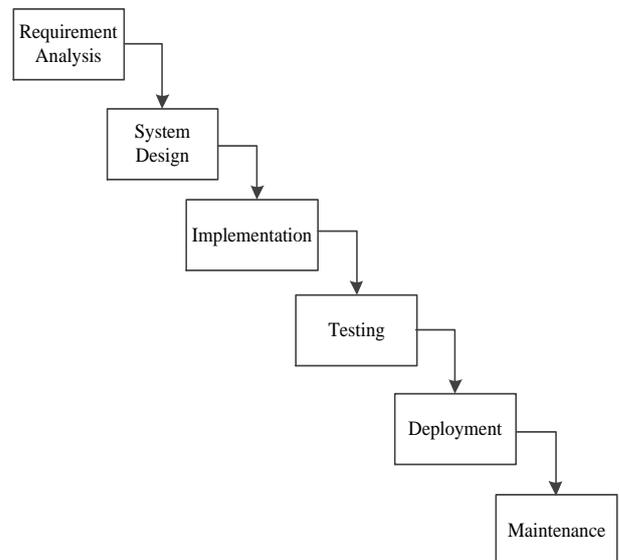
Gambar 2. Asymmetric Encryption

C. SDLC (Software Development Life Cycle)

Software Development Life Cycle (SDLC) merupakan model sistematis yang digunakan untuk pengembangan perangkat lunak. SDLC model yang menggambarkan bagaimana perangkat lunak dikembangkan, diimplementasikan, dirawat dan diganti (Gurung et al., 2020). Terdapat beberapa jenis model yang masuk dalam SDLC diantaranya Waterfall, Spiral, Rapid Application Development (RAD), Prototipe, dan Agile. Masing-masing model memiliki kelebihan dan kekurangan, dan dipilih dengan tujuan agar perangkat lunak dapat dikembangkan dengan semurah mungkin, efisien, dan se-efektif mungkin. (Gurung et al., 2020). Dalam penelitian ini pengembangan aplikasi *chat* menggunakan model Waterfall.

D. Waterfall Model

Waterfall model memiliki siklus sekuensial mirip aliran air terjun. Model *waterfall* harus dilakukan pertahap hingga selesai sebelum pindah ke tahap berikutnya. Dengan demikian tahapannya tidak tumpang tindih satu dengan yang lain. Model ini juga tidak mengijinkan kembali ke proses sebelumnya dan mengubahnya (Gurung et al., 2020). Gambar 3 merupakan diagram model *waterfall*.



Gambar 3. Waterfall Model

E. Unified Modeling Language (UML)

Unified Modeling Language (UML) merupakan bahasa spesifikasi standar digunakan untuk mendokumentasikan, menentukan, dan membuat perangkat lunak. UML adalah metode pengembangan sistem berorientasi objek dan juga alat yang mendukung pengembangan sistem.

Alat yang digunakan dalam Desain berorientasi objek berdasarkan pada UML-nya adalah sebagai berikut (Hendini, 2016).

1. *Use Case Diagram*
Use Case Diagram merupakan pemodelan untuk kelakuan (*Behavior*) sistem informasi yang akan dibuat. *Use Case* juga digunakan untuk mengetahui fungsi apa saja yang ada pada sistem informasi dan siapa yang berhak menggunakan fungsi tersebut.
2. *Activity Diagram*
Activity Diagram mewakili sebuah aliran kerja (*workflow*) atau aktivitas dari sistem serta proses bisnis.
3. *Class Diagram*
Class Diagram adalah hubungan antara kelas dan deskripsi rinci dari setiap kelas dalam model desain sistem yang juga menunjukkan aturan dan tanggung jawab entitas yang akan menentukan perilaku sistem. *Class Diagram* juga menunjukkan *attribute* serta *method* yang terdapat dalam sebuah kelas dan *constraint* yang terkait dengan objek yang terkoneksi.

F. Entity Relationship Diagram (ERD)

Entity Relationship Diagram merupakan diagram yang menggambarkan hubungan antar data dalam sebuah basis data berdasarkan diagram hubungan antara objek (entitas) dengan objek lainnya. Komponen utama yang menyusun model hubungan entitas adalah entitas, atribut, dan hubungan. Ketiga komponen tersebut dapat digambarkan dengan *Entity Relationship Diagram*. ERD terdiri dari lima komponen utama yaitu 1) *Entity* yang menunjukkan objek atau konsep yang terdapat data didalamnya. *Entity* biasanya merujuk pada table yang digunakan pada *database*. 2) *Attributes*, yang menunjukkan properti atau karakteristik dari *entity*, 3) *Relationship*, menyatakan hubungan antar *entity*, 4) *Actions*, yang menjelaskan bagaimana *entity* membagi informasi, dan 5) *Lines*, yaitu garis yang menghubungkan antar *entity*.

