

Android-Based Person-to-Person Contact Tracing Application for Covid-19 Pandemic in Indonesia

Aplikasi *Contact Tracing* Secara Orang-ke-Orang Berbasis Android Pada Pandemi Covid-19 di Indonesia

Mochammad Naufal Nibros¹⁾, Irwan Alnarus Kautsar²⁾

^{1,2,3)}Program Studi Informatika, Universitas Muhammadiyah Sidoarjo, Indonesia

*Email Penulis Korespondensi: 171080200234@umsida.ac.id¹⁾, irwan@umsida.ac.id²⁾

Abstract. *The Covid-19 pandemic caused problems globally, including Indonesia, and prompted a WHO lockdown program. However, vital community activities require people to be active outside their homes. A digital contact tracing app using barcode technology to record user interactions and collect data on the time and duration of interactions is necessary to monitor and track infected or previously virus-exposed individuals. The collected data is analyzed to determine the level of infection risk for each user and provide contact history information to users and stakeholders. The authors created a person-to-person digital contact tracing application using the waterfall method, facilitating Android-based digital contact tracing that effectively mitigates the spread of Covid-19 in Indonesia.*

Keywords - Android, Covid-19, Digital contact tracing, Waterfall Method

Abstrak. *Pandemi Covid-19 telah menyebabkan berbagai masalah di seluruh dunia, termasuk di Indonesia dan mendorong program lockdown dari WHO. Namun, aktivitas penting masyarakat membutuhkan orang untuk aktif di luar rumah. Aplikasi pelacakan kontak digital yang menggunakan teknologi barcode untuk mencatat interaksi pengguna dan mengumpulkan data tentang waktu dan durasi interaksi diperlukan untuk memantau dan melacak individu yang terinfeksi atau sebelumnya terpapar virus. Data yang terkumpul dianalisis untuk menentukan tingkat risiko infeksi untuk setiap pengguna dan memberikan informasi riwayat kontak kepada pengguna dan stakeholder yang membutuhkan data pelacakan kontak. Peneliti menciptakan pola aplikasi pelacakan kontak digital dari orang ke orang menggunakan metode waterfall, memfasilitasi pelacakan kontak digital berbasis Android yang secara efektif dapat mengurangi penyebaran Covid-19 di Indonesia.*

Kata Kunci - Android, Covid-19, Pelacakan kontak digital

I. PENDAHULUAN

Pandemi Covid-19 merupakan fenomena yang menjadi perhatian dari seluruh dunia pada semua lapisan masyarakat. Covid-19 adalah penyakit yang disebabkan oleh virus corona dan telah ditetapkan sebagai pandemi sejak tanggal 11 Maret 2020 [1]. *Coronaviruses* (CoV) termasuk dalam keluarga virus yang menyebabkan penyakit mulai dari flu biasa hingga penyakit yang lebih serius seperti *Middle East Respiratory Syndrome* (MERS-CoV) dan *Severe Acute Respiratory Syndrome* (SARS-CoV) [2]. Penyakit yang disebabkan oleh virus corona atau dikenal juga dengan Covid-19 ini merupakan virus corona jenis baru yang diberi nama SARS-CoV-2 dan telah ditemukan pada tahun 2019 di Wuhan, China [3]. Penyebaran virus ini tergolong sangat cepat dalam penularannya, sebagai akibatnya Covid-19 juga dapat menyerang anak-anak, bayi, ibu hamil dan menyusui tanpa memandang usia [4]. Dalam penelitian [5] penyebaran varian baru virus SARS-CoV-2 menyebabkan peningkatan jumlah kasus harian positif Covid-19 di seluruh dunia, salah satunya di Indonesia. Berdasarkan data Kementerian Kesehatan RI, per 29 Juni 2021, terdapat lebih dari satu kematian per minggu untuk 100.000 penduduk dan peningkatan tersebut terus berlanjut dari waktu ke waktu.

Sejauh ini, belum ditemukan tentang obat yang dapat secara efektif mengatasi Covid-19 [6]. Perawatan dilakukan secara eksklusif sesuai dengan keluhan pasien dan tingkat keparahannya. Bagi mereka yang mengalami gejala ringan disarankan untuk mengisolasi diri di rumah dengan memantau gejala dan mengikuti pedoman isolasi diri sesuai protokol nasional. Mereka yang memiliki gejala berat harus mencari bantuan dari tenaga medis atau mengisolasi dan merawatnya di rumah sakit. Dengan belum ditemukan obat untuk mengobati penyakit ini, maka tugas terbesarnya adalah mencegah penularan dan penyebarannya. Itu sebabnya organisasi kesehatan dunia (WHO) mengumumkan program *lockdown* yang harus diikuti oleh negara-negara yang terjangkit virus corona [7]. Program utamanya yaitu penerapan *stay at home* dengan menghindari kegiatan diluar rumah. Tindakan *lockdown* dirancang untuk membantu mencegah penyebaran virus corona di suatu wilayah sehingga masyarakat di wilayah tersebut dapat mengambil langkah-langkah agar terhindar dari penyebaran wabah pandemi tersebut [8]. Tetapi hampir aktivitas masyarakat dilakukan di luar rumah sehingga diperlukan penyaluran informasi secara cepat dan efisien untuk pemantauan, pelacakan, dan penghentian penyebaran Covid-19 di Indonesia.

Pelacakan kontak adalah alternatif dari pengganti *lockdown*, karena *lockdown* tidak efektif dalam jangka panjang karena ketidakmampuan pemerintah untuk memperbaiki kerusakan ekonomi yang diakibatkannya. Pelacakan kontak merupakan upaya pencegahan utama yang direkomendasikan WHO untuk memerangi penyebaran Covid-19. Pelacakan kontak manual (tradisional) membutuhkan waktu pelatihan yang lama dan tidak efisien, sehingga diperlukan solusi digital. Teknologi informasi menjadi faktor penting dalam implementasi pelacakan kontak. Saat ini, hampir 80 jenis aplikasi berbasis *smartphone* sedang dikembangkan di beberapa negara dengan menggunakan berbagai teknologi, antara lain: Bluetooth, GPS, dan kode QR. Dari 15 aplikasi pelacakan kontak yang tersedia saat ini, 40% di antaranya menggunakan teknologi GPS, sedangkan 60% sisanya menggunakan teknologi Bluetooth [9].

