Sistem Deteksi Nomor Telepon dan Rekening Bank Terindikasi Penipuan Berbasis Aplikasi Android dan Web

Joko Handoko[1], Kevin[2], Paulus[3], Zulpa Salsabila[4]
[1,2,3,4]Universitas Mikroskil, Jl. M.H. Thamrin, No. 112, 124, 140, Medan 20212
[1,2,3,4]Fakultas Informatika, Program Studi Sistem Informasi, Universitas Mikroskil, Medan
[1]joko.han1201@gmail.com, [2]thatkevin2000@gmail.com, [3]paulus@mikroskil.ac.id,
[4]zulpa.salsabila@mikroskil.ac.id

Abstrak

Hampir setiap pengguna ponsel pernah menerima panggilan atau pesan masuk dari nomor telepon yang tidak dikenal. Terkadang, panggilan atau pesan masuk demikian adalah penipuan untuk menjebak pengguna ponsel melakukan transfer uang ke rekening bank. Pengguna ponsel rentan menjadi sasaran tindakan kriminal karena sulit mendeteksi nomor telepon dan rekening bank yang dipakai untuk penipuan. Penelitian ini dimaksudkan untuk menghasilkan solusi aplikasi Android agar pengguna dapat berbagi laporan penipuan, menelusuri riwayat laporan nomor telepon dan rekening bank tertentu yang terindikasi dipakai untuk penipuan, serta mendeteksi dan memblokir panggilan masuk yang berisiko dengan tetap menghargai privasi pengguna. Aplikasi Android didukung dengan situs *web* khusus untuk mengelola informasi yang dibutuhkan. Keseluruhan sistem dikembangkan melalui proses berulang dan bertahap dengan metode *Rapid Application Development* (RAD) dan beberapa alat pengembangan yaitu: MariaDB, Kotlin, Android Studio, dan Laravel 7. Aplikasi Android dan situs *web* yang dihasilkan masing-masing telah melewati 37 dan 17 kasus uji dalam pengujian *blackbox* sehingga telah layak dipakai untuk meningkatkan kenyamanan pengguna dalam menggunakan ponsel karena mampu mendeteksi nomor telepon dan rekening bank yang terindikasi penipuan.

Kata kunci: privasi pengguna, blokir panggilan masuk, panggilan penipuan, pesan penipuan, laporan penipuan

Abstract

Almost every cellphone user has received incoming calls or messages from unknown phone numbers. Sometimes, these are scam calls or messages to trap cellphone users doing money transfers to bank accounts. Mobile phone users are vulnerable to this criminal act because they are hard to detect the phone numbers and bank accounts used for scams. This research aims to produce an Android app solution so that users can share scam reports, browse the report history of specific phone numbers and bank accounts used for scams, and detect and block risky incoming calls while respecting user privacy. The Android app is supported with a dedicated website to manage the required information. The entire system was developed through an iterative and gradual process using the Rapid Application Development (RAD) method and several development tools: MariaDB, Kotlin, Android Studio, and Laravel 7. The Android application and website have passed 37 and 17 test cases in black box testing. Systems have been suitable for increasing user convenience in using a mobile phone because they can detect scam-indicated phone numbers and bank accounts.

Keywords: *user privacy, block incoming calls, scam calls, scam messages, scam reports*

1. PENDAHULUAN

Modus penipuan yang sering dilakukan melalui telepon adalah mengatasnamakan suatu perusahaan/bank dan lembaga keuangan untuk meminta kode verifikasi *One Time Password* (OTP) atau *token*/*Personal Identification Number* (PIN) untuk mengambil isi saldo korban. Para korban bisa berasal

dari berbagai kalangan, mulai dari masyarakat umum hingga para profesional yang bahkan profesinya erat berkaitan dengan bidang keamanan [1]. Menurut FTC, lembaga perlindungan konsumen di Amerika Serikat, kontak telepon menjadi sarana penipuan tertinggi yaitu mencapai 31% dari 498.000 kasus penipuan dengan total kerugian 436 juta dolar [2]. Di Indonesia sendiri sepanjang tahun 2021, Kementerian Kominfo mengumumkan bahwa pihaknya telah menerima laporan aduan transaksi daring sebanyak 115.756, baik yang terjadi di *e-commerce* maupun media sosial [3].

Layanan pengaduan transaksi daring semacam itu tidak bisa dimanfaatkan secara langsung oleh pengguna telepon untuk segera mengidentifikasi dan memblokir panggilan telepon masuk atau mengidentifikasi nomor rekening yang terindikasi penipuan. Pengguna telepon bisa saja tertipu meskipun nomor penipu tersebut telah dilaporkan sebelumnya. Aplikasi identifikasi panggilan seperti Getcontact mampu mengidentifikasi nomor telepon yang masuk berdasarkan label nama kontak yang didapat dari pengguna lain dan bisa melakukan blokir otomatis, meskipun hanya terbatas pada panggilan *spam*. Masalah pada aplikasi Getcontact adalah pemberian label pada nomor telepon tidak disertai bukti sehingga sangat mudah untuk memberikan label palsu. Tambahan lagi, Getcontact memiliki akses terhadap data pribadi yang cukup mengkhawatirkan seperti membaca dan mengunggah daftar kontak, membaca panggilan keluar dan masuk, sampai membaca isi SMS [4].

Oleh karena itu, penelitian ini dilakukan untuk menghasilkan suatu aplikasi *mobile* untuk berbagi laporan penipuan yang disertai bukti, menelusuri riwayat laporan nomor telepon dan rekening bank tertentu yang terindikasi dipakai untuk penipuan, mendeteksi panggilan masuk yang pernah dilaporkan sebelumnya dan memblokirnya secara otomatis. Di sisi lain, aplikasi juga perlu menghargai privasi pengguna yang dicapai dengan membatasi aplikasi agar hanya bisa membaca nomor telepon panggilan masuk, menolak panggilan masuk, dan melakukan pemblokiran terhadap nomor telepon. Selain itu, pengguna diberikan opsi untuk menyembunyikan identitas mereka dari pengguna lain saat membuat laporan pada aplikasi.

2. TINJAUAN PUSTAKA

2.1 Aplikasi *Mobile*

Aplikasi *mobile* adalah aplikasi yang berjalan pada perangkat *mobile* seperti ponsel, tablet, dan jam tangan. Aplikasi *mobile* awalnya hanya ditujukan untuk kebutuhan mendasar seperti layanan telepon dan pengiriman pesan. Akan tetapi, perkembangan *mobile computing* yang begitu pesat mengakibatkan aplikasi *mobile* meluas ke sektor lain seperti hiburan sampai ke layanan pemesanan daring. Aplikasi *mobile* pada saat ini sudah mampu untuk menyediakan komputasi dan layanan yang lebih rumit seperti *Global Positioning System* (GPS) dan layanan berbasis lokasi [5].