III. METODE PENELITIAN

Penelitian ini menggunakan metode *System Development Life Cycle* yang meliputi tahap analisis kebutuhan perangkat lunak, desain perangkat lunak, pembuatan kode program, pengujian perangkat lunak dan perawatan perangkat lunak.

Pada tahap analisis kebutuhan perangkat lunak dilakukan analisis terhadap kebutuhan yang diperlukan dalam sebuah sistem aplikasi chatting. Kemudian tahap desain dilakukan untuk merancang Aplikasi Chatting, dengan menggunakan UML (*unified modelling language*) yaitu, *Use Case*, *Activity Diagram*, dan *Class Diagram*.

Selanjutnya adalah tahap pembuatan kode program. Tahap ini merupakan tahap di mana kami melakukan pembuatan kode untuk program yang akan dibuat, yaitu Aplikasi *Chatting*

dengan menggunakan *Python3*, membuat koneksi antara *database* dengan sistem, dan membuat *Grapical User Interface* (GUI). Tahap selanjutnya adalah tahap pengujian perangkat lunak. Pada tahap ini, dilakukan pengujian terhadap aplikasi *chatting* yang telah dibuat dengan menggunakan metode *Black box testing*. *Black box testing* ini merupakan pengujian terhadap *Software* tersebut terbebas dari *error*, dan melihat apakah *output* telah sesuai dengan fungsionalitas.

IV. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan Perangkat Lunak

Dari hasil analisis yang telah dilakukan oleh kami, sebuah aplikasi *chat* memerlukan elemen-elemen yang penting berikut supaya aplikasinya dapat berjalan dengan baik diantaranya adalah 1) Pengguna (*User*), 2) Teman (*Friend*), 3) Chat, dan 4) Group. Pengguna memiliki peran yang paling penting dalam sebuah aplikasi *chat*, tanpa pengguna maka aplikasi *chat* tidak akan dapat berjalan. Supaya pengguna dapat melakukan aktivitas *chatting* maka membutuhkan pengguna lainnya, supaya teman ini memiliki peran yang penting pada aplikasi *chatting*. Selain dari teman, *chat* ini juga memiliki peran yang penting, tanpa *chat* maka para pengguna tidak dapat melakukan aktivitas *chatting* kepada pengguna lainnya. Berbeda dengan teman, grup merupakan sebuah fitur tambahan kepada aplikasi *chat* yang memungkinkan sekumpulan pengguna dapat melakukan aktivitas *chatting* dalam sebuah "wadah" yang sama. Selain dari empat elemen ini, terdapat juga hal-hal yang lainnya seperti, *Graphical User Interface* (GUI) yang menarik.

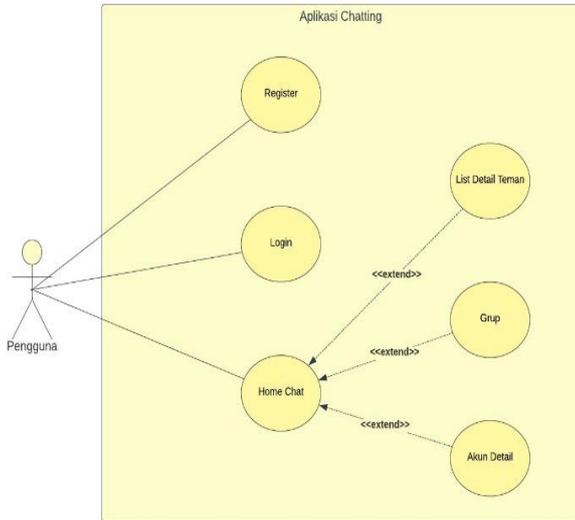
B. Desain Perangkat Lunak

Setelah kami selesai melakukan analisis, Langkah selanjutnya merancang aplikasi *chatting* yang akan dibuat, yaitu dengan menggunakan beberapa alat seperti, *Use Case*, *Activity Diagram*, dan *Class Diagram*. Gambar 3 memperlihatkan apa yang dapat dilakukan oleh pengguna di dalam aplikasi *chatting* ini. Pengguna dapat mengakses tiga halaman utama yaitu halaman Register, Login dan Home Chat. Setelah melakukan registrasi pengguna dapat login dan dapat mengakses halaman Home chat. Pada halaman Home Chat pengguna dapat mengakses halaman List Detail Teman, Group, dan Akun Detail.

