TESIS

SIGNATURE VERIFICATION BERBASIS DEX CRC DAN ALGORITMA BLAKE2 UNTUK MENCEGAH SERANGAN APPLICATION REPACKAGING PADA APLIKASI ANDROID

Signature Verification Based On Dex Crc And Blake2 Algorithm To Prevent Application Repackaging Attack In Android Application

ILHAM

D082201007

PROGRAM STUDI S2 TEKNIK INFOMATIKA

DEPARTEMEN TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2024

PENGAJUAN TESIS

SIGNATURE VERIFICATION BERBASIS DEX CRC DAN ALGORITMA BLAKE2 UNTUK MENCEGAH SERANGAN APPLICATION REPACKAGING PADA APLIKASI ANDROID

Tesis

Sebagai Salah Satu Syarat untuk Mencapai Gelar Magister

Program Studi Teknik Informatika

Disusun dan diajukan oleh

ILHAM

D082201007

Kepada

FAKULTAS TEKNIK

UNIVERSITAS HASANUDDIN

GOWA

2024

TESIS

SIGNATURE VERIFICATION BERBASIS DEX CRC DAN ALGORITMA BLAKE2 UNTUK MENCEGAH SERANGAN APPLICATION REPACKAGING PADA APLIKASI ANDROID

Ilham
D082201007

Telah dipertahankan di hadapan Panitia Ujian Tesis yang dibentuk dalam rangka penyelesaian studi pada Program Magister Teknik Informatika Fakultas Teknik
Universitas Hasanuddin
Pada tanggal 22 Januari 2024
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

Pembimbing Utama

Dr-Eng.Ir. Muhammad Niswar, ST, M.InfoTech.
NIP. 197309221999031001

Pembimbing Pendamping

Dr. Eng. Ady Wahyudi Paundu, S.T., M.T.
NIP. 197503132009121003

Dekan Fakultas Teknik
Universitas Hasanuddin

Prof. Dr.Eng. Ir. Muhammad Isran Ramli, M.T. IPM., ASEAN.Eng.
NIP. 19730926 200012 1 002

Ketua Program Studi
S2 Teknik Informatika

Dr. Ir. Zahir Zainuddin, M.Sc.
NIP. 19640427 198910 1 002

PERNYATAAN KEASLIAN TESIS DAN PELIMPAHAN HAK CIPTA

Yang bertanda tangan dibawah ini :

Nama : Ilham

Nomor Mahasiswa : D082201007

Program Studi : S2 Teknik Informatika

Dengan ini menyatakan bahwa, tesis berjudul "signature verification berbasis dex crc dan algoritma blake2 untuk mencegah serangan reverse engineering pada aplikasi android" adalah karya saya dengan arahan dari komisi pembimbing (Dr. Eng. Muhammad Niswar, S.T., M.IT dan Dr. Eng. Ady Wahyudi Paundu, S.T., M.T). Karya ilmiah ini belum diajukan dan tidak sedang diajukan dalam bentuk apa pun kepada perguruan tinggi mana pun. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan maupun tidak diterbitkan dari penulis lain telah disebutkan dalam teks dan dicantumkan dalam Daftar Pustaka tesis ini. Sebagian dari isi tesis ini telah dipublikasikan di Jurnal/Prosiding (iJIM (International Journaal of Interactive Mobile Technologies) sebagai artikel dengan judul "Signature Verification Based on Dex CRC and Blake2 Algorithm to Prevent Application Repackaging Attack in Android Application".

Dengan ini saya limpahkan hak cipta dari karya tulis saya berupa tesis ini kepada Universitas Hasanuddin.

Gowa, 10 Januari 2024

Yang Menyatakan

Ilham

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Tuhan Yang maha Esa karena berkat rahmat dan karunia-Nya sehingga tugas akhir yang berjudul "**SIGNATURE VERIFICATION BERBASIS DEX CRC DAN ALGORITMA BLAKE2 UNTUK MENCEGAH SERANGAN APPLICATION REPACKAGING PADA APLIKASI ANDROID**" ini dapat diselesaikan sebagai salah satu syarat dalam menyelesaikan jenjang Strata-2 pada Departement Teknik Informatika Fakultas Teknik Universitas Hasanuddin.