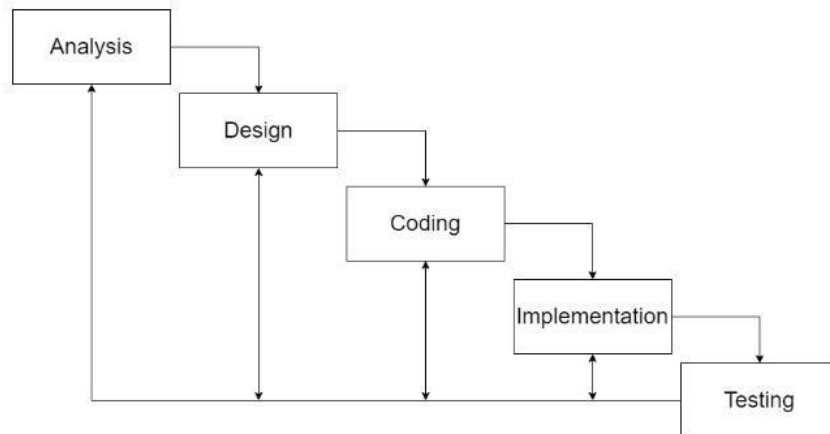
Pada penelitian ini mengusulkan sebuah sistem dimana pengguna menjalankan pelacakan kontak digital dengan melakukan *scan QR Code* untuk mendapatkan informasi kontak, dan mendapatkan notifikasi *case positive* jika sebelumnya pernah melakukan kontak dengan pengguna yang terinfeksi Covid-19. Pola yang digunakan untuk merekam kontak pertemuan adalah menandai jika saling bertemu. Pengembangan *contact tracing* ini disebut juga dengan pola *person-to-person*, yaitu dengan menggunakan riwayat pengguna yang pernah melakukan kontak dengan cara menandai pengguna lain yang ditemuinya, sehingga dengan pola tersebut dapat meningkatkan akurasi pelacakan kontak [10]. Kebutuhan untuk pelacakan kontak dengan pola *person-to-person* dapat meningkatkan proses *tracing* menjadi lebih efisien dengan memungkinkan individu merekam dan melacak interaksi mereka dengan orang lain secara digital. Ini dapat membantu pemerintah kesehatan masyarakat dengan cepat mengidentifikasi individu yang mungkin telah terpapar penyakit dan memberi mereka panduan tentang cara mengisolasi diri atau mencari pertolongan medis jika perlu.

Pada penelitian sebelumnya yang berjudul Sistem Pelacakan Kontak Covid-19 Menggunakan Teknologi *QR Code* Berbasis Web, membahas tentang sistem pelacakan kontak berbasis *QR Code* dengan menggunakan bahasa pemrograman PHP, Javascript dan DBMS MySQL [9]. Pada penelitian tersebut aplikasi tidak memunculkan data lokasi dan masih berbasis website. Sedangkan dalam penelitian terbaru aplikasi menampilkan informasi *latitude longitude device* pengguna dan sudah berbasis Android. Selanjutnya penelitian yang berjudul Model Sistem Pendataan *Contact Tracing* COVID-19 Berbasis Mobile dan Web, membahas tentang perancangan sistem untuk aplikasi pendataan *contact tracing* Covid-19 dengan *QR Code* tetapi dalam penelitian tersebut belum di kembangkan sampai ke tahap implementasi [11]. Penelitian yang berjudul *Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs*, membahas tentang aplikasi ponsel TraceTogether yang dirilis oleh pemerintah Singapura dengan fungsi untuk membantu pejabat kesehatan dalam melacak paparan setelah individu yang terinfeksi diidentifikasi [12]. Perbedaan riset tersebut dengan penelitian terbaru adalah aplikasi menggunakan *bluetooth* sebagai pertukaran data, kemudian dari server mengidentifikasi kedua kontak tersebut apakah salah satu terdapat riwayat terinfeksi virus. Penggunaan *bluetooth* menyebabkan kekhawatiran tentang privasi data dan kerentanan keamanan dikarenakan sebagian besar aplikasi yang rentan terkena menggunakan teknologi *bluetooth*, penggunaan *bluetooth* berpotensi membahayakan pertukaran informasi dan juga data yang mungkin dicuri jika ada peretas yang berhasil masuk ke dalam sistem. Sedangkan dalam riset terbaru pelacakan kontak digital menggunakan teknologi *QR Code* sebagai kunci utama untuk menjaga kebijakan privasi dan keamanan pengguna. Pemakaian *bluetooth* juga terdapat dalam dua penelitian terdahulu yang berjudul *Bluetrace: A Privacy-Preserving Protocol for Community-Driven Contact Tracing Across Borders* [13] dan *Quantifying SARS-Cov-2 Transmission Suggests Epidemic Control With Digital Contact Tracing* [14], membahas tentang pelacakan kontak klasik dengan kuesioner versus penelusuran kontak seketika algoritmik yang dibantu oleh aplikasi ponsel. Kedua penelitian tersebut menjelaskan teknologi yang dipakai masih berbasis *bluetooth* sehingga keamanan privasi masih kurang terjamin, tetapi dalam penelitian *Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing* memiliki persamaan dengan riset ini yaitu aplikasi akan mengirim notifikasi ke pengguna jika terdapat pengguna yang berpotensi tertular setelah melakukan kontak dengan pengguna positif Covid-19.

Secara keseluruhan, penulis ingin membuat suatu software atau sistem yang nantinya akan dapat mengatasi permasalahan penyebaran virus corona (Covid-19) yang semakin cepat. Sehingga dengan adanya aplikasi pelacakan kontak digital pola *person-to-person* dengan menggunakan metode *waterfall* harapannya dapat berperan penting dalam membantu menahan penyebaran penyakit menular khususnya Covid-19 dan melindungi kesehatan masyarakat.

II. METODE

Metode *waterfall* adalah salah satu metode yang dapat ditempuh dalam menerapkan tahapan pengembangan suatu sistem [15]. Dalam metode *waterfall* terdapat lima tahapan yang saling terkait mempengaruhi satu sama lain yaitu tahapan *analysis, design, coding, implementation, dan testing*.



Gambar 1. Metode Waterfall

Pada gambar 1 pengembangan aplikasi menggunakan metode *waterfall* sehingga dalam proses dapat dilakukan secara berurutan dan berkelanjutan.

1. **Analysis**
Tahapan ini bertujuan untuk menganalisis berbagai kebutuhan yang dibutuhkan dalam mengembangkan aplikasi pelacakan kontak digital baik itu berupa identifikasi pengguna, tujuan penelitian, perangkat yang digunakan dan konsep aplikasi, sehingga dapat meminimalisir permasalahan yang akan terjadi terutama dari sisi pengguna.
2. **Design**
Proses ini dilakukan berdasarkan kebutuhan desain sistem dengan penggambaran menggunakan pemodelan *United Modeling Language (UML)* yaitu *sequence diagram*, *activity diagram*, dan *database structure* sebelum coding dimulai. Perancangan *sequence diagram* untuk menjelaskan interaksi antar objek-objek dalam sebuah sistem secara terperinci. Sedangkan perancangan *activity diagram* menjelaskan proses bisnis dan urutan aktivitas dalam sebuah proses.
3. **Coding**
Pada tahapan ini perancangan sistem direalisasikan dalam bentuk rangkaian program atau unit program. penerapan *QR Code* pada Android menggunakan bahasa pemrograman kotlin. *QR Code* adalah sebuah kode yang digunakan untuk menyimpan dan mentransfer data. Penerapan *QR Code* pada Android dapat dilakukan dengan menggunakan *library ZXing (Zebra Crossing)*. Dalam aplikasi ini untuk menampilkan *QR Code* membutuhkan *plaintext* yang di enkripsi menggunakan algoritma AES 256
4. **Implementation**
Implementasi yaitu mengimplementasikan rancangan sistem yang dibuat dari tahap-tahapan sebelumnya untuk membuat aplikasi antarmuka. Beberapa tugas dilakukan pada tahapan implementasi, yaitu mengimplementasikan desain ke dalam kode sumber, kemudian menyempurnakan arsitektur dan menguji sistem. Implementasi menggunakan bahasa pemrograman kotlin dan perangkat lunak Android Studio Dolphin v2021.3.1.
5. **Testing**
Pelaksanaan uji coba dijalankan dengan *alpha testing* untuk memastikan fungsi aplikasi tiap halaman dan proses *create* dan *scanning QR Code* berhasil. Kemudian mengidentifikasi masalah (*bug*) yang terjadi di dalam aplikasi sehingga dapat dilakukan perbaikan sesuai dengan permasalahan yang telah ditemukan. Pada pengujian tahap kedua adalah *beta testing*, uji coba dilakukan untuk memastikan kualitas dan keandalan aplikasi dapat melakukan pelacakan kontak dengan menggunakan pola *person-to-person*.