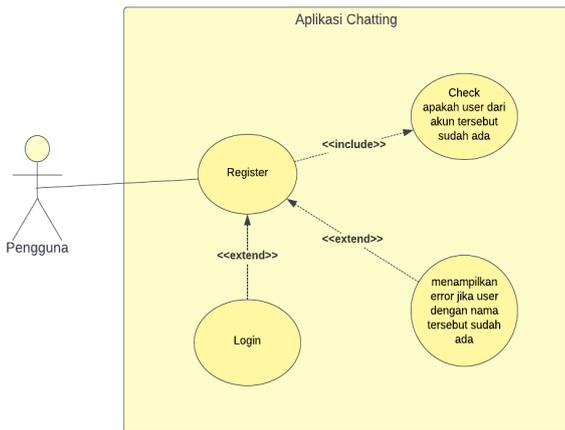
Gambar 5 memperlihatkan proses dari registrasi pengguna. Proses ini dimulai dengan pengguna melakukan *input* pada *Username*, *Password*, *Email*, *Nomor Telepon*, dan *Tentang Pengguna*. Kemudian akan dicek apakah pengguna tersebut sudah ada atau tidak, jika ada maka akan keluar pesan *error* yang mengatakan bahwa pengguna sudah ada, namun jika tidak ada maka pengguna akan berhasil terbuat, terdapat juga proses di mana jika pengguna tidak mengisi apa pun maka akan muncul pesan *error* mengatakan meminta pengguna untuk memasukkan sesuatu. Selain dari itu pengguna juga dapat menuju ke halaman login melewati halaman *register*.

Gambar 6 memperlihatkan proses dari *login* pengguna. Proses ini dimulai dengan pengguna melakukan *input* pada *Username*, dan *Password* kemudian akan dicek apakah *Username*, dan *Password* tersebut betul, jika salah maka akan keluar pesan *error* mengatakan bahwa akun tidak ada, namun jika betul maka pengguna akan menuju ke dalam halaman *home* aplikasi. Selain dari itu pengguna juga dapat menuju ke

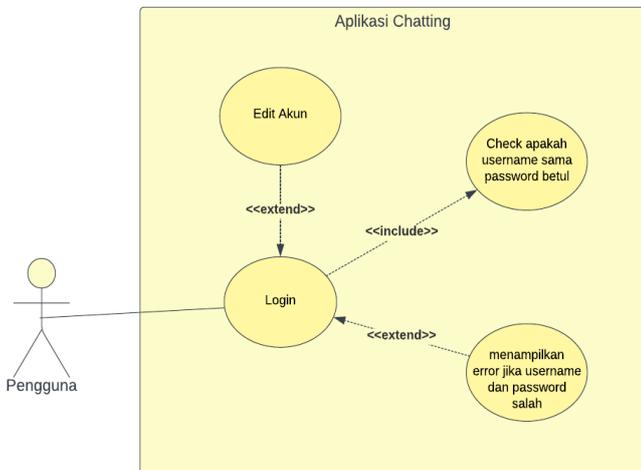
halaman *register* melewati halaman *login* jika ternyata pengguna belum mendaftarkan diri pada aplikasi ini.



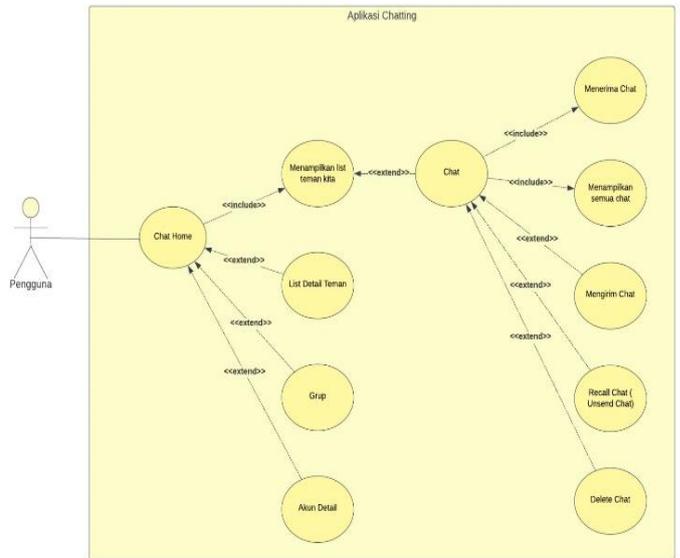
Gambar 4. Use Case Diagram Aplikasi Chatting