Berdasarkan teknologi yang digunakan, aplikasi *mobile* terbagi ke dalam tiga bagian:

1. *Native application* yang dibuat khusus untuk sistem operasi (*mobile* OS) tertentu. Karena dibuat secara khusus untuk sistem operasi tertentu, aplikasi jenis ini cenderung berkinerja lebih baik dan memiliki lebih banyak akses terhadap perangkat keras pengguna. Akan tetapi, *native application* tidak bisa dijalankan pada sistem operasi lain.
2. *Web application* yang diakses melalui peramban *web* pada perangkat *mobile*. *Web application* bukan aplikasi yang berdiri sendiri untuk perangkat *mobile* melainkan merupakan situs *web* dengan UI yang responsif terhadap tampilan *mobile*. Karena diakses melalui peramban *web*, pengalaman pengguna bisa berbeda-beda tergantung pada peramban yang digunakan.
3. *Hybrid application* yang merupakan gabungan dari *web application* dan *native application*. *Hybrid application* berjalan melalui peramban akan tetapi mampu mengakses berbagai fitur-fitur yang terdapat pada *native application*.

2.2 *Rapid Application Development*

Rapid Application Development (RAD) adalah metodologi untuk mempercepat pengembangan sistem informasi dan menghasilkan sistem informasi yang fungsional. Tujuan utama dari semua pendekatan RAD adalah untuk memangkas waktu pengembangan. Karena tiap tahapan merupakan proses yang berkelanjutan, RAD memungkinkan tim pengembang untuk melakukan perubahan yang

diperlukan secara cepat saat terjadi perubahan yang tidak diperkirakan sebelumnya. RAD terdiri dari empat tahapan [6], yaitu:

1. Perencanaan persyaratan
   Perencanaan persyaratan menggabungkan elemen-elemen dari tahap perencanaan sistem dan analisis sistem pada metodologi *System Development Life Cycle* (SDLC). Pengguna, manajer dan staf TI berdiskusi dan menyepakati ruang lingkup, batasan, dan kebutuhan sistem. Tahapan ini selesai apabila kesepakatan telah tercapai.
2. Desain pengguna
   Di dalam tahapan desain, pengguna berinteraksi dengan analis sistem dan membangun model dan *prototype* yang mewakili semua masukan, proses, dan keluaran sistem. Desain pengguna merupakan proses berkelanjutan dan interaktif yang memberikan pengguna kesempatan untuk memahami, memodifikasi, dan menyetujui model sistem yang memenuhi kebutuhan mereka.
3. Konstruksi sistem
   Tahapan konstruksi berfokus pada pengembangan program dan aplikasi. Akan tetapi pada tahapan ini, perubahan atau perkembangan masih bisa diusulkan dan diterapkan.
4. *Cutover*
   Fase *cutover* menyerupai tahapan akhir pada fase implementasi di dalam SDLC, seperti konversi data, pengujian, dan pelatihan pengguna.

2.3 Kenyamanan Bertelepon

Gangguan keamanan seperti SMS penipuan (*scam*), SMS sampah (*spam*), virus, telepon penipuan (*scam phone call*) merupakan contoh kecil dari banyaknya gangguan keamanan bertelepon di Indonesia. Tercatat sebanyak 91% dari 100 responden mengatakan bahwa dirinya pernah mengalami gangguan keamanan [7]. Cara-cara yang bisa dilakukan untuk melindungi diri dari gangguan keamanan yaitu [8]:

1. Berhati-hati apabila ditelepon oleh nomor yang dicurigai
2. Menggunakan aplikasi pihak ketiga dalam membantu mengelola panggilan masuk
3. Menyadari akan pentingnya keamanan privasi untuk melawan para penipu

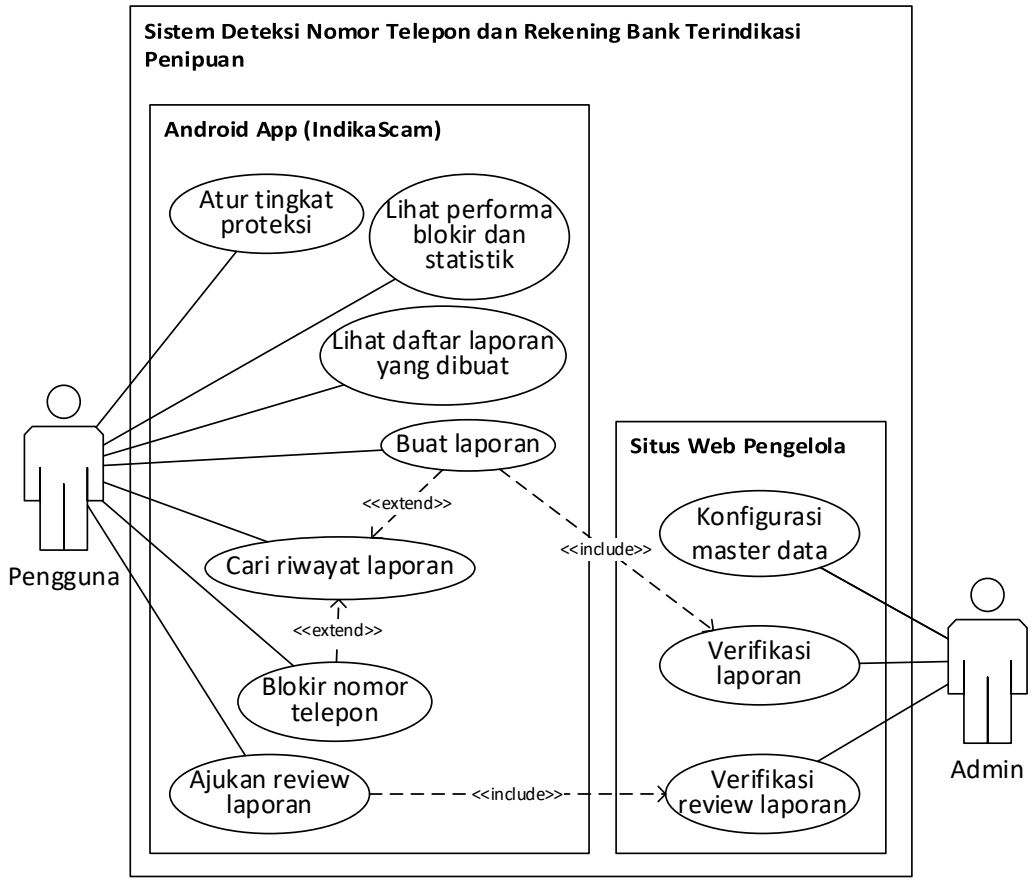
3. METODE PENELITIAN

Metodologi penelitian yang digunakan mengikuti metode pengembangan sistem *Rapid Application Development* (RAD).