III. HASIL DAN PEMBAHASAN

A. Analisis Kebutuhan Perangkat Lunak

a. Identifikasi Pengguna

Pada tahapan ini dilakukan analisis pengguna yang akan menggunakan sistem. Proses ini dilakukan untuk mengidentifikasi kebutuhan yang dibutuhkan dalam mengembangkan aplikasi pelacakan kontak digital dari pengguna.

b. Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk menemukan solusi dari sistem yang akan dirancang, yaitu aplikasi *contact tracing* berbasis *mobile device person-to-person*. Hasil dari analisis ini menghasilkan pemahaman tentang keseluruhan sistem sebagai persiapan ke tahap perancangan (desain).

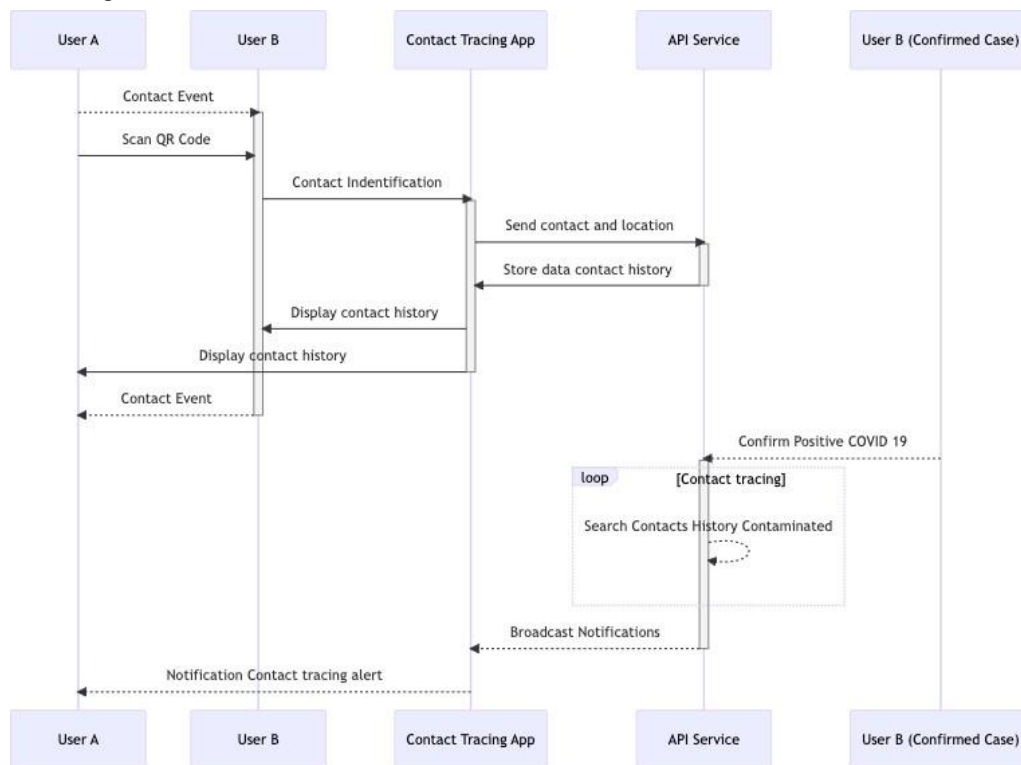
- c. Perangkat yang Digunakan
Untuk menjalankan aplikasi secara sempurna sesuai dengan kebutuhan *smartphone* dibutuhkan *hardware* dan *software* yang mumpuni. Dalam penelitian ini operasi sistem yang dibutuhkan minimum Android 5.1 Lollipop.
- d. Konsep Aplikasi
Konsep aplikasi bisa dilihat dalam tabel berikut:

Tabel 1. Deskripsi Konsep Aplikasi

No.	Nama	Deskripsi
1.	Judul	Aplikasi <i>Contact Tracing</i> metode <i>Person-to-Person</i>
2.	Pengguna	Masyarakat umum di Indonesia
3.	Fitur	<i>Login and Register, Scan QR Code, Contact History, Notification Contact Tracing Positive Case</i>
4.	Interaksi	Sebelum pengguna dapat mengakses fitur utama, pengguna diharuskan untuk <i>login</i> terlebih dahulu. Jika pengguna tidak mempunyai <i>user account</i> bisa melakukan pendaftaran di halaman <i>register</i> . Di dalam menu utama terdapat tombol pindai kode QR, memberikan informasi daftar riwayat kontak saat pengguna melakukan kontak, notifikasi pengguna yang telah melakukan kontak dekat disuatu lokasi dengan seseorang yang telah didiagnosis terpapar COVID-19.

B. Desain UML

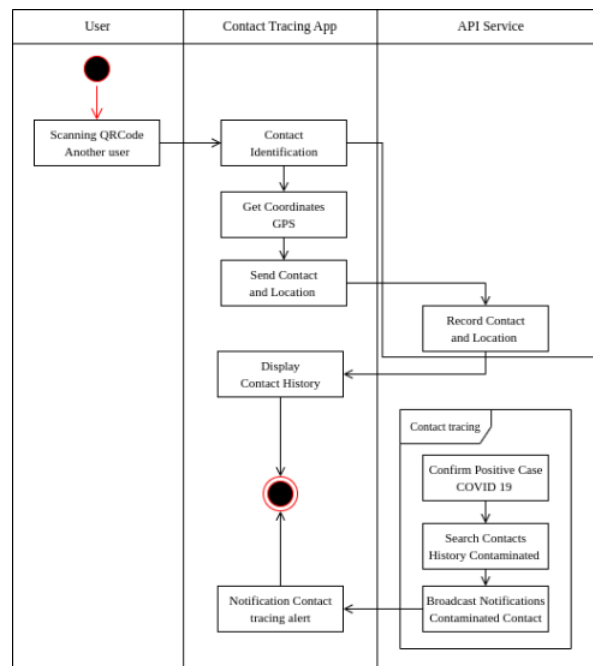
a. Sequence Diagram



Gambar 2. Sequence Diagram

Sequence diagram pada suatu sistem ditampilkan secara visual sehingga menjadi penting dalam memudahkan perancangan dan analisis. *Sequence diagram* dalam bentuk UML paling banyak digunakan untuk memodelkan sistem dinamis, yang berpusat untuk mengidentifikasi alur-alur di suatu sistem. Pada gambar 2 menunjukkan *sequence diagram* peristiwa *user A* melakukan kontak dengan *user B*, kemudian *user B* terkonfirmasi positif Covid-19 sehingga *user A* mendapatkan notifikasi peringatan dari aplikasi pelacakan kontak.