Gambar 5. Use Case Diagram pada Halaman Register

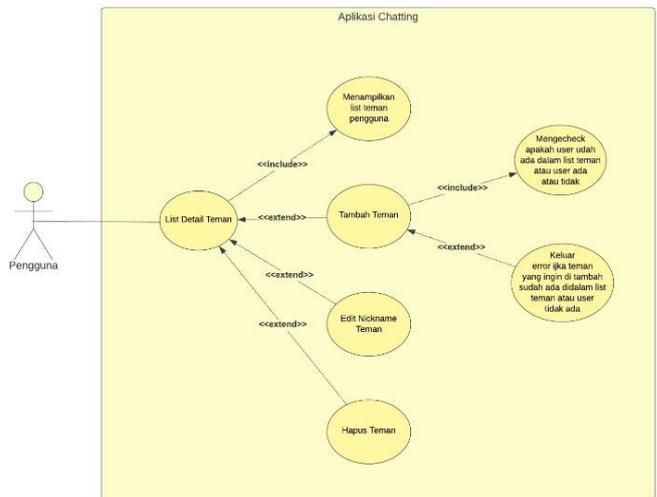


Gambar 6. Use Case Diagram pada Halaman Login



Gambar 7. Use Case Diagram pada Halaman Chat Home

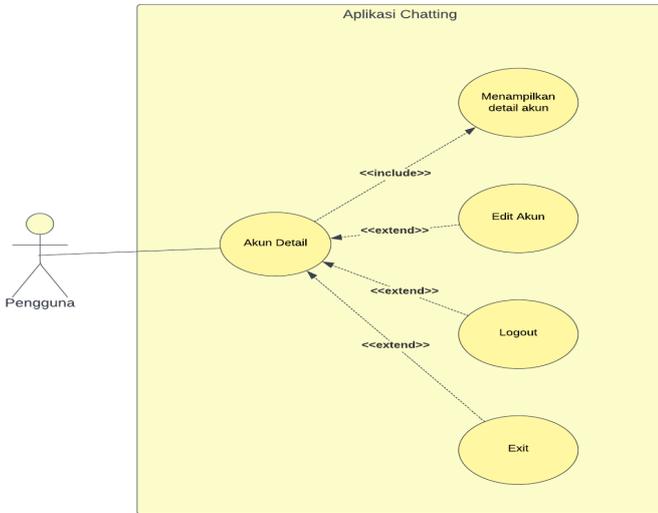
Gambar 7 memperlihatkan proses pada halaman *chat home*. Pengguna akan menuju ke halaman ini jika proses *login*-nya berhasil. *Chat home* akan menampilkan semua daftar teman pengguna, dan jika pengguna memilih salah satu temannya maka akan muncul *chat box* yang menampilkan semua *chat* yang terjadi di antara 2 pengguna, kemudian pengguna juga dapat mengirimkan pesan, menarik balik pesan yang sudah dikirim, dan juga hapus semua *chat*. Selain dari itu pengguna juga dapat menuju ke halaman *List detail teman*, *Grup*, dan *Akun Detail* melewati halaman *Chat Home*.



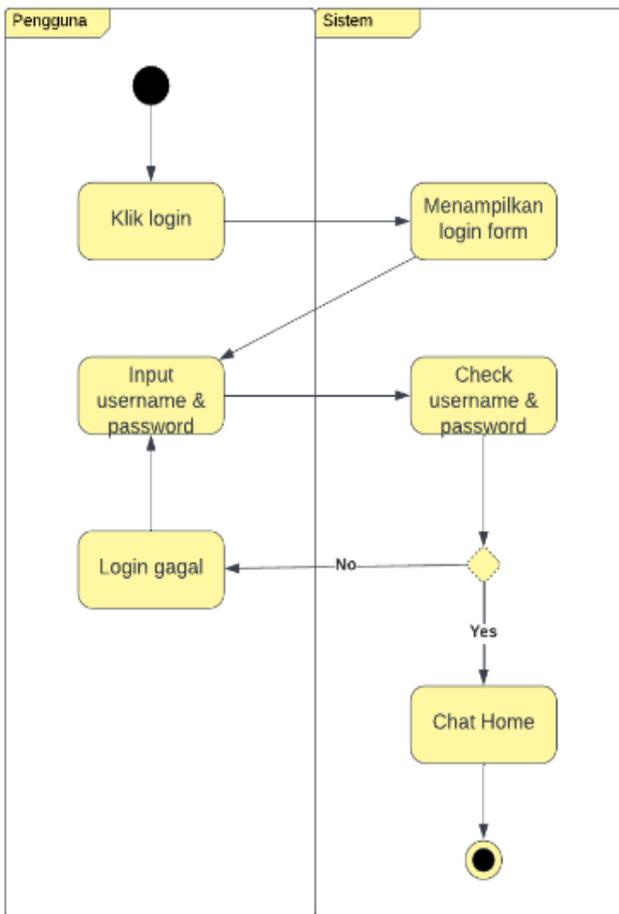
Gambar 8. Use Case Diagram pada Halaman List Detail Teman

Gambar 8 memperlihatkan proses pada halaman *List Detail Teman*. Pengguna dapat mengakses ke halaman ini melewati halaman *Chat Home*, Halaman ini akan menampilkan list teman pengguna secara detail seperti, ID, Nickname, dan Tentang Pengguna. Di halaman ini pengguna juga dapat menambahkan teman yang di mana pengguna akan melakukan input pada username dari pengguna yang akan ditambahkan, kemudian akan dicek apakah user ini ada, atau user ini sudah menjadi teman dari pengguna, jika user tidak ada atau user

sudah menjadi teman maka akan keluar error untuk pengguna. Selain dari itu Pengguna juga dapat mengubah nickname dari teman dan menghapus teman.



