

## IMPLEMENTASI *UNIFIED MODELING LANGUAGE* (UML) PADA PERANCANGAN APLIKASI *WIFITALKIE* BERBASIS TCP/IP

Afrig Aminuddin

Sistem Informasi, Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta  
Jl. Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta 55283 INDONESIA

Email: [afriq@amikom.ac.id](mailto:afriq@amikom.ac.id)

(Diterima: 8 April 2019, direvisi: 15 Mei 2019, disetujui: 15 Mei 2019)

### ABSTRAK

Di dunia komunikasi analog kita mengenal perangkat yang bernama HT (*Handy Talkie*). Perangkat ini bekerja dengan menggunakan sinyal elektromagnetik pada frekuensi radio tertentu. Perangkat ini berfungsi sebagai pengirim dan penerima sinyal radio. Sinyal yang dikirimkan adalah sinyal suara yang telah diubah menjadi sinyal elektromagnetik. Untuk dapat berkomunikasi satu sama lain, maka harus ada kesepakatan antar pengguna untuk menyetel perangkatnya pada frekuensi yang sama. Salah satu kelemahan dari perangkat ini adalah kualitas suara yang cenderung *noisy* dan sangat bergantung pada kondisi cuaca. Dengan seiring teknologi semikonduktor yang berkembang pesat, terciptalah perangkat digital yang semakin bervariasi kegunaannya. Saat ini sudah banyak diciptakan perangkat digital yang dapat menggantikan perangkat analog secara keseluruhan. Sebagai contohnya adalah pesawat televisi. Pesawat televisi digital memberikan kualitas yang jauh lebih baik daripada perangkat televisi analog dengan ukuran yang jauh lebih ramping. Contoh yang lain saat ini tersedia *smartphone* yang memiliki fitur yang sangat lengkap yang tertanam pada perangkat yang berukuran relatif kecil. Salah satu fiturnya adalah *wifi*. Dengan fitur ini sebuah *smartphone* dapat terhubung satu sama lain, bahkan dapat terhubung dengan internet dengan mudahnya. Dalam rangka digitalisasi perangkat analog dan tersedianya fitur *wifi* di dalam *smartphone* ini, maka diciptakan sebuah aplikasi *WiFiTalkie*. Cara kerjanya yaitu *smartphone* akan mengirimkan sinyal suara dengan memanfaatkan teknologi TCP/IP melalui jaringan *wifi*, kemudian *smartphone* yang lain di dalam *network* yang sama menerima sinyal ini dan memprosesnya kembali menjadi sinyal suara. Aplikasi ini dibangun dengan mengimplementasikan metode perancangan *Unified Modeling Language* (UML). Hasil penelitian ini menunjukkan bahwa kualitas suara yang dihasilkan oleh *WiFiTalkie* jauh lebih baik daripada HT yang berbasis pada sinyal analog.

**Kata Kunci:** *WiFiTalkie*, analog, digital, internet, UML.

### 1 PENDAHULUAN

Selain kualitas suara yang cenderung *noisy*, teknologi HT juga memiliki keterbatasan *half-duplex* yaitu setiap perangkat HT yang diatur pada frekuensi yang sama hanya ada satu perangkat yang dapat mengirimkan suara pada waktu yang sama. Jika suatu saat terjadi *jamming* maka frekuensi tersebut tidak dapat digunakan di wilayah tersebut. *Jamming* terjadi jika salah satu pengguna HT pada frekuensi tertentu dengan sengaja menekan tombol *transmit* tanpa melepaskannya. Hal ini mengakibatkan semua perangkat HT lain hanya bisa mendengarkan dan tidak bisa mengirimkan suara. Sebaliknya teknologi yang digunakan pada perangkat digital berbasis pada TCP/IP. Teknologi ini memungkinkan komunikasi data berjalan secara *full-duplex*. Sehingga setiap perangkat dapat mengirimkan dan menerima data pada saat bersamaan. Sehingga tidak akan terjadi *jamming* pada komunikasi perangkat digital. Sinyal frekuensi analog bersifat terbuka atau tidak terenkripsi artinya semua orang dapat mendengarkan percakapan HT asalkan dia mengatur perangkat HT miliknya pada frekuensi yang sama. Oleh karena itu kita sering mendengar istilah-istilah angka di dalam percakapan HT seperti “86”. Hal ini bertujuan agar orang awam selain anggota komunitas tidak mengetahui arti dari angka-angka tersebut. Pada komunikasi data digital telah ditemukan sistem enkripsi yang sangat kuat seperti RSA. Sehingga tidak ada yang bisa membaca data saat dikirimkan kecuali pengirim dan penerima data. Tentunya hal ini sangat berbeda dengan komunikasi perangkat analog. Berdasarkan

beberapa permasalahan tersebut di atas maka pada penelitian ini akan diciptakan sebuah aplikasi *smartphone* yang diharapkan dapat digunakan untuk media komunikasi sebagaimana HT bekerja.

## 2 TINJAUAN PUSTAKA

Sebelum aplikasi diciptakan, terlebih dahulu dibutuhkan beberapa komponen utama agar aplikasi dapat berjalan sebagaimana mestinya. Komponen pertama aplikasi ini adalah sebuah *audio codec* yang bertanggung jawab melakukan kompresi terhadap data suara. Pada penelitian sebelumnya yang dilakukan oleh penulis. Penulis telah menemukan *audio codec* yang tepat untuk digunakan pada komunikasi suara digital [1]. *Codec* yang dimaksud adalah Opus [2][3]. Dengan *audio codec* ini, maka diharapkan aplikasi dapat meminimalisir *latency* yang terjadi dengan cara kompresi optimal terhadap data suara. Komponen yang kedua adalah sebuah protokol atau arsitektur yang memungkinkan komunikasi antara beberapa perangkat *smartphone* dapat berjalan sebagaimana mestinya. Sebelumnya penulis juga telah meneliti sebuah desain protokol yang sesuai untuk aplikasi WiFiTalkie ini [4]. Protokol tersebut adalah protokol *hybrid*. Protokol ini didesain khusus untuk komunikasi antar perangkat *smartphone* secara berkesinambungan.