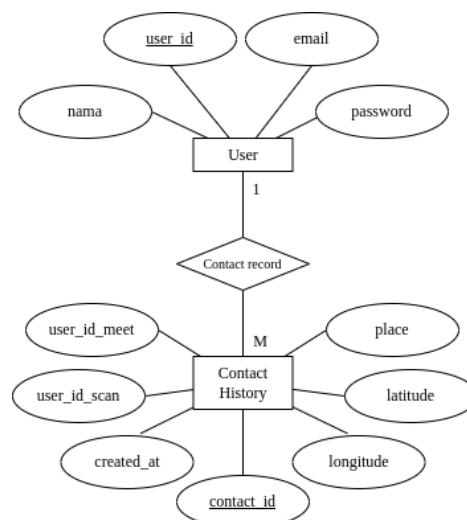
b. Activity Diagram



Gambar 3. Activity Diagram

Diagram aktivitas dapat digunakan untuk menggambarkan bagian dari *activity client* (kerangka kerja), *client* yang memainkan setiap tindakan, dan perkembangan setiap proses secara berurutan. Gambar 3 menunjukkan *activity diagram* di mana pengguna melakukan pemindaian kode QR, kemudian data dikenali dan dikirim ke *API Service* untuk menyimpan kontak. *API Service* kemudian akan menerima pemicu konfirmasi positif Covid-19 dan mengirimkan pesan *broadcast* ke pengguna yang terkontaminasi akibat kontak sebelumnya.

c. Database Structure



Gambar 4. Relation Entity Relationship Diagram

Pembuatan skema database memerlukan penyimpanan data yang dipetakan ke *API Service* di mana data ini akan digunakan sebagai acuan *contact tracing*. Rancangan tersebut dibuat agar sesuai dalam pengembangan sistem. Data dasar dalam perancangan aplikasi ini menggunakan metode Entity Relationship Diagram (ERD). Gambar 4 menunjukkan relasi antar entitas secara keseluruhan di mana ERD dari sistem melibatkan 2 entitas yaitu *user* dan *contact history*.

C. Pembuatan Kode Program

a. QR Code Implementation

Berikut ini penerapan kode QR pada aplikasi dalam penelitian ini:

1. Membuat Enkripsi dan Deskripsi AES 256

Input dan *output* dari algoritma AES terdiri dari urutan data sebesar 128 bit. Urutan data yang sudah terbentuk dalam satu kelompok 128bit tersebut disebut juga sebagai *plaintext* yang nantinya akan dienkripsi menjadi *cipher text*. *Cipher key* dari AES dengan panjang 256 bit. Kode program di bawah ini digunakan untuk membuat enkripsi & dekripsi pada aplikasi.

```
private fun cipher(opmode: Int, secretKey: String, initVector: String): Cipher {
    val c = Cipher.getInstance("AES/CBC/PKCS5Padding")
    val hashSec = MessageDigest.getInstance("SHA-256")
    hashSec.update(secretKey.toByteArray(Charsets.UTF_8))
    val secBytes = hashSec.digest().copyOfRange(0, 32)
    val hashIV = MessageDigest.getInstance("SHA-256")
    hashIV.update(initVector.toByteArray(Charsets.UTF_8))
    val ivBytes = hashIV.digest()
    val iv = IvParameterSpec(ivBytes.toHexString().substring(0,
16).toByteArray(Charsets.UTF_8))
    val sk = SecretKeySpec(secBytes.toHexString().substring(0,
32).toByteArray(Charsets.UTF_8), "AES")
    c.init(opmode, sk, iv)
    return c
}
```

Gambar 5. Kode Program Metode Cipher

Metode **cipher()** diatas digunakan untuk melakukan enkripsi dan dekripsi dengan input parameter *Cipher mode*, *secretkey* dan *initvector*. Metode tersebut dipanggil menghasilkan *output Cipher*.

```
fun encrypt(str: String, secretKey: String, initVector: String):String {
    val encrypted = cipher(Cipher.ENCRYPT_MODE, secretKey,
initVector).doFinal(str.toByteArray(Charsets.UTF_8))
    return String(Base64.encode(String(Base64.encode(encrypted,
Base64.DEFAULT)).toByteArray(Charsets.UTF_8), Base64.DEFAULT))
}

fun decrypt(str: String, secretKey: String, initVector: String):String{
    val byteStr = Base64.decode((str.toByteArray(Charsets.UTF_8)),
Base64.DEFAULT)
    return String(cipher(Cipher.DECRYPT_MODE, secretKey,
initVector).doFinal(byteStr))
}
```

Gambar 6. Kode Program Metode Encrypt

Metode **encrypt()** diatas digunakan untuk melakukan enkripsi *plaintext* kemudian memanggil metode **cipher()** dengan menggunakan mode *Cipher.ENCRYPT_MODE* dan menghasilkan *output string* yang di *encode* menggunakan base64. Sedangkan metode **decrypt()** digunakan untuk melakukan dekripsi base64 kemudian memanggil metode **cipher()** dengan menggunakan mode *Cipher.DECRYPT_MODE* dan menghasilkan *output string plaintext*.

2. Membuat QR Code pada Android

QR Code biasanya digunakan untuk menyimpan informasi seperti *link website*, nomor telepon, atau alamat email. Pada penelitian ini menggunakan hasil enkripsi dari AES 256. Dibawah ini kode program digunakan untuk membuat QR Code:

```

val encrypted =
AES256.encrypt("{\"unique_id\":12132,\"url\":\"deomain.com/src/image.png\",\"create
d_at\":1671037861349}\", secretKey, initVector)

with(binding.homeImgQrcode)
  /** Kode untuk encode ke QR Code ***/
  setImageBitmap(QRLEncoder(encrypted, QRGContents.Type.TEXT, 800).bitmap)
}

```

Gambar 7. Kode Program Membuat QR Code

3. Membaca QR Code pada Android

Untuk membaca *QR Code* dibutuhkan akses *permission camera*, saat kamera di akses kemudian menangkap pola *QR Code* pada kamera. Kemudian aplikasi akan mengirimkan *callback* ke *method onActivityResult* dengan membawa variabel berisi *string*. Setelah itu *content callback* yang dikirimkan di *decrypt* dengan AES 256. Berikut potongan kode program yang digunakan untuk membaca *QR Code*:

```

override fun onActivityResult(requestCode: Int, resultCode: Int, data: Intent?) {
    val parser = IntentIntegrator.parseActivityResult(requestCode, resultCode, data)
    val output = AES256.decrypt(String(Base64.decode(parser.contents,
Base64.DEFAULT)), secretKey, initVector)
}

```

Gambar 8. Kode Program Membaca QR Code

b. Broadcast Notification FCM

Pada tahapan ini menjelaskan bagaimana *broadcast notification* bekerja di Android dan API. Dengan memanfaatkan layanan yang ada pada firebase yaitu *Firebase Cloud Messaging* (FCM) dapat mengirimkan notifikasi ke pengguna, bahkan jika aplikasi tersebut tidak sedang terbuka atau dijalankan. Ini sangat bermanfaat untuk mengirim notifikasi penting kepada pengguna, seperti pemberitahuan konfirmasi riwayat *contact positive COVID-19*. Berikut ini penerapan notifikasi pada aplikasi dalam penelitian ini:

1. Mendefinisikan Metode *subscribeToTopic* pada Aplikasi Android

Di *Firebase Cloud Messaging* (FCM) memungkinkan untuk menerima pesan yang dipublikasikan ke suatu *topic*. Untuk menggunakan fitur tersebut dapat memanggil fungsi ***subscribeToTopic()*** untuk berlangganan topik berdasarkan *email user*, kemudian mengimpor modul *Firebase Messaging* dan memanggil fungsi *subscribeToTopic* pada *instance Firebase Messaging*.

```

Firebase.messaging.subscribeToTopic(user_email).addOnCompleteListener { task ->
    var msg = getString(R.string.msg_subscribed)
    if (!task.isSuccessful) {
        msg = getString(R.string.msg_subscribe_failed)
    }
    Toast.makeText(baseContext, msg, Toast.LENGTH_SHORT).show()
}

```

Gambar 9. Kode Program Broadcast Notification

2. Menerima Pesan yang Dikirimkan ke Aplikasi Android

Agar aplikasi dapat menerima notifikasi secara *realtime* membutuhkan *class services Firebase Messaging Service*. *Firebase Messaging Service* adalah kelas layanan yang digunakan untuk mengelola fungsionalitas perpesanan dalam aplikasi Android, termasuk menerima dan menangani pesan, menampilkan notifikasi, dan melakukan tindakan tertentu berdasarkan konten pesan secara *real-time*.

```
class HandlingMessagingService : FirebaseMessagingService() {
    override fun onMessageReceived(remoteMessage: RemoteMessage) {
        if (remoteMessage.data.isNotEmpty()) {
            remoteMessage.notification?.let { it.body?.let { body ->
showNotification(body) } }
        }
    }
    private fun showNotification(messageBody: String) {
        val intent = Intent(this, MainActivity::class.java)
        intent.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP)

        val pendingIntent = PendingIntent.getActivity(this, 0, intent,
PendingIntent.FLAG_IMMUTABLE)

        val channelId = getString(R.string.default_notification_channel_id)
        val defaultSoundUri =
RingtoneManager.getDefaultUri(RingtoneManager.TYPE_NOTIFICATION)
        val notificationBuilder =
            NotificationCompat.Builder(this, channelId)
                .setSmallIcon(R.drawable.ic_stat_ic_notification)
                .setContentTitle(getString(R.string.fcm_message))
                .setContentText(messageBody)
                .setAutoCancel(true)
                .setSound(defaultSoundUri)
                .setContentIntent(pendingIntent)

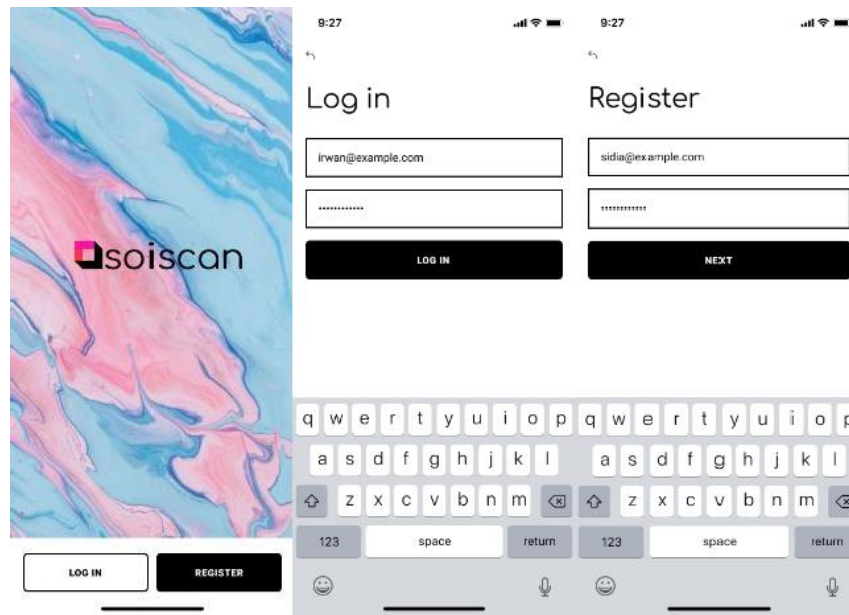
        val notificationManager =
            getSystemService(Context.NOTIFICATION_SERVICE) as NotificationManager

        notificationManager.notify(0, notificationBuilder.build())
    }
}
```

Gambar 10. Kode Program Menerima Notifikasi

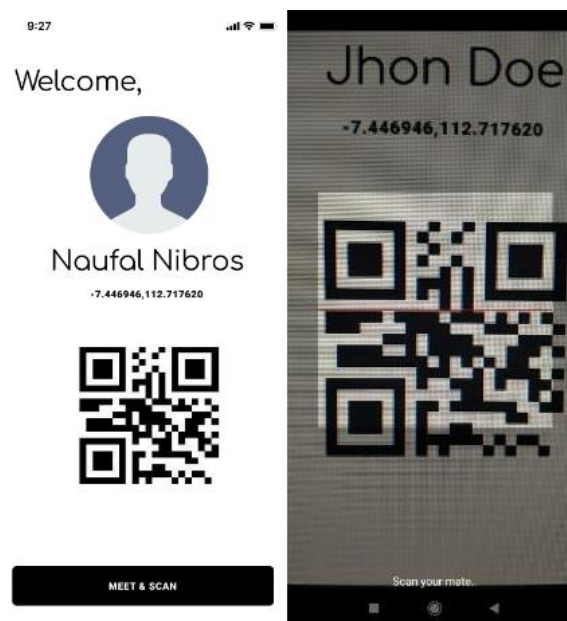
D. Implementasi Aplikasi

Berikut ini adalah implementasi antar muka sistem aplikasi pelacakan kontak digital mulai dari tampilan *splashscreen* hingga tampilan halaman *preview* yang telah dibangun:



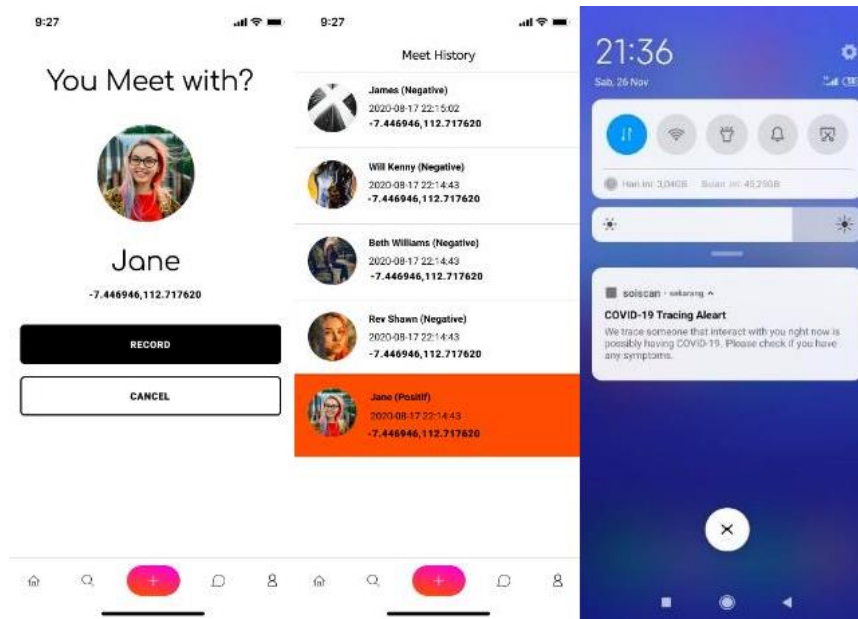
Gambar 11. Halaman Splashscreen, Login, dan Register

Pada gambar 11 menjelaskan tampilan antarmuka dibuat sesuai dengan desain antarmuka sebelumnya. Halaman *login* digunakan untuk mengautentikasi sebelum menggunakan fitur aplikasi, sedangkan halaman *register* digunakan untuk mendaftarkan *user* yang belum memiliki akun.