3.1 Perencanaan Persyaratan

Perencanaan persyaratan sistem dilakukan dengan observasi aplikasi sejenis, analisis kebutuhan fungsional, dan analisis kebutuhan non fungsional.

a. Observasi aplikasi sejenis
   Observasi dilakukan pada aplikasi Getcontact dan Kredibel dalam rangka merencanakan persyaratan sistem (*system requirements*). Dengan mempelajari fitur-fitur, kelebihan, dan kekurangan pada kedua aplikasi tersebut, maka aplikasi baru perlu dibangun dengan berdasarkan pada praktik-praktik yang dinilai sudah baik. Selain itu, aplikasi baru diupayakan untuk menghindari kekurangan yang terdapat pada kedua aplikasi tersebut, misalnya: pelaporan penipuan harus berdasarkan bukti-bukti yang dimiliki, pencarian nomor telepon atau rekening bank untuk melihat riwayat pelaporan penipuan, dan pemblokiran secara otomatis sesuai keinginan pengguna.
b. Analisis persyaratan fungsional
   Persyaratan fungsional dianalisis dengan menggambarkan diagram *use case* (Gambar 1) yang dilengkapi dengan deskripsi *use case* (Tabel 1).



Gambar 1. Diagram *Use Case*

Tabel 1. Deskripsi *Use Case* Sederhana

No	*Use Case*	Deskripsi
1	Atur tingkat proteksi	Pengguna mengatur tingkat proteksi dengan pemblokiran panggilan secara otomatis. Tingkat proteksi mencakup: Tinggi (blokir otomatis terhadap semua jenis gangguan), Sedang (blokir otomatis terhadap jenis gangguan penipuan), dan Rendah (tanpa blokir otomatis terhadap jenis gangguan).
2	Lihat performa blokir dan statistik	Pengguna melihat kinerja dan statistik blokir otomatis selama menggunakan aplikasi.
3	Lihat daftar laporan yang dibuat	Pengguna melihat seluruh laporan (dan termasuk pengajuan *review*) yang pernah dibuat sebelumnya oleh pengguna.
4	Buat laporan	Pengguna melaporkan gangguan yang dialaminya kepada sistem.
5	Cari riwayat laporan	Pengguna mencari riwayat laporan nomor telepon atau rekening.
6	Blokir nomor telepon	Pengguna melakukan blokir terhadap nomor telepon secara pribadi.
7	Ajukan *review* laporan	Pengguna mengajukan *review* terhadap laporan yang diajukan terhadap nomor telepon atau rekening yang dimiliki pengguna.
8	Konfigurasi data master	Admin menambah dan mengubah data master, yaitu: data bank, *platform* tempat terjadinya penipuan, produk yang digunakan untuk menipu, dan tipe laporan.

9	Verifikasi Laporan	Admin memverifikasi laporan pengguna.
10	Verifikasi *review* laporan	Admin memverifikasi pengajuan *review* dari pengguna terhadap laporan yang ditujukan kepadanya.

c. Persyaratan Non fungsional
Persyaratan non fungsional dianalisis dengan kerangka PIECES (Tabel 2).

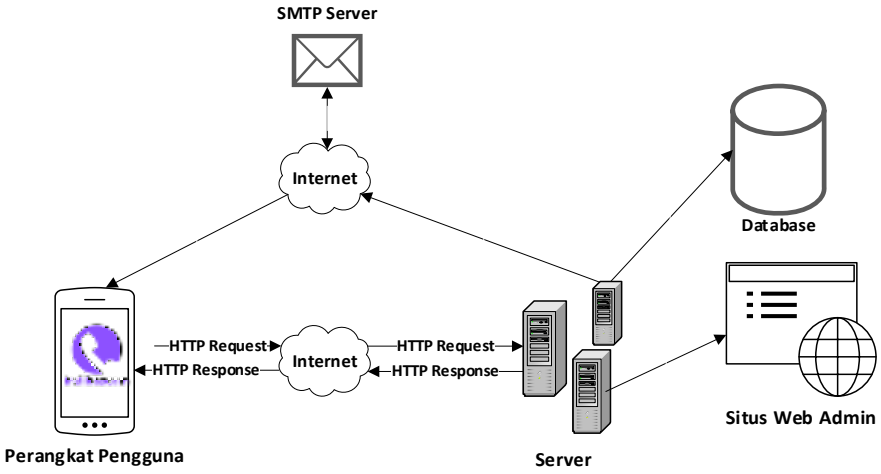
Tabel 2. Persyaratan Non fungsional

No	Kriteria	Persyaratan
1	*Performance*	Sistem harus bisa mengidentifikasi ancaman dan memblokirnya secara otomatis dalam waktu kurang dari 5 detik, yang merupakan durasi layanan penyaringan telepon dari sistem operasi *Android*.
2	*Information*	Informasi ancaman panggilan diperoleh tepat waktu dan secara otomatis.
3	*Efficiency*	Proteksi dengan blokir panggilan masuk dilakukan secara otomatis sesuai pengaturan pengguna.
4	*Control*	• Sistem dapat membedakan pengguna yang masuk ke dalam sistem menggunakan prosedur otentikasi, opsi anonimitas untuk melindungi kerahasiaan identitas pelapor. • Sistem tidak memiliki pengendalian atas data pribadi pengguna seperti daftar kontak dan SMS.
5	*Economy*	-
6	*Service*	Layanan tersedia bagi pengguna selama pengguna memiliki telepon genggam dengan sistem operasi *Android* dan koneksi internet.