Setelah dua komponen utama ditemukan, maka selanjutnya akan dibahas mengenai beberapa metode yang tepat digunakan untuk perancangan aplikasi WiFiTalkie. UML atau *Unified Modeling Language* adalah suatu metode yang digunakan untuk perancangan sistem pada pengembangan aplikasi berbasis *Object Oriented Programming* (OOP) [5]. UML bertujuan untuk menganalisa proses bisnis dan kemudian memodelkannya kedalam notasi – notasi tertentu [6]. Notasi – notasi ini bermaksud untuk memvisualisasikan setiap *entity* pada OOP dan relasi antar setiap *entity*. UML dapat menggambarkan sistem secara dinamis maupun statis. Untuk gambaran dinamis sistem menggunakan *use case diagram*, *activity diagram*, dan *sequence diagram*. Sedangkan untuk gambaran statis sistem menggunakan *class diagram* [7]. *Use case diagram* memvisualisasikan interaksi satu aktor atau lebih pada sistem, baik yang dilakukan aktor terhadap sistem maupun yang dilakukan sistem terhadap aktor [8]. Selain itu diagram ini juga digunakan untuk mengetahui semua fungsi yang terdapat di dalam sistem dan menjelaskan tentang siapa saja yang berhak menggunakan fungsi-fungsi tersebut [9]. *Activity Diagram* adalah visualisasi dari *workflow* kegiatan atau tindakan yang dilakukan oleh aktor dan sistem secara iterasi dan konkurensi [10]. Dalam UML, *activity diagram* dapat digunakan untuk menggambarkan bisnis dan operasional langkah demi langkah alur kerja komponen dalam sistem. Sebuah *activity diagram* dapat menunjukkan aliran kontrol keseluruhan system [11]. *Class diagram* memvisualisasikan struktur kelas-kelas dari suatu sistem dan merupakan tipe diagram yang paling banyak dipakai. *Class diagram* memperlihatkan hubungan antar kelas dan penjelasan detail tiap-tiap kelas di dalam suatu sistem [12]. Selama proses analisis, *class diagram* memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem. Selama tahap desain, *class diagram* berperan dalam menangkap struktur dari semua kelas yang membentuk arsitektur sistem yang dibuat [13]. *Sequence diagram* ini menggambarkan interaksi antara objek di dalam dan di sekitar sistem (termasuk pengguna, *display*, dan sebagainya) berupa *message* yang digambarkan terhadap waktu. Diagram ini biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respon dari sebuah *event* untuk menghasilkan *output* tertentu [14]. Diawali dari sebuah aktivitas tertentu, kemudian berproses mengikuti urutan tertentu, yang bisa terlihat melalui *message* antar objeknya [15].

## 3 METODE PENELITIAN

### 3.1 Pengumpulan Data

Dalam pengumpulan data, metode yang digunakan adalah metode observasi. Observasi yang dilakukan adalah dengan melihat bagaimana perangkat *Handy Talkie* bekerja. Selain itu penulis juga mengumpulkan referensi baik dari buku, internet maupun sumber-sumber lainnya mengenai teknologi TCP/IP, bahasa pemrograman Java untuk aplikasi Android, *SQLite* untuk database sebagai tempat penyimpanan data, serta acuan untuk analisa perancangan sistem.

### 3.2 Analisis Sistem

Analisis sistem adalah teknik pemecahan masalah yang menguraikan bagian-bagian komponen dengan mempelajari seberapa baik bagian-bagian komponen tersebut bekerja dan berinteraksi untuk mencapai tujuan sistem. Analisis sistem merupakan tahap paling awal dalam pengembangan sistem yang menjadi pondasi menentukan keberhasilan sistem tersebut. Penulis menggunakan metode analisis *SWOT* (*Strengths, Weakness, Opportunity, Threats*) pada studi kasus ini. Analisis *SWOT* memiliki peranan guna menganalisis berbagai permasalahan yang bersinggungan dengan aplikasi yang dibuat. Berikut analisis *SWOT* terhadap aplikasi *WiFiTalkie* ini.

Kekuatan sistem lama yaitu HT memiliki daya jangkau sinyal radio yang cukup luas. Selain itu HT dapat menjangkau ribuan *device* sekaligus. Sedangkan kekuatan sistem baru adalah aplikasi dapat mengirimkan pesan berupa text, suara, maupun *file* [16]. Aplikasi juga dapat menampilkan informasi dari semua *device* yang terhubung.