Gambar 12. Halaman Home dan Scan QR Code

Pada gambar 12 menjelaskan fitur utama aplikasi hanya dapat diakses setelah berhasil melakukan *login* dengan memasukkan *email* dan *password*, selanjutnya *user* akan masuk ke *home page*. Pada halaman *home*, pengguna memiliki informasi tentang kode QR yang dapat dipindai oleh pengguna lain. Terdapat *button scan* yang berfungsi sebagai akses kamera untuk memindai kode QR pengguna lain.



Gambar 13. Halaman Preview dan Notifikasi

Pada gambar 13 menjelaskan setelah pengguna berhasil melakukan *scan QR Code*, hasil *preview* dari profil pengguna yang di pindai akan ditampilkan. Pada halaman *preview scan profile* juga terdapat *action record* untuk menyimpan kontak pertemuan, *action cancel* digunakan untuk membatalkan *record* kontak pertemuan. Jika pengguna melakukan *record* pertemuan dengan pengguna lain maka pada halaman *meet history* akan menyimpan rekaman pertemuan dan menampilkannya dalam *list history meet*.

E. Hasil Alpha dan Beta Testing

a. Pengujian Alpha

Pengujian *Alpha* menggunakan teknologi *blackbox*, teknik ini menguji apakah aplikasi bekerja dengan baik pada sistem operasi *user*. Di bawah ini adalah hasil pengujian *Alpha* pada aplikasi pelacakan kontak digital.

Tabel 2. Pengujian Alpha Halaman Authentikasi

Skenario	Hasil Diharapkan	Hasil Uji
Klik <i>button Login</i>	Navigasi ke Halaman <i>Login</i>	Ok
Klik <i>button Register</i>	Navigasi ke Halaman <i>Register</i>	Ok

Berdasarkan hasil pengujian *alpha* pada tabel 2 untuk halaman autentikasi menunjukkan bahwa aplikasi berhasil diakses sesuai dengan yang diharapkan, pengujian untuk menekan pilihan *login* menuju halaman *login* dan pengujian untuk menekan pilihan *register* menuju halaman *register* telah berhasil dilakukan.

Tabel 3. Pengujian Alpha Halaman Login

Skenario	Hasil Diharapkan	Hasil Uji
Menginputkan <i>Email</i> yang benar	Navigasi ke Halaman <i>Home</i>	Ok
Menginputkan <i>Password</i> yang benar	Navigasi ke Halaman <i>Home</i>	Ok
Menginputkan <i>Email</i> yang salah	Muncul pesan “Email yang Anda masukkan salah”	Ok
Menginputkan <i>Password</i> yang salah	Muncul pesan “Email yang Anda masukkan salah”	Ok

Berdasarkan hasil pengujian pada tabel 3 untuk navigasi *login* ke halaman *home* menunjukkan bahwa aplikasi telah mencapai hasil yang diharapkan, pengujian untuk memasukkan *email*, *password*, dan kata pilihan yang benar akan menuju halaman *home* aplikasi.

Tabel 4. Pengujian Alpha Halaman Register

Skenario	Hasil Diharapkan	Hasil Uji
----------	------------------	-----------

Menginputkan <i>Email</i> yang benar	Navigasi ke Halaman <i>Home</i>	Ok
Menginputkan <i>Password</i> yang benar	Navigasi ke Halaman <i>Home</i>	Ok
Menginputkan Nama yang benar	Navigasi ke Halaman <i>Home</i>	Ok
Menginputkan Nama tidak diisi	Muncul pesan “Nama harus diisi”	Ok
Menginputkan <i>Email</i> tidak sesuai format	Muncul pesan “Email yang Anda masukkan tidak sesuai”	Ok
Menginputkan <i>Email</i> tidak diisi	Muncul pesan “Email harus diisi”	Ok
Menginputkan <i>Password</i> tidak diisi	Muncul pesan “Password harus diisi”	Ok

Berdasarkan hasil pengujian pada tabel 4 untuk navigasi *register* ke halaman *home* menunjukkan bahwa aplikasi telah sesuai dengan hasil yang diharapkan, pengujian untuk memasukkan *email*, *password*, nama, dan kata pilihan yang benar akan menuju ke halaman *home* aplikasi.

Tabel 5. Pengujian Alpha Halaman Home

Skenario	Hasil Diharapkan	Hasil Uji
Membuka Halaman <i>Home</i>	Menampilkan Nama Pengguna	Ok
	Menampilkan <i>Latitude Longitude Device</i> pengguna	Ok
	Menampilkan <i>QR Code</i> pengguna	Ok
Klik button “ <i>Meet & Scan</i> ”	Navigasi ke Halaman <i>Scan QR Code</i>	Ok

Berdasarkan hasil pengujian tabel 5 untuk navigasi *home* menunjukkan bahwa aplikasi telah sesuai dengan hasil yang diharapkan, pengujian untuk menampilkan nama pengguna, *latitude longitude device*, dan *QR Code* dapat dilakukan dengan mudah, pengujian untuk menekan pilihan *meet & scan* menuju halaman *scan QR Code* berhasil dilakukan.

Tabel 6. Pengujian Alpha Navigasi Scan QR Code

Skenario	Hasil Diharapkan	Hasil Uji
<i>Scan QR Code</i> berhasil	Navigasi ke Halaman Hasil <i>Scan QR Code</i>	Ok
<i>Scan QR Code</i> gagal	Muncul pesan “QR Code tidak valid”	Ok

Berdasarkan hasil pengujian pada tabel 6 untuk navigasi *scan QR Code* menunjukkan bahwa aplikasi telah sesuai dengan hasil yang diharapkan, pengujian untuk pemindaian kode QR dapat dilakukan dengan mudah.

Tabel 7. Pengujian Alpha Halaman Hasil Scan QR Code

Skenario	Hasil Diharapkan	Hasil Uji
Navigasi ke Halaman Hasil <i>Scan QR Code</i>	Menampilkan Nama pengguna	Ok
	Menampilkan <i>Latitude Longitude Device</i> pengguna	Ok
Klik button “ <i>Record</i> ”	Menyimpan kontak pertemuan ke Halaman <i>Meet History</i>	Ok
Klik button “ <i>Cancel</i> ”	Tidak menyimpan kontak pertemuan ke Halaman <i>Meet History</i>	Ok

Berdasarkan hasil pengujian pada tabel 7 untuk halaman hasil *scan QR Code* menunjukkan aplikasi telah sesuai dengan hasil yang diharapkan, pengujian untuk menampilkan nama pengguna, dan *latitude longitude device* dapat dilakukan dengan mudah, pengujian untuk menekan pilihan *record* menyimpan kontak pertemuan halaman *meet history* berhasil dilakukan, pengujian untuk menekan pilihan *cancel* tidak menyimpan kontak pertemuan halaman *meet history* berhasil dilakukan.