Gambar 9. Use Case Diagram pada Halaman Akun Detail



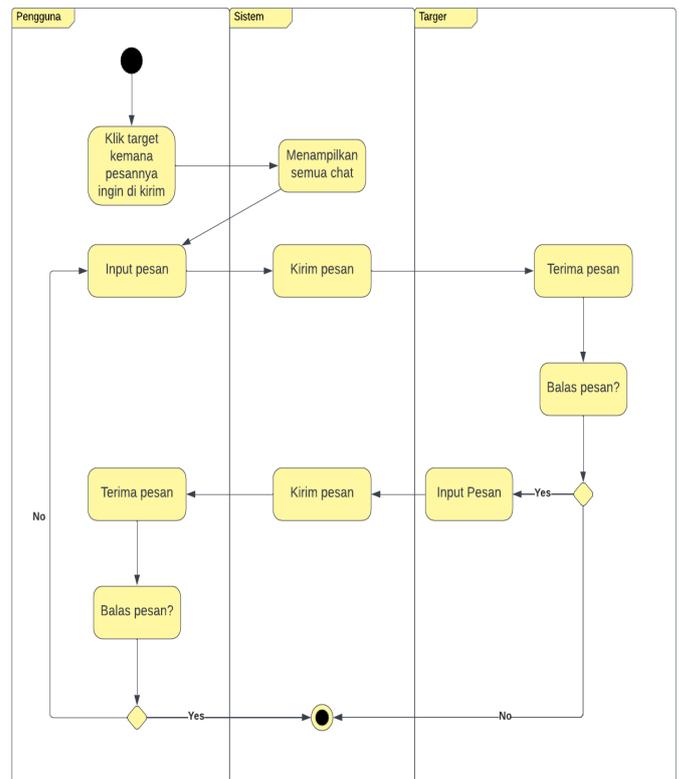
Gambar 10. Activity Diagram Login

Gambar 9 memperlihatkan proses pada halaman Akun Detail. Halaman ini akan menampilkan detail-detail dari pengguna, pengguna juga dapat melakukan edit pada detail

akunnya seperti, *Username, Password, Email, Nomor Telepon, dan Tentang Pengguna*. Selain dari itu pengguna dapat melakukan *logout* atau *exit* pada halaman ini.

Activity Diagram aplikasi ini ditunjukkan pada Gambar 10. Proses dimulai dari login yang dilakukan oleh pengguna sebelum masuk aplikasi. Setelah sistem menampilkan halaman login pengguna diminta memasukkan *username* dan *password*. Jika login berhasil, sistem akan menampilkan halaman *Chat Home*, dan pengguna dapat memulai percakapan dengan pengguna lain yang terdaftar.

Gambar 11 memperlihatkan proses *chatting*. Pengguna memilih target ke mana pesannya ingin dikirim, sistem akan menampilkan semua *chat* antara kedua pengguna ini, pengguna meng-*input* pesannya, pesan dikirim oleh sistem, dan target akan menerima pesannya. Jika target ingin membalas pesan, maka target akan meng-*input* pesan, kemudian pesan akan dikirim oleh sistem, dan pengguna mendapatkan pesannya. Proses ini akan selalu berulang hingga titik di mana pengguna ingin mengakhiri *chatting* tersebut.

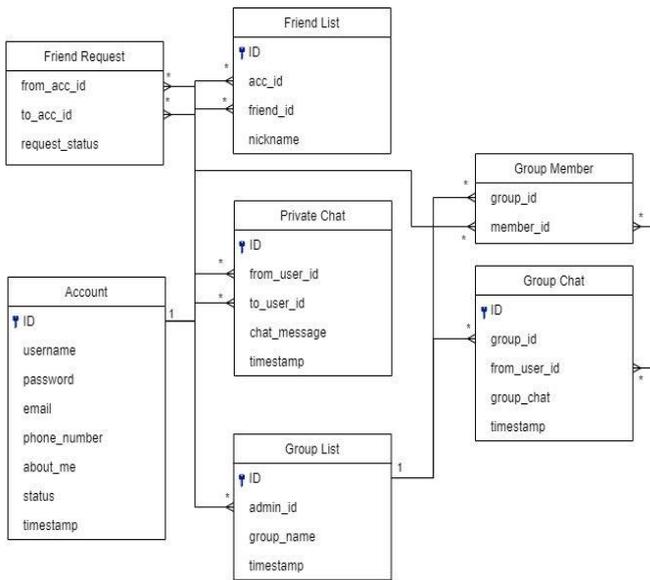


Gambar 11. Activity Diagram Proses Chatting

C. Perangkat lunak

Pada Tahap ini, kami melakukan proses pemrograman (*Coding*) dengan menggunakan bahasa pemrograman yaitu, *Python3*, serta membuat *database*. DBMS yang dipilih adalah *MySQL*, dan untuk mengoneksikan kodenya dengan *database*, kami menggunakan modul *PyMySQL*, rancangan table dalam *database* yang kami rancang ditunjukkan pada Gambar 12.