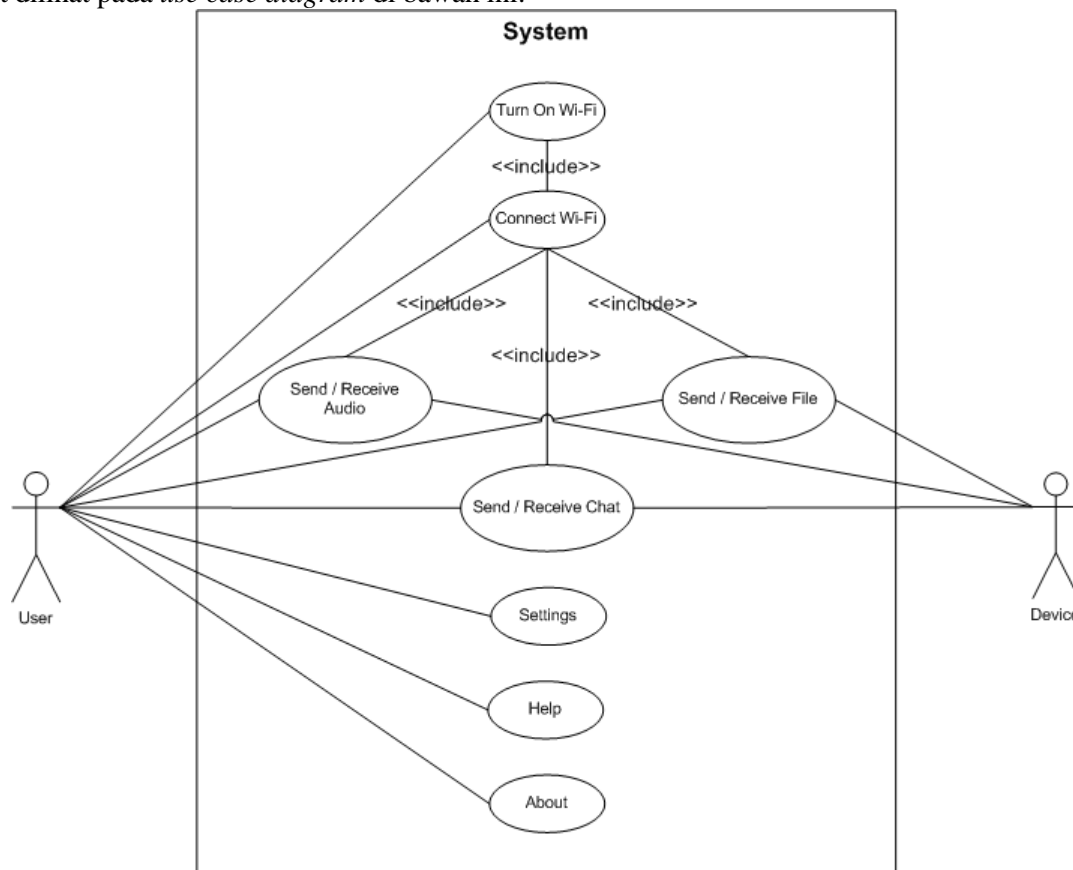
### 3.3 Perancangan UML

Perancangan aplikasi *WiFiTalkie* ini disesuaikan dengan kebutuhan di lapangan, yaitu kemudahan dalam menggunakan aplikasi ini. Sehingga maksud dan tujuan dari pembuatan aplikasi ini dapat terpenuhi.

Dalam perancangan aplikasi ini digunakan metode UML yang terdiri dari *use case diagram*, *activity diagram*, *sequence diagram* dan *class diagram*.

#### 3.3.1 Use Case Diagram

Aplikasi *WiFiTalkie* ini bekerja dengan cara *smartphone* akan mengirimkan sinyal suara dengan memanfaatkan teknologi TCP/IP melalui jaringan *wifi*, kemudian *smartphone* yang lain di dalam *network* yang sama menerima sinyal suara ini dan memrosesnya lebih lanjut. Di dalam aplikasi ini terdapat beberapa fitur yaitu pengiriman dan penerimaan audio, chat dan *file*. Untuk lebih jelasnya dapat dilihat pada *use case diagram* di bawah ini.



Gambar 1. Use Case diagram WiFiTalkie

Gambar 1 di atas merupakan *use casediagram* aplikasi WiFiTalkie. Pada diagram tersebut dijelaskan bahwa seorang aktor dapat menghidupkan *wifi* pada perangkat *smartphone* miliknya kemudian aktor tersebut dapat menghubungkan perangkatnya ke dalam jaringan *wifi*. Fitur utama pada aplikasi ini adalah *Audio Conference* untuk beberapa pengguna yang terhubung ke dalam jaringan *wifi* yang sama. Adapun fitur chat dan fitur *file* adalah fitur pelengkap yang disematkan ke dalam aplikasi ini. Untuk lebih jelasnya dapat dilihat pada skenario *use case* pada Tabel 1 di bawah ini. Interaksi antara aktor *user* dengan *use case audio* dijelaskan dalam skenario *use case* sebagai berikut:

**Tabel 1. Skenario Use Case Audio**

<b>Identifikasi</b>	
Nama	Audio
Tujuan	Agar user dapat berkomunikasi suara dengan semua <i>device</i> yang terkoneksi
Deskripsi	<i>User</i> melakukan input dan mendengarkan suara dari <i>device</i> yang terkoneksi
Aktor	<i>User</i>
<b>Skenario</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<i>User</i> memilih menu Talkies kemudian <i>user</i> melakukan input suara	Sistem akan menampilkan semua <i>device</i> yang terkoneksi kemudian sistem akan mengirim / menerima sinyal suara dari semua <i>device</i> yang terkoneksi

Interaksi antara aktor *user* dengan *use case chat* dijelaskan dalam skenario *use case* sebagaimana Tabel 2 berikut:

**Tabel 2. Skenario Use Case Chat**

<b>Identifikasi</b>	
Nama	Chat
Tujuan	Agar user dapat mengirim dan menerima <i>chat</i> dengan semua <i>device</i> yang terkoneksi
Deskripsi	<i>User</i> melihat <i>history chat</i> dan <i>user</i> dapat mengirim teks melalui <i>form chat</i> ke semua <i>device</i> yang terkoneksi
Aktor	<i>User</i>
<b>Skenario</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<i>User</i> memilih menu Chat kemudian menulis <i>chat</i>	Sistem menampilkan <i>history chat</i> kemudian sistem akan mengirimkan teks yang telah ditulis ke semua <i>device</i> yang terkoneksi