Tabel 8. Pengujian Alpha Halaman Meet History

Skenario	Hasil Diharapkan	Hasil Uji
Membuka Halaman <i>Meet History</i>	Menampilkan daftar riwayat pertemuan <i>case Negative</i>	Ok
	Menampilkan daftar riwayat pertemuan <i>case Positive</i>	Ok

Berdasarkan hasil pengujian pada tabel 8 untuk navigasi halaman *meet history* menunjukkan bahwa aplikasi telah sesuai dengan hasil yang diharapkan, pengujian untuk menampilkan daftar riwayat pertemuan *case negative* dan *case positive* telah sesuai dengan hasil yang diharapkan.

Tabel 9. Pengujian Alpha Push Notifikasi

Skenario	Hasil Diharapkan	Hasil Uji
Terdapat <i>Case Positive</i> pada Riwayat Pertemuan dalam rentang waktu H+7	Mendapatkan notifikasi peringatan “Contact Tracing Alert COVID19”	Ok

Berdasarkan hasil pengujian pada tabel 9 untuk *case positive* pada riwayat pertemuan dalam rentang waktu 7 hari menunjukkan bahwa aplikasi telah berhasil mendapatkan notifikasi peringatan “*contact tracing alert Covid-19*” sesuai dengan hasil yang diharapkan.

b. Pengujian Beta

Pengujian beta adalah pengujian langsung di lingkungan nyata pengguna tanpa kehadiran pihak developer. Pengujian ini dilakukan di Universitas Muhammadiyah Sidoarjo dengan responden yang dipilih secara acak yang memenuhi kriteria seperti pengguna ponsel Android dan sering berinteraksi dengan orang lain, dan didapatkan 8 responden untuk mencoba aplikasi pelacakan kontak digital. Kuesioner uji beta adalah sarana yang digunakan oleh pengguna aplikasi untuk memberikan penilaian terhadap aplikasi yang sedang dibangun. Berdasarkan kuesioner tersebut akan dilakukan perhitungan untuk menarik kesimpulan tentang evaluasi terhadap aplikasi yang dirancang. Kuesioner ini memuat 9 pertanyaan dengan menggunakan skala *likert* dengan skala 1 sampai dengan 4. Tabel 10 merupakan skor penilaian yang ditentukan dengan menggunakan skala *likert* untuk setiap pertanyaan dalam kuesioner.

Tabel 10. Skor Penilaian

Tingkat Kepuasan	Skala
Sangat setuju	4
Setuju	3
Tidak setuju	2
Sangat tidak setuju	1

Berdasarkan data dari hasil kuesioner, diperoleh persentase dari masing-masing jawaban dengan menggunakan rumus:

$$y = \frac{\sum(N.R)}{Skor\ Ideal} \times 100\% \quad (1)$$

Dimana:

- Y = Hasil persentase pengujian
- N = Skala *likert*
- R = Jumlah responden
- Skor Ideal = Skala *likert* tertinggi x Total responden

Tabel 11. Konversi Persentase

Tingkat Kepuasan	Persentase (%)
Sangat membantu	> 80%
Cukup Membantu	60% < 80%
Kurang Membantu	30% < 60%
Tidak Membantu	< 30%

Pengujian beta dalam penelitian ini menggunakan konversi persentase untuk mengukur seberapa baik sebuah aplikasi bekerja dan seberapa besar kemungkinan aplikasi tersebut memiliki bug atau error. Tingkat konversi pengujian beta mengacu pada persentase pengguna saat mencoba aplikasi. Konversi persentase bisa dilihat dalam tabel 11.

Tabel 12. Pengujian Beta Aplikasi Pelacakan Kontak Digital

Pertanyaan	Jawaban Responden	(N) Skala	(R) Responden	N.R	$\Sigma N.R$	Y	Hasil
1. Aplikasi ini mudah diakses oleh pengguna	Sangat setuju	4	4	16	25	78,125%	Aplikasi ini cukup membantu dalam mengakses <i>QR Code</i>
	Setuju	3	2	6			
	Tidak setuju	2	1	2			
	Sangat tidak setuju	1	1	1			
2. Desain Interaksi Aplikasi ini mudah dimengerti oleh pengguna	Sangat setuju	4	4	16	27	84,375%	Aplikasi ini menggunakan UI yang simple dan bisa membantu pengguna
	Setuju	3	3	9			
	Tidak setuju	2	1	2			
	Sangat tidak setuju	1	0	0			
3. Pengguna mudah mengakses Halaman <i>Login</i> pada Aplikasi	Sangat setuju	4	4	16	28	87,5%	<i>User</i> bisa terbantu dengan <i>form login</i> yang mudah diakses
	Setuju	3	4	12			
	Tidak setuju	2	0	0			
	Sangat tidak setuju	1	0	0			
4. Pengguna mudah mengakses Halaman <i>Register</i> pada Aplikasi	Sangat setuju	4	4	16	27	84,375%	Pengisian data diri sangat mudah diakses
	Setuju	3	3	9			
	Tidak setuju	2	1	2			
	Sangat tidak setuju	1	0	0			
5. Pengguna mudah mengakses Halaman <i>Home</i> pada Aplikasi	Sangat setuju	4	4	16	26	81,25%	<i>User</i> mudah mengakses <i>Home</i> dengan menampilkan informasi yang jelas
	Setuju	3	2	6			
	Tidak setuju	2	2	4			
	Sangat tidak setuju	1	0	0			
6. Pengguna mudah melakukan <i>Scan QR Code</i> pada Aplikasi	Sangat setuju	4	4	16	27	84,375%	<i>User</i> mudah melakukan <i>scanning QR Code</i> dan mudah terbaca
	Setuju	3	3	9			
	Tidak setuju	2	1	2			
	Sangat tidak setuju	1	0	0			
7. Desain Interaksi <i>Preview Scan QR Code</i> mudah dimengerti pengguna	Sangat setuju	4	4	16	26	81,25%	<i>User</i> mudah mengakses <i>scan barcode</i> dan mudah terkonfirmasi
	Setuju	3	2	6			
	Tidak setuju	2	2	4			
	Sangat tidak setuju	1	0	0			
8. Pengguna mudah mengakses Halaman <i>Meet History</i> pada Aplikasi	Sangat setuju	4	4	16	25	78,125%	Tampilan <i>meet history</i> cukup membantu dengan adanya titik koordinat yang tepat
	Setuju	3	2	6			
	Tidak setuju	2	1	2			
	Sangat tidak setuju	1	1	1			
9. Pesan Notifikasi peringatan riwayat pelacakan mudah dimengerti pengguna	Sangat setuju	4	3	12	25	78,125%	Notifikasi yang di dapat terkadang tidak <i>real time</i>
	Setuju	3	3	9			
	Tidak setuju	2	2	4			
	Sangat tidak setuju	1	0	0			