Kami juga menggunakan modul *Tkinter* untuk membuat *Grapical User Interface (GUI)* dari aplikasi *chatting* tersebut. Berikut merupakan hasil GUI dari aplikasi *chatting* yang telah dirancang.

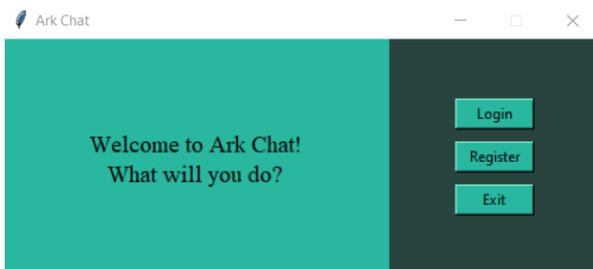


Gambar 12. Rancangan Tabel Database pada Aplikasi *Chatting*

Gambar 13 merupakan rancangan GUI untuk halaman utama pada saat aplikasi pertama kali diakses. *User* dapat memilih tiga menu yaitu Login untuk masuk ke dalam aplikasi jika telah memiliki akun. Jika belum memiliki akun, maka *user* dapat memilih menu Register. Tombol Exit digunakan untuk keluar dari aplikasi. Rancangan GUI untuk halaman Register ditunjukkan pada Gambar 14. Pada halaman ini *user* diminta untuk memasukkan data profil yang nantinya akan disimpan dalam database. Untuk login *user* perlu memasukkan data *username* dan *password* yang benar agar dapat masuk ke aplikasi. Rancangan GUI untuk login dapat dilihat pada Gambar 15.

Setelah *user* berhasil masuk, maka akan ditampilkan halaman *chat home* yang digunakan untuk melakukan percakapan (*chat*) seperti tampak pada Gambar 16. Pada sisi sebelah kiri ditampilkan daftar nama *friends* yang telah bergabung pada akun *user*. Pada sisi sebelah kanan merupakan tempat *history* dari percakapan yang telah dilakukan oleh *user* bersangkutan.

Pada Gambar 17 ditunjukkan rancangan halaman Account Detail yang menampilkan informasi tentang akun pengguna. Pada halaman ini disediakan fitur untuk mengganti *username*, *password*, *email*, *phone number* dan tentang pengguna. Pada Gambar 18 ditunjukkan desain halaman *Friend List* yang menampilkan daftar teman, sedang pada Gambar 19 merupakan desain halaman Group Chat.



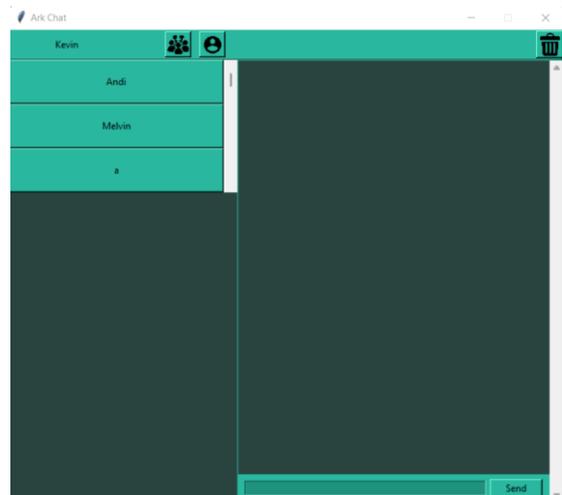
Gambar 13. Rancangan GUI *Main*



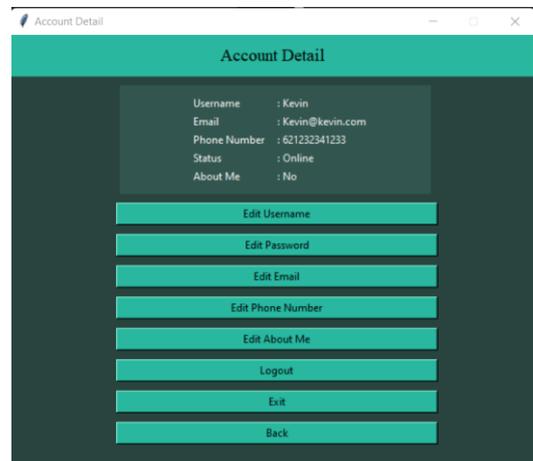
Gambar 14. Rancangan GUI *Register Form*