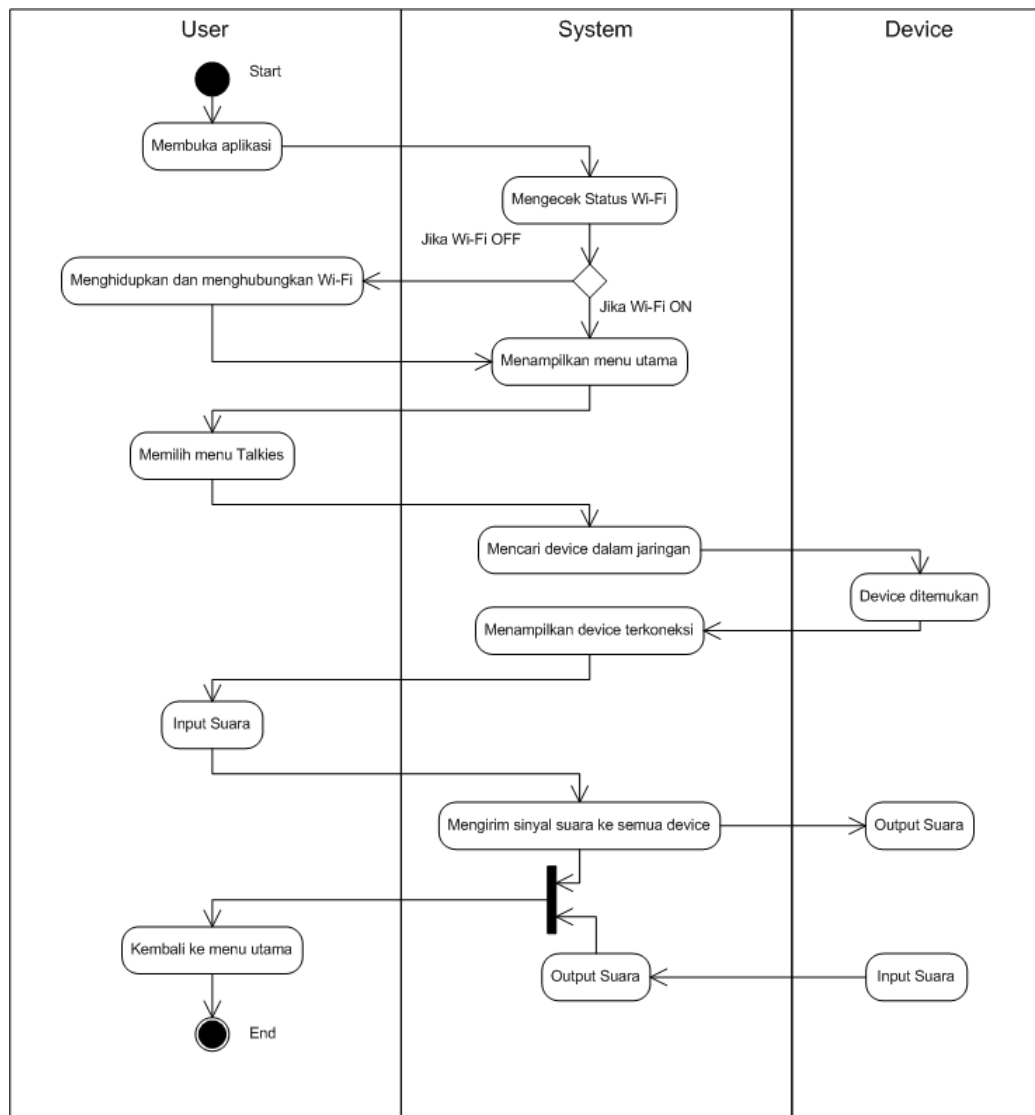
Interaksi antara aktor *user* dengan *use case files* dijelaskan dalam skenario *use case* sebagaimana Tabel 3 berikut:

**Tabel 3. Skenario Use Case Files**

<b>Identifikasi</b>	
Nama	Files
Tujuan	Agar user dapat mengirim dan menerima <i>file</i> dengan semua <i>device</i> yang terkoneksi
Deskripsi	<i>User</i> melihat <i>history</i> pengiriman <i>file</i> dan <i>user</i> dapat mengirim <i>file</i> ke semua <i>device</i> yang terkoneksi
Aktor	<i>User</i>
<b>Skenario</b>	
<b>Aksi Aktor</b>	<b>Reaksi Sistem</b>
<i>User</i> memilih menu Files kemudian <i>user</i> memilih <i>file</i> yang akan dikirim	Sistem menampilkan <i>history file</i> kemudian sistem akan mengirimkan <i>file</i> yang telah dipilih ke semua <i>device</i> yang terkoneksi

### 3.3.2 Activity Diagram

Activity diagram menggambarkan setiap skenario *use case* yang telah dibuat pada *use case diagram*. Activity diagram *audio* melibatkan sistem dan dua aktor, yaitu *user* dan *device*.