3.2 Rancangan Sistem

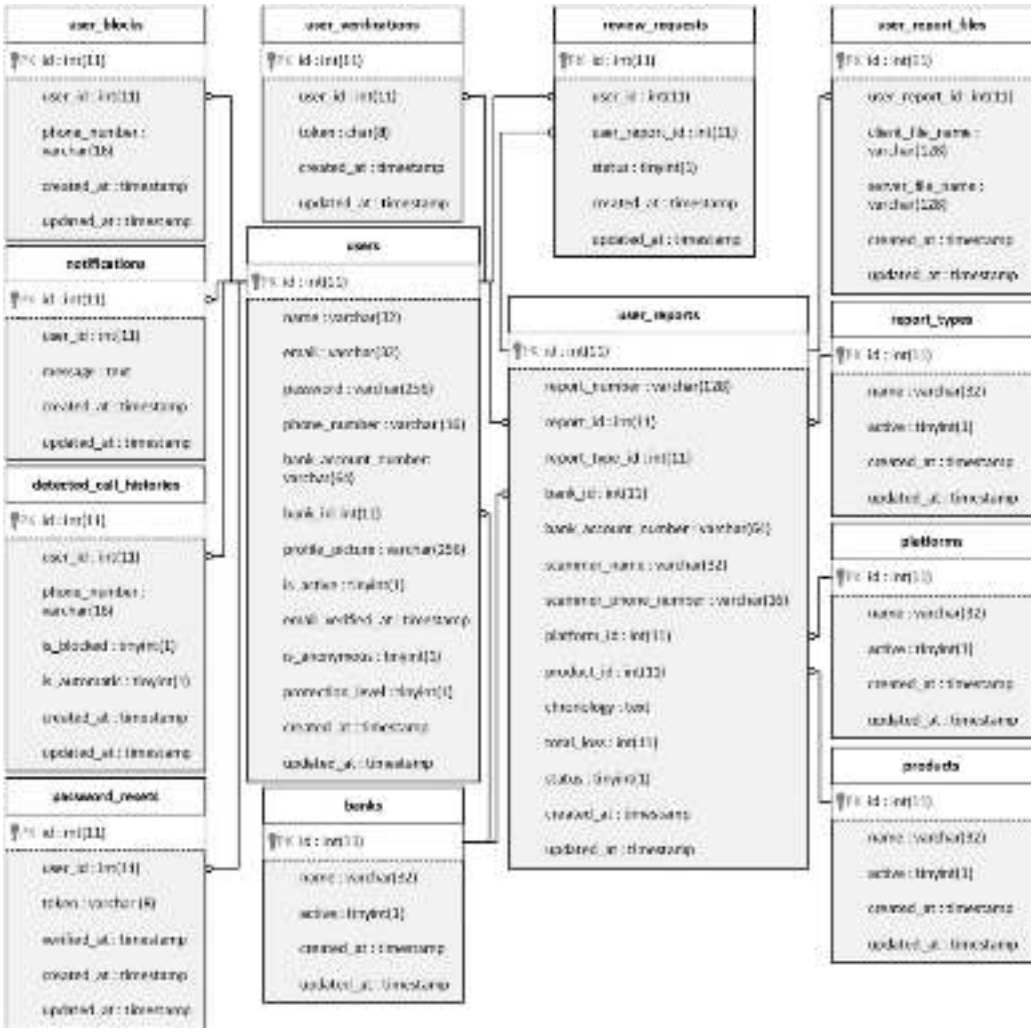
Rancangan fisik sistem pada Gambar 2 dapat dijelaskan sebagai berikut:

1. Perangkat Pengguna: Aplikasi *mobile* dibangun dengan basis Android karena *mobile* OS ini menguasai 91% pangsa pasar Indonesia [9]. Dalam menggunakan aplikasi, pengguna memerlukan koneksi internet untuk melakukan pendaftaran akun, verifikasi *email*, *login*, lapor, cari nomor telepon atau rekening, blokir panggilan masuk dan sebagainya. Aplikasi Android dikembangkan dengan bahasa pemrograman Kotlin dan IDE Android Studio.
2. *Simple Mail Transfer Protocol* (SMTP) *Server*: Pengiriman email melalui SMTP untuk melakukan verifikasi email pengguna saat pendaftaran akun dan perubahan *password*.
3. Situs *web* admin: Situs *web* admin dikembangkan dengan PHP *framework* Laravel 7.
4. *Server*: *Server* digunakan untuk menjalankan API yang akan mendengarkan permintaan dari *client* yang berupa pengguna aplikasi *mobile* dan situs *web* admin.
5. Basis data: Basis data yang digunakan adalah basis data relasional.



Gambar 2. Ilustrasi Rancangan Fisik Sistem

Rancangan basis data relasional yang dibutuhkan sistem adalah sebagai berikut.



Gambar 3. Struktur Relasi tabel

3.3 Konstruksi

Konstruksi sistem dapat dirincikan sebagai berikut:

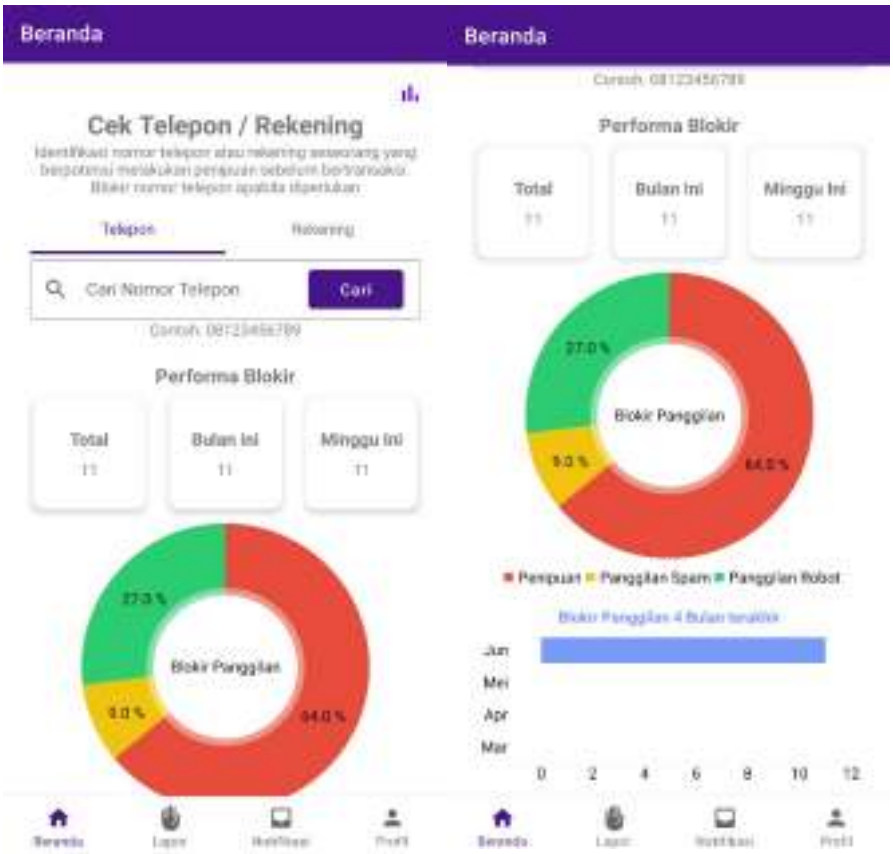
1. Perangkat Pengguna: Aplikasi Android dibangun dengan bahasa pemrograman Kotlin dan IDE Android Studio.
2. *Simple Mail Transfer Protocol* (SMTP) *Server*: Pengiriman *email* melalui SMTP Office 365 dibangun dengan komponen *Swift Mailer* yang merupakan *library* pengiriman *email* aplikasi PHP.
3. Situs *web* admin: Situs *web* admin dan RESTful API dibangun dengan PHP *framework* Laravel 7.
4. *Server*: *Server* yang digunakan adalah Apache HTTP Server. Pada lingkungan pengembangan, Apache HTTP Server dijalankan pada komputer dengan OS Windows 10 menggunakan XAMPP. Agar bisa diakses secara eksternal, digunakan aplikasi Ngrok untuk mengekspos *port* server lokal.
5. Basis data: Basis data dikelola dengan DBMS MariaDB.