Tabel 12 memiliki beberapa nilai dengan persentase yang tidak cukup tinggi bisa dilihat dalam pertanyaan nomor 1,8, dan 9. Angka 1 menunjukkan persentase sebesar 78,125%. Hal ini dipicu oleh perbedaan poin yang diberikan oleh responden. Analisis penilaian ini dapat disebabkan oleh beberapa faktor seperti perbedaan pengalaman pengguna saat menggunakan teknologi atau aplikasi. Pengguna yang paham teknologi merasa aplikasi ini sangat membantu

sementara pengguna yang kurang paham teknologi masih merasa kesulitan. Variabel lain yang dapat mempengaruhi hasil adalah perangkat yang digunakan untuk mengakses aplikasi atau kondisi jaringan internet. Jika perangkat atau jaringan internet tidak mendukung aplikasi, pengguna dapat kesulitan menggunakan aplikasi tersebut. Pada poin 8 aplikasi menampilkan koordinat tanpa teks nama tempat, hal ini membantu pengguna karena titik koordinat dapat menunjukkan lokasi yang tepat. Tetapi jika ingin mengetahui lokasi tersebut pengguna harus mengecek nomor titik koordinat dengan gmaps sehingga hal ini cukup membuat repot pengguna. Pada poin 9, pesan notifikasi ditampilkan dalam bahasa inggris, hal ini membuat beberapa pengguna bingung atau tidak memahami maksud dari pesan tersebut jika pengguna tidak menguasai bahasa inggris. Pesan notifikasi peringatan dapat dianggap tidak efektif jika pengguna tidak memahami pesan yang ditampilkan. Selebihnya hasil pengujian *beta* memiliki persentase cukup tinggi hal tersebut menunjukkan bahwa dalam pembuatan dan penggunaan aplikasi pelacakan kontak digital sudah sesuai tujuan, yaitu aplikasi cukup membantu dalam fungsionalitasnya, mudah digunakan (*user friendly*), dan tata letak yang sesuai sehingga memudahkan pengguna memperoleh informasi kontak. Dalam keseluruhan, hasil tersebut menunjukkan betapa kompleksnya pengalaman pengguna dan faktor-faktor yang memengaruhinya. Oleh karena itu, untuk memaksimalkan manfaat dari sebuah aplikasi, penting bagi pengembang untuk memperhatikan *feedback* dari pengguna dan terus memperbaiki aplikasi tersebut.

VII. SIMPULAN

Penelitian ini akhirnya menggambarkan bahwa aplikasi *contact tracing* berbasis Android dengan menggunakan pola *person-to-person* telah berhasil dirancang dan dibangun dengan menggunakan bahasa pemrograman kotlin dan IDE Android Studio. Aplikasi yang dibuat dapat melakukan *record contact* pertemuan menggunakan *QR Code*, dapat menerima notifikasi dan *contact history* pertemuan. Aplikasi ini memungkinkan individu untuk merekam dan melacak interaksi pengguna dengan orang lain, yang dapat membantu pejabat kesehatan masyarakat dengan cepat mengidentifikasi dan menilai kontak dekat apakah tertular Covid-19 atau tidak, serta memberi pengguna panduan dan dukungan yang diperlukan. Namun, penting bagi aplikasi pelacakan kontak untuk dikembangkan dan diterapkan dengan cara yang menghargai privasi dan keamanan pengguna. Dalam pengembangan aplikasi *contact tracing* saat ini dilakukan hanya sampai pada tahap memberikan informasi lokasi *user*. Penting juga agar aplikasi ini disertai dengan tindakan lain, seperti pengujian dan karantina, sehingga efektif dalam menahan penyebaran penyakit menular. Secara keseluruhan, aplikasi pelacakan kontak digital dapat menjadi alat yang berguna dalam memerangi penyakit menular, tetapi harus digunakan sebagai bagian dari pendekatan komprehensif untuk kesehatan masyarakat. Hasil pengujian dengan menggunakan metode *alpha* dan *beta testing* menunjukkan bahwa komponen uji dapat berfungsi sebagaimana yang diharapkan. Sistem ini tidak memiliki riwayat kesehatan pengguna yang dapat digunakan sebagai parameter kerentanan terhadap virus menular seperti Covid-19, diharapkan penelitian lebih lanjut dapat dikembangkan.

UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih yang sebesar-besarnya kepada Universitas Muhammadiyah Sidoarjo dan kepada pihak-pihak yang membantu pelaksanaan penelitian hingga selesainya penulisan laporan penelitian ini.

REFERENSI

- [1] J. Moudy and R. A. Syakurah, "Pengetahuan terkait Usaha Pencegahan Coronavirus Disease (COVID-19) di Indonesia," vol. 4, no. 3, pp. 333–346, 2020.
- [2] J. Biomed *et al.*, "Coronavirus vaccine development : from SARS and MERS to COVID - 19," pp. 1–23, 2020, doi: 10.1186/s12929-020-00695-2.
- [3] D. I. Ngronggah, R. Medis, F. Kesehatan, U. D. Bangsa, D. P. Sari, and N. S. Atiqoh, "Dengan Kepatuhan Penggunaan Masker Sebagai Upaya Pencegahan Penyakit Covid-19," vol. 10, no. 1, 2020.
- [4] A. Khomaroh and Puspitasari, "Hubungan Jumlah Leukosit dan Laju Endap Darah Terhadap C- Reactive Protein pada Pasien Terkonfirmasi Covid-19 di Rumah Sakit Umum Daerah Gresik," vol. 22, pp. 1–10, 2023, doi: 10.21070/ijins.v22i.844.
- [5] R. T. Yakhamid and N. A. R. Zaqi, "Efektivitas PPKM Darurat Dalam Penanganan Lonjakan Kasus Covid-19," vol. 19, pp. 235–244, 2021.
- [6] C. W. Morfi *et al.*, "Kajian Terkini CoronaVirus Disease 2019 (COVID-19)," vol. 2019, pp. 1–8, 2020.
- [7] S. Djelantik, "Kerjasama Global Menangani The ‘ Great Lockdown ’; Pendekatan Diplomasi Multijalur," 2020.
- [8] A. W. Saputra and I. Simbolon, "Hubungan Tingkat Pengetahuan Tentang Covid- Untuk Mengurangi Penyebaran Covid-19 Di Kalangan Mahasiswa Berasrama Universitas Advent Indonesia," pp. 1–7, 2020.

-
- [9] M. H. Dar, B. Bangun, M. Informatika, U. Labuhanbatu, T. Informasi, and U. Labuhanbatu, "Sistem pelacakan kontak covid-19 menggunakan teknologi qr code berbasis web," vol. VII, no. 3, pp. 283–292, 2021.
- [10] S. Ojagh, S. Saeedi, and S. H. L. Liang, "A Person-to-Person and Person-to-Place COVID-19 Contact Tracing System Based on OGC IndoorGML," 2021.
- [11] S. Gatenia and A. Prabawa, "Model Sistem Pendataan Contact Tracing COVID-19 Berbasis Mobile dan Web," vol. 6, no. 1, pp. 1801–1808, 2021.
- [12] H. Cho, D. Ippolito, and Y. W. Yu, "Contact Tracing Mobile Apps for COVID-19: Privacy Considerations and Related Trade-offs," 2020.
- [13] J. Bay *et al.*, "BlueTrace : A privacy-preserving protocol for community-driven contact tracing across borders," 2020.
- [14] L. Ferretti *et al.*, "Quantifying SARS-CoV-2 transmission suggests epidemic control with digital contact tracing," vol. 6936, pp. 0–7, 2020, doi: 10.1126/science.abb6936.
- [15] T. Cahyono, S. Setianingsih, and D. Iskandar, "Implementation Of The Waterfall Method In The Design Of A Website-Based Book Lending System," vol. 3, no. 3, 2022.