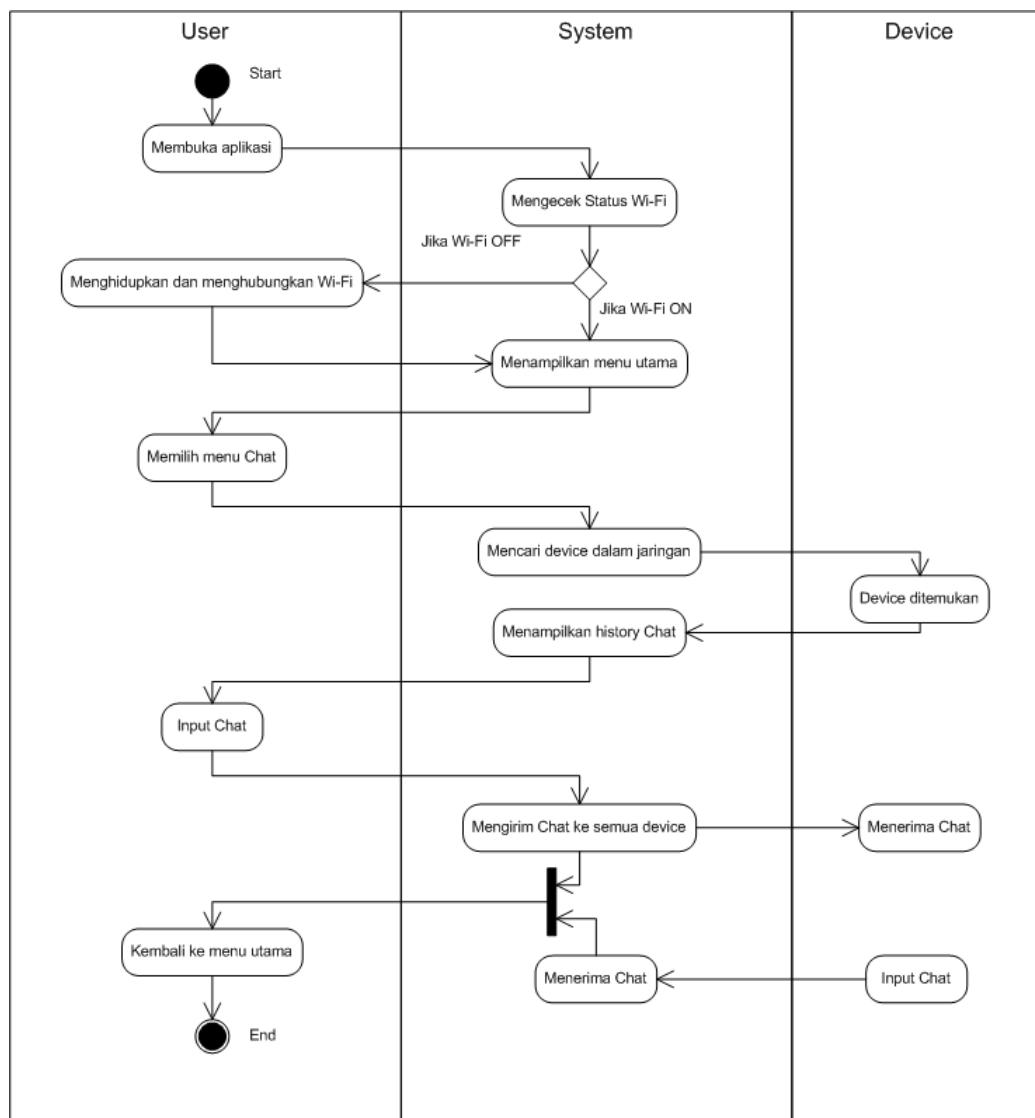
Gambar 2. Activity diagram audio

Berdasarkan activity diagram pada Gambar 2 di atas, aktifitas bermula dari aktor *user* membuka aplikasi. Kemudian sistem akan mengecek status *wifi*. Apakah *wifi* dalam keadaan hidup atau mati. Jika *wifi* dalam keadaan mati, maka *user* diminta untuk menghidupkan dan menghubungkan *wifi* ke jaringan. Kemudian sistem akan menampilkan menu utama. Sebaliknya jika *wifi* dalam keadaan hidup maka sistem langsung menampilkan menu utama.

Langkah selanjutnya adalah *user* diminta memilih menu *Talkies*. Kemudian sistem akan mencari semua *device* di dalam jaringan yang terinstall aplikasi *WiFiTalkie*. Jika *device* telah ditemukan maka sistem menampilkan semua *device* yang terkoneksi dengan jaringan *wifi* tersebut.

Sampai di titik ini *user* dapat mulai melakukan input berupa suara. Kemudian sistem akan mengkonversi suara tersebut ke dalam bentuk digital kemudian mengirimkannya ke semua *device* yang terkoneksi dengan jaringan. Pada *device* tujuan, data digital tersebut akan diolah kembali menjadi suara dan akan diperdengarkan kepada pengguna pada semua *device* tujuan. Di saat yang sama, *device* tujuan juga dapat memberikan umpan balik berupa suara kepada *user* pengirim dengan proses yang hampir sama seperti di saat *user* mengirimkan suara pertama kalinya. Setelah *user* selesai berkomunikasi suara, *user* dapat kembali ke menu utama dan aktifitas diakhiri.

Activity diagram Chat juga melibatkan sistem dan dua aktor, yaitu *user* dan *device*.



**Gambar 3. Activity diagram chat**

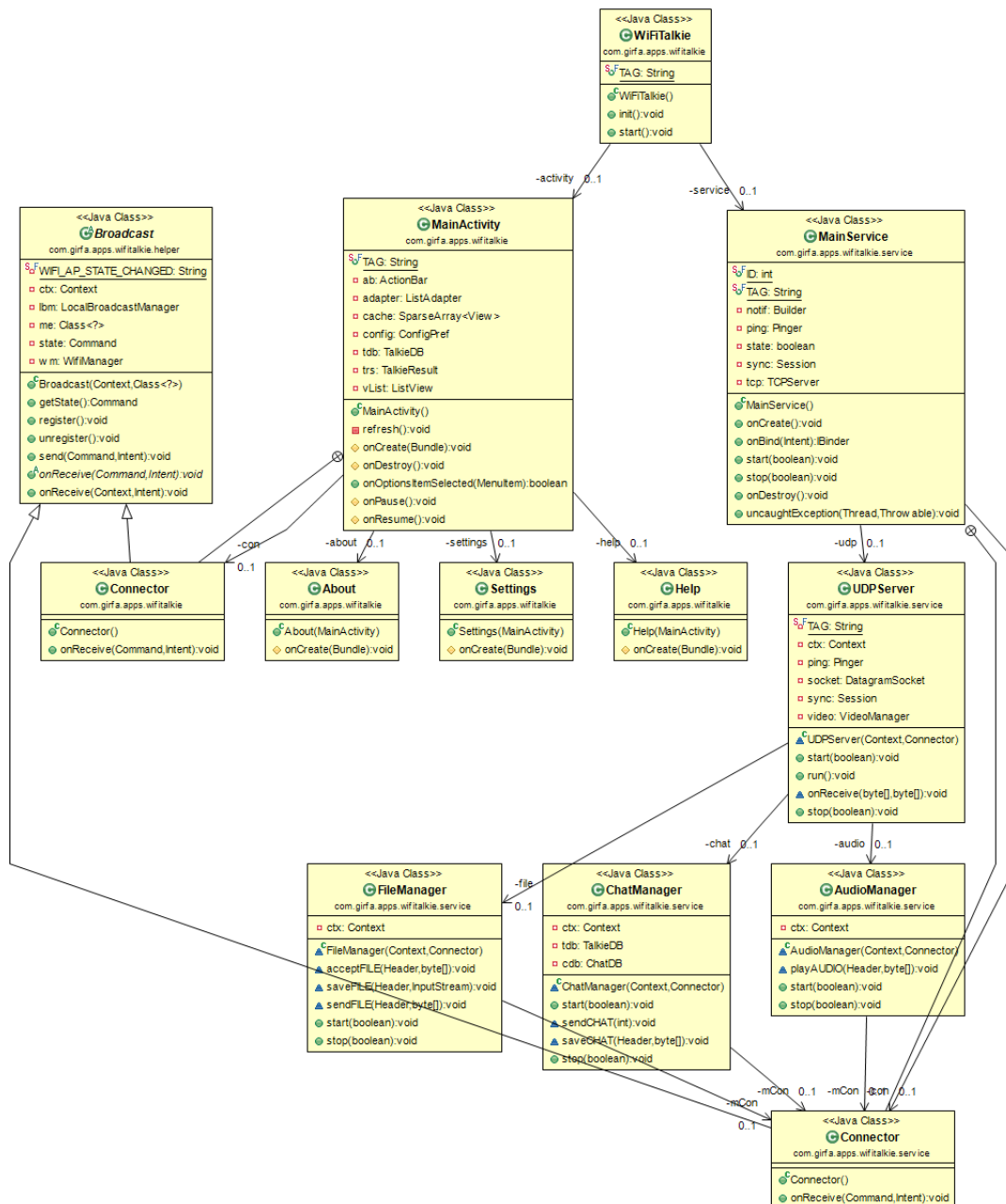
Berdasarkan Gambar 3 di atas, aktifitas dimulai dengan aktor *user* membuka aplikasi. Kemudian sistem akan mengecek status *wifi*. Apakah *wifi* dalam keadaan hidup atau mati. Jika *wifi* dalam keadaan hidup maka sistem langsung menampilkan menu utama. Sebaliknya jika *wifi* dalam keadaan mati, maka *user* diminta untuk menghidupkan dan menghubungkan *wifi* ke jaringan. Kemudian sistem akan menampilkan menu utama.