4. HASIL DAN PEMBAHASAN

4.1 Hasil

Hasil sistem (aplikasi Android dan situs web) dijelaskan sebagai berikut.

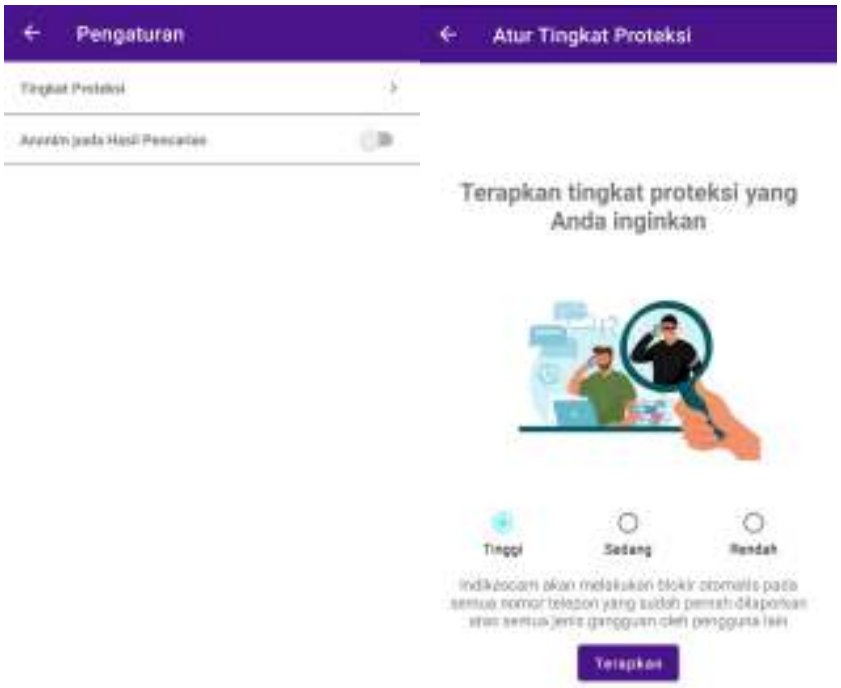
1. Aplikasi *mobile*
   a. Layar Beranda (Gambar 4)
   Pada halaman ini tersedia sebuah ikon batang pada sudut kanan atas untuk membuka kompilasi data laporan, lalu tersedia area untuk melakukan pencarian terhadap nomor telepon atau nomor rekening. Performa blokir memperlihatkan performa aplikasi menghalau panggilan-panggilan yang berpotensi mengganggu yang berasal dari laporan pengguna lain. Bagan donat mendeskripsikan jenis-jenis gangguan yang berhasil diblokir; bagan batang memperlihatkan jumlah blokir empat bulan terakhir.



Gambar 4. Layar Beranda

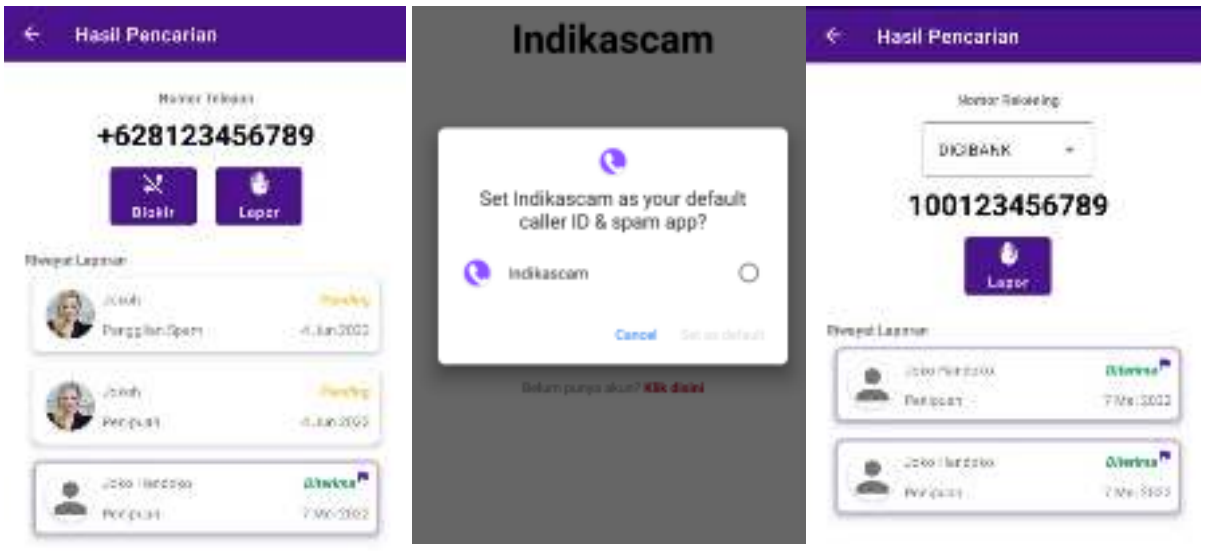
   b. Layar Pengaturan (Gambar 5)
   Pengguna aplikasi bisa mengatur tingkat proteksi (sesuai penjelasan pada Tabel 1). Selain itu, pengguna bisa mengatur anonimitas akunnya pada tampilan hasil pencarian. Dengan opsi anonim,

nama lengkap dan foto profil pengguna selaku pelapor akan disembunyikan dari laporan tersebut saat ditampilkan pada hasil pencarian yang dilakukan pengguna lain.



Gambar 5. Layar Pengaturan

c. Layar Hasil Pencarian (Gambar 6)
Pengguna bisa melihat hasil pencarian pada nomor telepon atau nomor rekening. Pada hasil pencarian nomor telepon, pengguna bisa melakukan blokir pribadi yang akan membuat pengguna tidak menerima panggilan dari nomor telepon yang dicari dengan cara mengetuk tombol "Blokir".