Gambar 15. Gambar IV.3.3 Gambar GUI *Login Form*



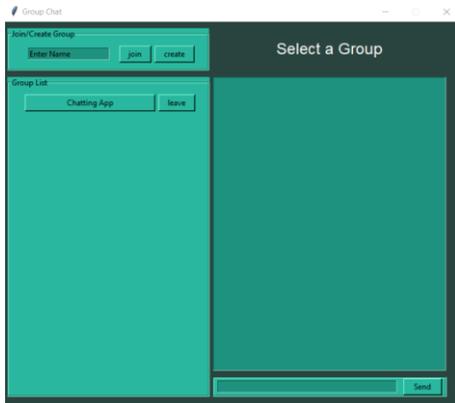
Gambar 16. Rancangan GUI *Chat Home*



Gambar 17. Rancangan GUI *Account Detail*



Gambar 18. Rancangan GUI Friend List



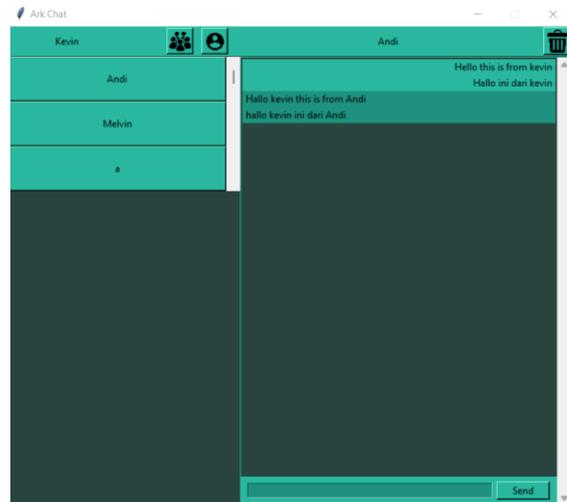
Gambar 19. Rancangan GUI Group Chat

Selain dari ini, kami juga mengimplementasikan fitur enkripsi pada data chat yang dikirim dari pengguna dengan menggunakan modul python yaitu *Cryptography* dengan metode Enkripsi Simetris, sehingga dapat meningkatkan keamanan & *privacy* bagi para pengguna. Enkripsi kami terapkan pada penyimpanan data percakapan (teks) di dalam database. Pada saat membaca kembali teks percakapan yang tersimpan pada database, diterapkan dekripsi dengan menggunakan kunci yang sama seperti enkripsi pada sisi aplikasi. Gambar 20 menunjukkan hasil enkripsi data percakapan yang tersimpan pada *Table private_chat field chat_message* di dalam database.

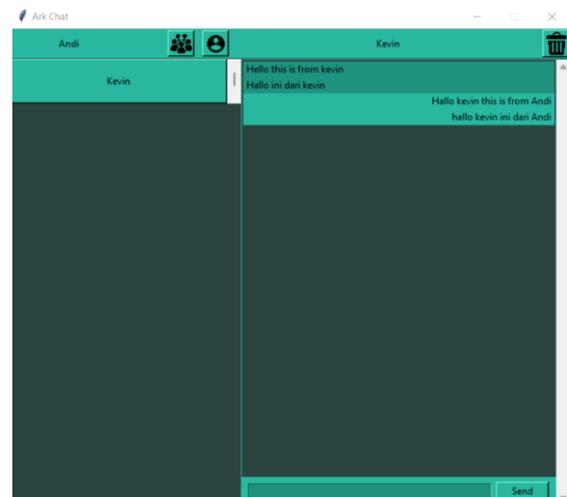
D. Pengujian Perangkat Lunak

Pada tahap ini, kami melakukan pengujian terhadap aplikasi *chatting* antar pengguna (Andi dan Kevin) dan dalam *group*. Tampak pada Gambar 21 dan Gambar 22 bahwa pada

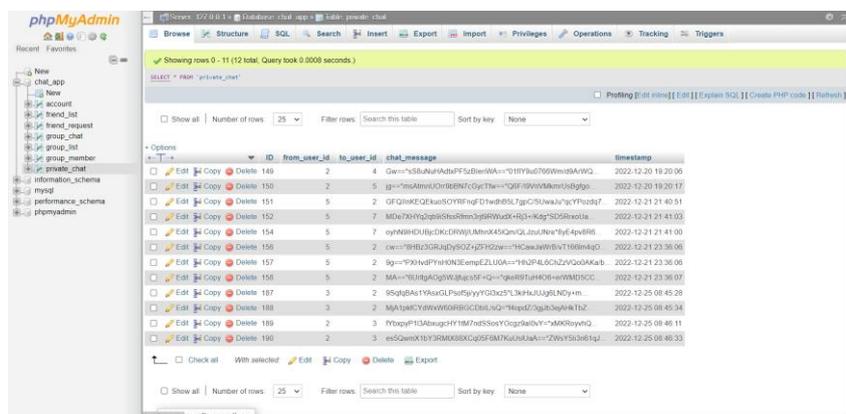
kedua sisi pengguna hasil percakapan yang berhasil dikirim dan diterima oleh masing-masing pengguna. Hasil percakapan dalam *group* ditampilkan pada Gambar 23, tampak bahwa percakapan dapat ditampilkan dengan format yang berbeda dengan percakapan pribadi.