Langkah selanjutnya *user* diminta untuk memilih menu Chat. Kemudian sistem akan mencari semua *device* di dalam jaringan dan sistem akan menampilkan *historychat* yang pernah dilakukan oleh semua *device* di dalam jaringan tersebut.

Selanjutnya *user* melakukan input berupa *chat* ke dalam aplikasi. Sistem akan mengirimkan *chat* tersebut ke semua *device* di dalam jaringan. Di sisi lain *device* penerima akan menampilkan *chat* tersebut ke dalam *historychat*. Selain itu di saat yang sama, semua *device* di dalam jaringan berkesempatan untuk saling mengirimkan *chat* satu sama lain. Usai *user* saling berkomunikasi melalui *chat*, *user* akan kembali ke menu utama dan aktifitas diakhiri.

### 3.3.3 Class Diagram

Setelah *use case diagram* dan *activity diagram* selesai didesain, langkah selanjutnya adalah mendesain *class diagram*. *Class diagram* adalah diagram yang menjelaskan tentang keterikatan antar objek di dalam sebuah sistem. Pada aplikasi ini terdiri dari 13 *class* utama. Yang pertama adalah *class WiFiTalkie*. *Class* ini bertugas untuk menampilkan halaman pertama saat aplikasi di buka yang biasa disebut dengan *splash screen*. Selanjutnya dari *class* ini akan memulai *MainActivity* dan *MainService*. *MainActivity* menyediakan tampilan menu utama aplikasi. Sedangkan *MainService* berjalan di *background* untuk memproses data yang dikirim dan diterima oleh sistem dan device lain. *MainService* akan tetap berjalan meskipun pengguna sedang menggunakan aplikasi lain.

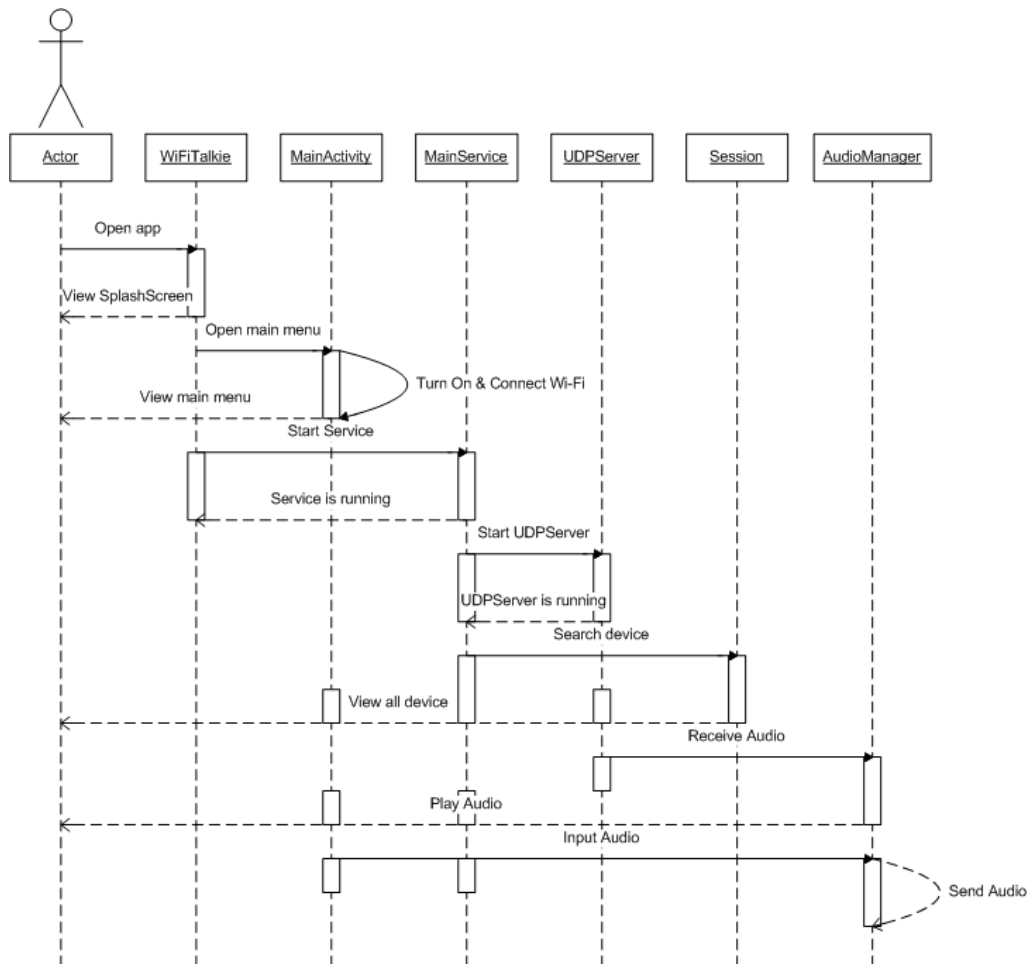


Gambar 4. Class diagram WiFiTalkie

Pada Gambar 4 di atas memperlihatkan hubungan antar kelas pada aplikasi WiFiTalkie. *MainService* yang berjalan di *background* membuat sebuah server dengan protocol UDP. Server ini melayani tiga fitur utama aplikasi yaitu *AudioManager*, *ChatManager* dan *FileManager*. Setiap *Manager* terhubung dengan *class MainService* melalui sebuah *class Connector*.