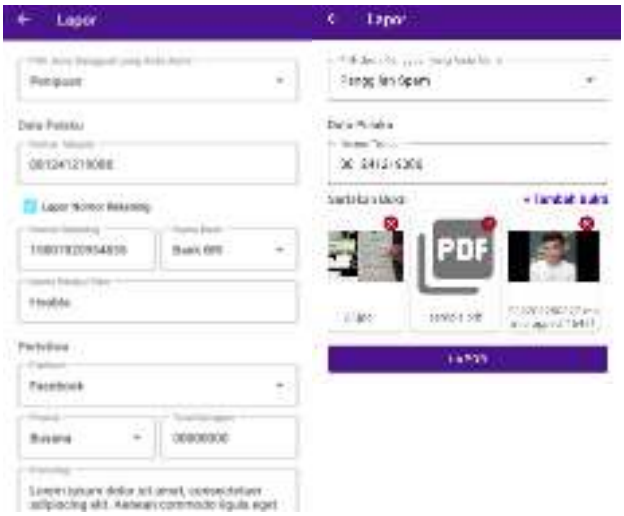


Gambar 6. Layar Hasil Pencarian

Aplikasi menggunakan layanan Android *Call Screening Service* untuk blokir panggilan masuk, baik yang berasal dari laporan pengguna maupun blokir pribadi. Layanan ini membutuhkan aplikasi terdaftar di perangkat untuk penyaringan telepon (tampilan permintaan *Caller & ID Spam* pada Gambar 6). Pada hasil pencarian nomor rekening terdapat kolom untuk menampilkan nama

bank atau memilih bank jika nomor rekening yang dicari terdapat lebih dari 1 jenis bank. Tombol "Lapor" untuk melaporkan nomor telepon atau nomor rekening sesuai dengan hasil pencarian pengguna yang akan dijelaskan pada tampilan lapor. Pengguna juga bisa mengajukan permohonan tinjauan ulang pada riwayat laporan yang statusnya "Diterima".

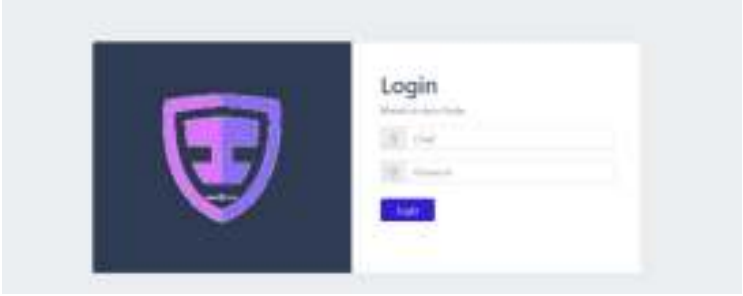
d. Layar Lapor (Gambar 7)
Pengguna dapat melaporkan gangguan yang dialaminya.



Gambar 7. Halaman Lapor

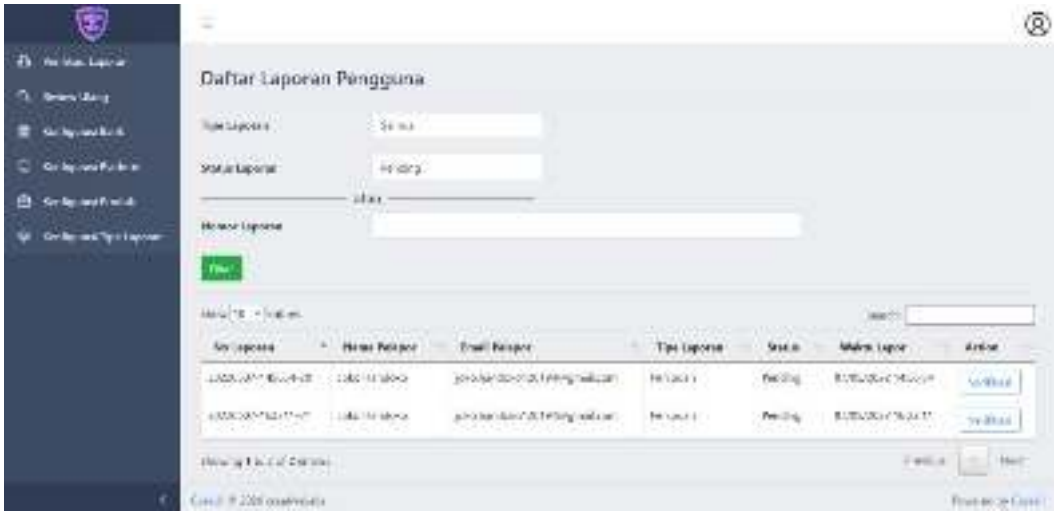
Setelah memilih jenis gangguan (penipuan, panggilan *spam*, atau panggilan robot), halaman akan menampilkan formulir yang sesuai dengan jenis gangguan tersebut. Bukti yang disertakan mencakup gambar dan PDF. Jika perlu, tiap bukti gambar bisa diperbesar dengan cara mengetuk gambar bukti dan tiap bukti PDF bisa dibuka dengan aplikasi ketiga untuk membaca konten PDF. Jika sistem sudah memiliki riwayat pelaporan terhadap suatu nomor dengan status diterima, sistem akan menolak laporan tersebut.

2. Situs *web* admin
   a. Halaman *Login* (Gambar 8)
   Admin harus melakukan *login* ke situs *web* dengan memasukkan *email* dan *password* untuk menggunakan berbagai fitur situs *web*.



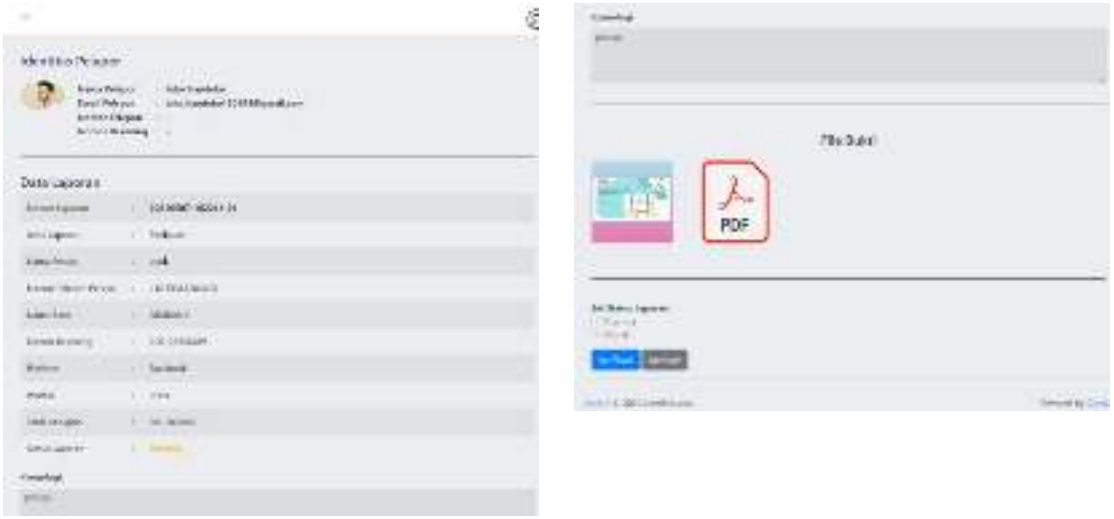
Gambar 8. Halaman *Login*

   b. Halaman Daftar Laporan Pengguna (Gambar 9)
   Admin bisa melihat daftar laporan pengguna yang masuk ke dalam aplikasi. Admin dapat menyaring data laporan berdasarkan tipe laporan, status laporan dan nomor laporan. Laporan pengguna memiliki *pagination* untuk pemberian halaman pada tabel data. Admin bisa masuk ke halaman verifikasi laporan pengguna dengan menekan tombol Verifikasi yang terdapat di sisi kanan setiap baris.