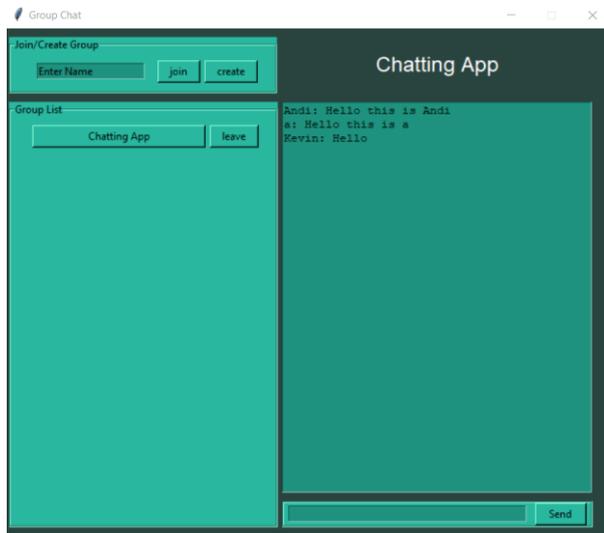
Gambar 21. Hasil percakapan dari Sisi Kevin



Gambar 22. Hasil Percakapan dari Sisi Andi



Gambar 20. Hasil Enkripsi Disimpan Pada Tabel Database



Gambar 21. Hasil Percakapan Menggunakan *Group Chat*

V. KESIMPULAN

Pada artikel ini telah dirancang aplikasi *Chatting* dengan metode *Software Development Life Cycle* (SDLC). Aplikasi dapat mengirimkan dan menerima pesan teks dengan baik. Untuk pengembangan selanjutnya perlu ditambahkan fitur untuk dapat mengirimkan pesan berupa gambar, pesan suara ataupun video. Pembuatan *key* untuk keperluan enkripsi juga dapat digenerasi menggunakan algoritma tertentu sehingga tingkat keamanan komunikasi data dapat lebih baik.

VI. DAFTAR PUSTAKA

- [1] Ali, A. H., & Sagheer, A. M. (2017). Design of Secure Chatting Application with End to End Encryption for Android Platform. *Iraqi Journal for Computers and Informatics*, 43(1), 22–27. <https://doi.org/10.25195/2017/4315>
- [2] Bardis, N. G. (2008). Design of a secure chat application based on AES cryptographic algorithm and key management. *Proceedings of the 10th WSEAS International Conference on Mathematical Methods, Computational Techniques and Intelligent Systems*, 486–491. <https://www.researchgate.net/publication/234793505>
- [3] Chandra, S., Bhattacharyya, S., Paira, S., & Alam, S. S. (2014). A study and analysis on symmetric cryptography. *2014 International Conference on Science Engineering and Management Research, ICSEMR 2014*. <https://doi.org/10.1109/ICSEMR.2014.7043664>
- [4] Gurung, G., Shah, R., & Jaiswal, D. P. (2020). Software Development Life Cycle Models-A Comparative Study. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 30–37. <https://doi.org/10.32628/CSEIT206410>
- [5] Hendini, A. (2016). PEMODELAN UML SISTEM INFORMASI MONITORING PENJUALAN DAN STOK BARANG (STUDI KASUS: DISTRO ZHEZHA PONTIANAK). *JURNAL KHATULISTIWA INFORMATIKA, VOL. IV, N.*
- [6] Santoso, P. P., Rilvani, E., Trisnawan, A. B., Adiyarta, K., Napitupulu, D., Sutabri, T., & Rahim, R. (2018). Systematic literature review: Comparison study of symmetric key and asymmetric key algorithm. *IOP Conference Series: Materials*

Science and Engineering, 420(1). <https://doi.org/10.1088/1757-899X/420/1/012111>

- [7] Vashi*, D., Bhadka, Dr. H. B., Patel, Dr. K., & Garg, Dr. S. (2019). Performance of Symmetric and Asymmetric Encryption Techniques for Attribute Based Encryption. *International Journal of Recent Technology and Engineering (IJRTE)*, 8(4), 176–182. <https://doi.org/10.35940/IJRTE.C6597.118419>
- [8] Yulianto, B., Heriyanni, E., Dewi, L. C., & Adinugroho, T. Y. (2015). Architecture and Implementation of Instant Messaging in Educational Institution. *Procedia Computer Science*, 59, 5–13. <https://doi.org/10.1016/j.procs.2015.07.331>