### 3.3.4 Sequence Diagram

Langkah selanjutnya adalah mendesain *sequence diagram*. *Sequence diagram* menjelaskan aksi aktor terhadap sistem yang digambarkan secara rinci pada setiap *class* yang terdapat pada sistem. Pada Gambar 5 dapat dijelaskan bahwa ketika pertama kali aktor membuka aplikasi, *class WiFiTalkie* membuka *class* utama yaitu *class MainActivity*. Di saat yang bersamaan, *class MainActivity* akan mengaktifkan *wifi* pada perangkat Android melalui sebuah fungsi yang terdapat pada *class MainService*. Ketika *wifi* sudah aktif, *MainService* memberikan respon kepada *WiFiTalkie*. Selanjutnya *WiFiTalkie* menjalankan *MainService* untuk selalu siap sedia di *background*. Sesaat kemudian, *MainService* memberikan respon terhadap *WiFiTalkie* bahwa *MainService* berhasil berjalan di *background* dan siap menerima perintah selanjutnya.



**Gambar 5. Sequence diagram Audio**

Untuk dapat menerima data berupa *audio* dari *device* lain, maka *MainService* menjalankan *UDPServer*. Pada kasus ini setiap *device* akan menjalankan *UDPServer*. Setelah *UDPServer* berhasil berjalan, ia akan memberi tahu kepada *MainService* bahwa server telah berjalan. Masing-masing *device* yang terhubung akan dibuatkan *Session*. Ketika *device* lain mengirimkan data berupa suara, *Session* akan mengirimkan suara kepada aktor melalui *MainService* dan *MainActivity*. Sedangkan *AudioManager* bertugas untuk melakukan *decode* terhadap data suara yang akan dikirim maupun yang diterima. Setiap detail diatas akan berjalan berulang dalam waktu yang sangat singkat.

### 3.4 Perancangan Antar Muka

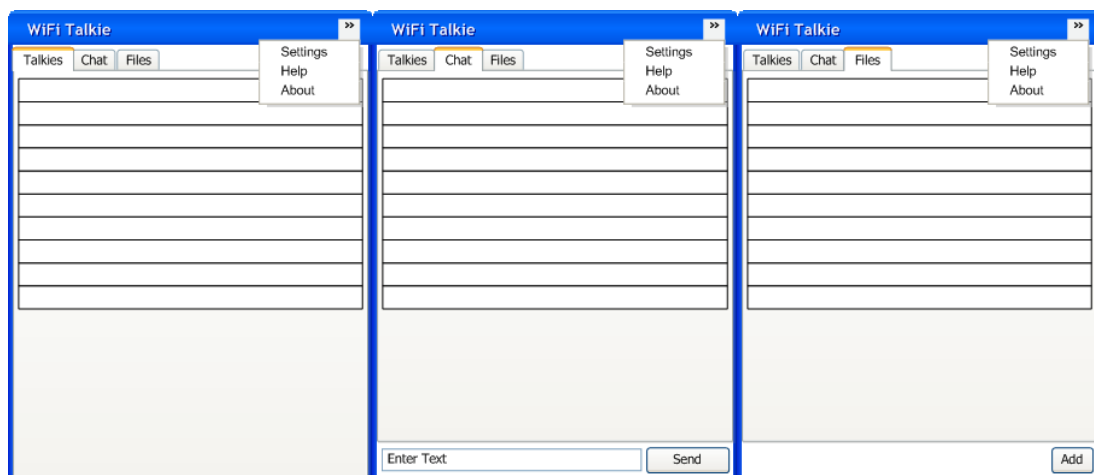
Di dalam membangun sebuah aplikasi akan melewati tahap perancangan antarmuka. Dengan perancangan ini diharapkan tata letak aplikasi dapat disesuaikan dengan kebutuhan pengguna. Berikut perancangan tata letak aplikasi *WiFiTalkie*.





**Gambar 6. Perancangan *Splash Screen* dan *Turn On Wifi***

Pada Gambar 6 merupakan perancangan halaman depan atau *Splash Screen* dan halaman untuk menyalakan koneksi *wifi*. Sedangkan perancangan menu utama aplikasi dapat dilihat pada Gambar 7.



**Gambar 7. Perancangan Menu Utama Aplikasi**

Terdapat tiga menu utama di dalam perancangan aplikasi ini yaitu menu Talkies yang digunakan untuk komunikasi VoIP, menu Chat yang digunakan untuk fitur *chatting*, dan menu Files yang digunakan untuk fitur pengiriman *file*.

## 4 HASIL DAN PEMBAHASAN

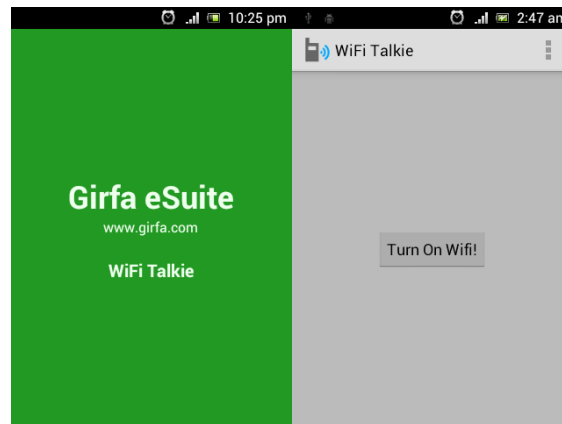
Tahap ini berisi tentang hasil implementasi dari analisis dan perancangan yang sudah dibahas bab sebelumnya. Implementasi dan pembahasan pembuatan aplikasi WiFi Talkie ini menggunakan *smartphone* Sony Xperia Z5 Compact dan Asus Zenfone Life.