Gambar 9. Halaman Daftar Laporan Pengguna

c. Halaman Verifikasi Laporan Pengguna (Gambar 10)
Admin bisa melihat detail laporan pengguna dan melakukan verifikasi terhadap laporan tersebut dengan memberikan status Diterima atau Ditolak. Hasil verifikasi laporan diteruskan dengan mengirimkan notifikasi ke aplikasi Android pengguna. Menerima satu laporan berarti menerima laporan lain yang ditujukan untuk nomor telepon dan nomor rekening yang sama secara otomatis.



Gambar 10. Halaman Verifikasi Laporan Pengguna

d. Halaman Konfigurasi Data Master
Admin dapat mengelola konfigurasi data dalam tiga kategori: bank, *platform* tempat penipuan terjadi, produk yang digunakan untuk menipu, dan tipe laporan. Gambar 11 memperlihatkan halaman konfigurasi tipe laporan.

Gambar 11. Halaman Konfigurasi Tipe Laporan

4.2 Pembahasan

Untuk memastikan fungsionalitas sistem sesuai dengan persyaratan, maka dilakukan pengujian *black box* pada aplikasi Android dan situs *web* admin. Pengujian mencakup pengujian tampilan, fungsi aplikasi, dan keberhasilan integrasi dengan layanan pihak ketiga (seperti Office 365 untuk pengiriman *email*).

Pengujian aplikasi Android dilakukan dengan emulator berspesifikasi Android 10, RAM 1536MB, dan 2 CPU *core*. Terdapat 37 skenario/kasus uji sebagai berikut (contoh rincian uji di Tabel 3):

1. Uji masukan nama lengkap saat daftar akun
2. Uji masukan *email* saat daftar akun
3. Uji masukan kata sandi saat daftar akun
4. Uji masukan konfirmasi kata sandi saat daftar akun
5. Uji verifikasi OTP saat daftar akun
6. Uji masukan email saat ubah kata sandi
7. Uji masukan email saat ubah kata sandi
8. Uji verifikasi OTP saat ubah kata sandi
9. Uji masukan kata sandi baru
10. Uji masukan konfirmasi kata sandi baru
11. Uji masukan *email* saat *login*
12. Uji masukan kata sandi saat *login*
13. Uji panduan
14. Uji masukan kolom gangguan saat proses lapor
15. Uji masukan nama pelaku saat proses lapor
16. Uji masukan nama bank saat proses lapor
17. Uji masukan nomor rekening saat proses lapor
18. Uji masukan *platform* saat proses lapor
19. Uji masukan produk saat proses lapor
20. Uji masukan kronologi saat proses lapor
21. Uji masukan total kerugian saat proses lapor
22. Uji masukan *file* bukti saat proses lapor
23. Uji masukan foto profil saat ubah profil
24. Uji masukan nama lengkap saat ubah profil
25. Uji masukan nomor telepon saat ubah profil
26. Uji masukan nomor rekening saat ubah profil
27. Uji masukan nama bank saat ubah profil
28. Uji pengaturan
29. Uji pencarian nomor telepon atau rekening
30. Uji blokir dan buka blokir
31. Uji pengajuan *review* ulang
32. Uji daftar dan detail laporan pengguna
33. Uji daftar pengajuan *review* ulang pengguna
34. Uji keluar
35. Uji blokir panggilan masuk
36. Uji info grafik performa blokir
37. Uji notifikasi dan daftar blokir

Terdapat satu kasus uji yang gagal dilewati aplikasi Android, yakni uji nomor 35. Kegagalan ini berkaitan dengan blokir panggilan setelah emulator di-*restart*. Penambahan RAM menjadi 4GB tidak memperbaiki kegagalan; aplikasi tetap gagal memblokir setelah emulator *restart*. Kegagalan ini diduga karena proses *bootup* yang butuh waktu dan sumber daya emulator sedangkan Android membatasi durasi blokir hanya lima detik sejak panggilan masuk terdeteksi. Oleh karena itu, lingkungan pengujian dipindahkan ke ponsel Android Samsung Note 9 dengan RAM 6GB dan ponsel Realme 5 dengan RAM 3GB. Pada kedua ponsel ini, aplikasi berhasil melakukan pemblokiran dengan baik setelah ponsel di-*restart*. Dengan demikian dapat disimpulkan bahwa kegagalan dalam dua kasus uji hanya terjadi pada emulator namun tidak demikian jika aplikasi berjalan pada perangkat ponsel sesungguhnya. Rincian uji nomor 35 ditampilkan di Tabel 3.

Tabel 3. Rincian Uji Nomor 35 (Blokir Panggilan Masuk)

No	Deskripsi Uji	Data Tes	Hasil	Hasil yang Diharapkan	Status
1	Tingkat proteksi rendah	Nomor 0812345* 1 laporan diterima	App tak memblokir panggilan 0812345*	App tak memblokir panggilan 0812345*	Berhasil
2	Tingkat proteksi sedang	Nomor 0812345* 1 laporan penipuan	App memblokir panggilan 0812345*	App memblokir panggilan 0812345*	Berhasil
3	Tingkat proteksi tinggi	Nomor +6286531** 1 laporan panggilan spam	App memblokir panggilan +6286531**	App memblokir panggilan +6286531**	Berhasil
4	Blokir pribadi	Blokir pribadi nomor 086549873**	App memblokir panggilan 086549873**	App memblokir panggilan 086549873**	Berhasil
5	Buka blokir pribadi	Buka blokir pribadi nomor 08654987*	App tak memblokir panggilan 08654987*	App tak memblokir panggilan 08654987*	Berhasil
6	Aplikasi sedang dibuka	Nomor +6286531** 1 laporan panggilan spam	App memblokir panggilan +6286531**	App memblokir panggilan +6286531**	Berhasil
7	Aplikasi berjalan di *background*	Nomor +6286531** 1 laporan panggilan spam	App memblokir panggilan +6286531**	App memblokir panggilan +6286531**	Berhasil
8	Aplikasi di-*destroy / clear memory*	Nomor +6286531** 1 laporan panggilan spam	App memblokir panggilan +6286531**	App memblokir panggilan +6286531**	Berhasil
9	Emulator RAM 1536MB di-*restart*	Nomor +6286531** 1 laporan panggilan spam	App gagal memblokir panggilan +6286531**	App memblokir panggilan +6286531**	Gagal
10	Emulator RAM 4GB di-*restart*	Nomor +6286531** 1 laporan panggilan spam	App gagal memblokir panggilan +6286531**	App memblokir panggilan +6286531**	Gagal
11	App dijalankan di Galaxy Note 9, RAM 6GB, Android 10	Nomor +62853612* 1 laporan panggilan spam	App memblokir panggilan +62853612*	App memblokir panggilan +62853612*	Berhasil
12	App dijalankan di Realme 5, RAM 3GB, Android 10	Nomor +62853612* 1 laporan panggilan spam	App memblokir panggilan +62853612*	App memblokir panggilan +62853612*	Berhasil

Pengujian situs *web* admin dilakukan di peramban Google Chrome. Keseluruhan 17 skenario/kasus uji pada situs *web* admin sebagai berikut telah dapat dilewati dengan baik (contoh rincian uji di Tabel 4):

1. Uji *login* admin
2. Uji daftar laporan pengguna
3. Uji verifikasi laporan pengguna
4. Uji daftar *review* ulang
5. Uji verifikasi *review* ulang
6. Uji konfigurasi bank
7. Uji tambah bank
8. Uji ubah bank

9. Uji konfigurasi *platform*
10. Uji tambah *platform*
11. Uji ubah *platform*
12. Uji konfigurasi produk
13. Uji tambah produk
14. Uji ubah produk
15. Uji konfigurasi tipe laporan
16. Uji tambah tipe laporan
17. Uji ubah tipe laporan

Tabel 4. Uji Verifikasi Laporan Pengguna

No	Deskripsi Uji	Data Tes	Hasil	Hasil yang Diharapkan	Status
1	Menerima Laporan Pengguna	Set status laporan menjadi "Diterima"	Sistem mengubah status laporan menjadi "Diterima"	Sistem mengubah status laporan menjadi "Diterima"	Berhasil
2	Menolak Laporan Pengguna	Set status laporan menjadi "Ditolak"	Sistem mengubah status laporan menjadi "Ditolak"	Sistem mengubah status laporan menjadi "Ditolak"	Berhasil

5. KESIMPULAN

Berdasarkan hasil pengembangan dan pengujian aplikasi Android maupun situs *web* admin, dapat ditarik beberapa kesimpulan, antara lain:

1. Aplikasi Android yang dihasilkan telah berfungsi dengan baik dalam mendeteksi nomor telepon dan rekening bank yang terindikasi penipuan. Aplikasi Android dapat dimanfaatkan pengguna untuk memblokir panggilan masuk yang terindikasi pernah melakukan penipuan maupun gangguan telepon lainnya berdasarkan laporan pengguna lain. Pengguna juga dapat mengecek nomor rekening sebelum bertransaksi sehingga bisa lebih berhati-hati. Dengan demikian, pengguna akan lebih nyaman dalam menggunakan ponsel.
2. Aplikasi mampu meminimalkan kebutuhan informasi pada perangkat pengguna demi menjaga privasi pengguna dengan hanya menggunakan satu jenis izin (*permission*) yaitu *Caller ID & Spam* yang tidak dapat mengakses informasi-informasi di luar dari panggilan.

6. SARAN

Beberapa saran untuk pengembangan lanjutan ke depannya yaitu:

1. Menambah fitur verifikasi nomor telepon pengguna untuk digunakan saat pendaftaran di awal menggunakan aplikasi Android.
2. Mengembangkan verifikasi laporan pengguna yang masih dilakukan secara manual melalui situs *web* admin agar pelaksanaan verifikasi lebih efisien.
3. Sistem dilengkapi dengan fitur yang menciptakan aliran pendapatan bagi pengelola, seperti: pengenaan biaya terhadap layanan tingkat proteksi tinggi, kerja sama dengan para pelaku usaha, komersialisasi layanan API, dan iklan pada aplikasi Android.

DAFTAR PUSTAKA

[1] S. Palinggi, S. Palelleng, and L. R. Allolinggi, "Peningkatan Rasio Kejahatan Cyber dengan Pola Interaksi Sosio Engineering Pada Periode Akhir Era Society 4.0 di Indonesia," JIDS, vol. 4, no. 1, pp.145-163, Feb. 2020. doi: 10.38043/jids.v4i1.2314.

[2] Federal Trade Commision, "CSN Annual Data Book 2021," 2022. https://www.ftc.gov/system/files/ftc_gov/pdf/CSN%20Annual%20Data%20Book%202021%20Final%20PDF.pdf (accessed Apr. 28, 2022)

[3] CNN Indonesia, "Kominfo Catat Kasus Penipuan Online Terbanyak: Jualan Online," Okt. 15, 2021. https://www.cnnindonesia.com/teknologi/20211015085350-185-708099/kominfo-catat-kasus-penipuan-online-terbanyak-jualan-online (accessed Apr. 28, 2022)

[4] Getcontact, "Kebijakan Privasi Getcontact," Jun, 2021. https://www.getcontact.com/privacy (accessed Apr. 28, 2022)

[5] L. Valdellon, "What Are the Different Types of Mobile Apps? And How Do You Choose?" Nov. 2020, https://clevertap.com/blog/types-of-mobile-apps (accessed 1 May, 2022)

[6] S. Tilley and H. Rosenblatt, *Systems Analysis and Design*, 11 ed., Boston: Cengage Learning, 2017.

[7] R. Akraman, C. Candiwan, and Y. Priyadi, "Pengukuran Kesadaran Keamanan Informasi dan Privasi pada Pengguna Smartphone Android di Indonesia," *JSINBIS (Jurnal Sistem Informasi Bisnis)*, vol. 8, no. 2, pp. 115-122, Oct. 2018. doi: 10.21456/vol8iss2pp115-122.

[8] M. F. Mubarak, S. Yahya, and A. F. A. Shaazi, "A Review of Phone Scam Activities in Malaysia," *2019 IEEE 9th International Conference on System Engineering and Technology (ICSET)*, 2019, pp. 441–446. doi: 10.1109/ICSEngT.2019.8906491.

[9] Statcounter, "Mobile Operating System Market Share Indonesia," 2022. https://gs.statcounter.com/os-market-share/mobile/indonesia (accessed Mar. 21, 2022)