### 4.1 Implementasi Basis Data

Di dalam aplikasi ini *database* dibuat *on the fly* oleh *framework* Android. *Framework* Android menggunakan SQLiteOpenHelper sebagai *class* utama yang bertugas menangani *database*. Di dalam *class* ini terdapat sebuah *method* yang hanya dijalankan sekali saja ketika aplikasi pertama kali dijalankan setelah aplikasi dipasang, *method* ini adalah onCreate(SQLiteDatabase db). Oleh karena itu di dalam *method* inilah *database* dibuat.

### 4.2 Implementasi Antarmuka

Antarmuka aplikasi ini mengikuti *guideline* yang telah ditentukan oleh Google berkaitan dengan pengembangan aplikasi Android.



**Gambar 8. Implementasi Splash Screen dan Turn On Wifi**

Pada Gambar 8 di atas adalah hasil implementasi dari perancangan *Splash Screen* dan halaman untuk menyalakan *wifi* yang terdapat pada *smartphone*. Kemudian Gambar 9 di bawah ini adalah implementasi menu utama aplikasi sesuai dengan perancangan sebelumnya.



**Gambar 9. Implementasi Menu Utama Aplikasi**

Pada implementasinya, setiap menu memiliki tambahan tombol pada setiap menu utama aplikasi seperti terlihat pada gambar di atas. Di bagian bawah terdapat tombol untuk memutus koneksi dengan server setiap saat pengguna menghendaki untuk berhenti menerima atau mengirim data.

### 4.3 Pembahasan Kode Program

Pada pembahasan kode program ini akan ditampilkan potongan kode program utama pada aplikasi WiFiTalkie ini. Kode yang dimaksud adalah *AndroidManifest.xml* seperti pada Gambar 10 di bawah ini.

```

7      <uses-sdk
8          android:minSdkVersion="9"
9          android:targetSdkVersion="19" />
10
11     <uses-permission android:name="android.permission.INTERNET" />
12     <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
13     <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
14     <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
15     <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
16     <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />

```

**Gambar 10. Android Manifest.xml**

Pada kode program di atas, aplikasi ini menggunakan beberapa *permission* untuk dapat berjalan. *Permission* yang wajib ada adalah koneksi internet dan *wifi* sebagai jalur utama komunikasi data.

## 5 KESIMPULAN

Telah dibangun sebuah aplikasi *WiFiTalkie* yang mampu bekerja sebagaimana HT bekerja. Aplikasi ini memberikan kualitas suara yang jauh lebih baik daripada perangkat HT. Hal ini terjadi karena aplikasi ini menggunakan data digital yang memberikan kualitas suara yang lebih baik daripada sinyal analog. Aplikasi ini dapat berjalan secara *full-duplex* sehingga tidak terjadi lagi *jamming* seperti yang terjadi pada perangkat HT. Aplikasi ini menawarkan keamanan yang jauh lebih baik daripada HT yang berbasis pada frekuensi terbuka. Aplikasi ini menggunakan enkripsi data digital sehingga data tidak dapat diakses oleh selain pengirim dan penerima data.

Akan tetapi penelitian ini masih memiliki celah untuk dikembangkan. Salah satunya adalah bagaimana memperluas jaringan komunikasi digital yang masih terbatas pada area *coverage wifi*.

## REFERENSI

- [1] A. Aminuddin, W. Widyawan, and R. Ferdiana, "Analisis Performa Audio Codec pada Implementasi Voice Over IP (VOIP)," *Semnasteknomedia Online*, vol. 4, no. 1, pp. 19–23, 2016.
- [2] J.-M. Valin, K. Vos, and T. Terriberry, "Definition of the Opus audio codec," 2012.
- [3] A. Rämö and H. Toukoma, "Voice quality characterization of IETF Opus codec," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.
- [4] A. Aminuddin, W. Widyawan, and R. Ferdiana, "Perancangan Protokol Hybrid pada Implementasi Voice Over IP (VOIP) Berbasis TCP/IP," *Semnasteknomedia Online*, vol. 4, no. 1, pp. 13–17, 2016.
- [5] B. J. Cox, "Object-oriented programming: an evolutionary approach," 1986.
- [6] M. Huda, *Membangun Aplikasi Database*. Jakarta: PT.Elex Media Komputindo, 2010.
- [7] E. Triandini and I. G. Suardika, *Step by Step Desain Proyek Menggunakan UML*. Penerbit Andi, 2012.
- [8] M. Grechanik, K. S. McKinley, and D. E. Perry, "Recovering and using use-case-diagram-to-source-code traceability links," in *Proceedings of the the 6th joint meeting of the European software engineering conference and the ACM SIGSOFT symposium on The foundations of software engineering*, 2007, pp. 95–104.
- [9] N. Artina, "Penerapan Analisis Kebutuhan Metode Use Case pada Metode Pengembangan Terstruktur," *@ Igoritma*, vol. 2, no. 3, pp. 1–6, 2006.
- [10] R. M. Bastos and D. D. A. Ruiz, "Extending UML activity diagram for workflow modeling in production systems," in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, 2002, pp. 3786–3795.
- [11] L. P. Dewi, U. Indahyanti, and Y. Hari, "Pemodelan Proses Bisnis Menggunakan Activity Diagram UML dan BPMN (Studi Kasus Frs Online)," in *Seminar Nasional Teknik Industri Waluyo Jatmiko V. Surabaya*, 2012.
- [12] H. C. Purchase, L. Colpoys, M. McGill, D. Carrington, and C. Britton, "UML class diagram syntax: an empirical study of comprehension," in *Proceedings of the 2001 Asia-Pacific symposium on Information visualisation-Volume 9*, 2001, pp. 113–120.
- [13] P. Sulistyorini, "Pemodelan Visual dengan Menggunakan UML dan Rational Rose," *Dinamik*, vol. 14, no. 1, 2009.
- [14] X. Li, Z. Liu, and H. Jifeng, "A formal semantics of UML sequence diagram," in *2004 Australian Software Engineering Conference. Proceedings.*, 2004, pp. 168–177.
- [15] A. Nugroho, *rekayasa perangkat lunak menggunakan UML dan JAVA*. Penerbit Andi, 2009.
- [16] A. Aminuddin, "Analisis dan Desain Aplikasi DigiTalkie Berbasis TCP/IP," Universitas Gadjah Mada, 2017.