Penulis menyadari bahwa dalam penyusunan dan penulisan laporan tugas akhir ini tidak lepas dari bantuan, bimbingan serta dukungan dari berbagai pihak, dari masa perkuliahan sampai dengan masa penyusunan tugas akhir. Oleh karena itu, penulis dengan senang hati menyampaikan terima kasih kepada :

1. Tuhan Yang Maha Esa atas semua berkat, karunia serta pertolongan-Nya yang tiada batas, yang diberikan kepada penulis disetiap Langkah dalam pembuatan program hingga penulisan skripsi ini;
2. Kedua orang tua penulis, Bapak Abdul Malik, S.Pd dan Ibu Ramlah yang selalu memberikan dukungan, doa, dan semangat serta selalu sabar dalam mendidik penulis sejak kecil;
3. Bapak Dr. Eng. Muhammad Niswar, S.T., M.IT. selaku pembimbing I dan Bapak Dr. Eng. Ady Wahyudi Paundu, S.T., M.T. selaku pembimbing II yang selalu menyedakan waktu, tenaga, pikiran dan perhatian yang luar biasa untuk mengarahkan penulis dalam penyusunan tugas akhir.
4. Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys.,IPM., ASEAN.Eng., Dr. Amil Ahmad Ilham, S.T., M.IT., dan Mukarramah Yusuf, B.Sc. selaku dosen penguji yang telah memberikan saran sehingga laporan tesis ini menjadi lebih baik;
5. Bapak Dr. Ir. Zahir Zainuddin, M.Sc. selaku ketua Departement Magister Teknik Informatika Fakultas Teknik Universitas Hasanuddin atas bimbingannya selama masa perkuliahan.

6. Para sahabat, teman-teman dan kakak-kakak pascasarjana UNHAS yang telah memberikan begitu banyak bantuan selama penelitian dan diskusi *progress* penyusunan tugas akhir;
7. Teman-teman pascasarjana UNHAS Angkatan 2 atas dukungan dan semangat yang diberikan selama ini;
8. Ibu Yuanita serta segenap Staff Departemen Teknik Informatika yang telah membantu penulis;
9. Orang-orang berpengaruh lainnya yang tidak sempat disebutkan oleh penulis.

Akhir kata, penulis berharap semoga Allah SWT. Berkenan membalas segala kebaikan dari semua pihak yang telah banyak membantu. Semoga Tugas Akhir ini dapat memberikan manfaat bagi pengembangan ilmu. Aamiin.

Gowa, 10 Januari 2024

Ilham

ABSTRAK

Ilham. Signature Verification Berbasis Dex CRC dan Algoritma Blake2 Untuk Mencegah Serangan Reverse Engineering Pada Aplikasi Android. (Dibimbing oleh **Muhammad Niswar** dan **Ady Wahyudi Paundu**).

Pesatnya pertumbuhan aplikasi android telah menyebabkan semakin banyaknya kasus kejahatan siber, khususnya serangan *Reverse Engineering* pada aplikasi Android. Salah satu kasus *Reverse Engineering* yang paling sering terjadi adalah *Application Repackaging*, dimana aplikasi diunduh melalui Playstore atau situs resmi kemudian di kemas ulang dengan berbagai macam penambahan / perubahan terhadap fitur aplikasi. Salah satu cara untuk menghindari serangan *Application Repackaging* adalah dengan memeriksa *signature* suatu aplikasi. Namun, *hacker* tetap dapat memanipulasi verifikasi *signature* menggunakan *hook,* yaitu mengganti fungsi asli untuk membaca *signature* dengan fungsi yang telah dimodifikasi dan ditambahkan kedalam aplikasi. Pada penelitian ini dilakukan pengembangan metode verifikasi pada aplikasi Android dengan memanfaatkan Dex CRC dan algoritma Blake2 yang akan ditulis dalam Bahasa C menggunakan Java Native Interface (JNI). Hasil penelitian ini menunjukkan bahwa metode verifikasi menggunakan Dex CRC dan algoritma Blake2 dapat secara efektif melindungi aplikasi Android dari serangan *Application Repackaging* tanpa membebani kinerja aplikasil.

Kata Kunci : Reverse Engineering, Application Repackaging, Blake2, Android Protection

ABSTRACT

Ilham. *Signature Verification Based on Dex CRC and Blake2 Algorithm to Prevent Reverse Engineering Attack in Android Application. (Supervised by* ***Muhammad Niswar*** *dan* ***Ady Wahyudi Paundu****).*

The rapid growth of Android applications has led to more cyber- crime cases, specifically Reverse Engineering attacks on Android apps. One of the most common cases of reverse engineering is Application repackaging, where the application is downloaded via the Play Store or the official website and then repackaged with various additions or changes. One of the ways to avoid Appli- cation Repackaging attacks is to check the signature of an application. However, hackers can manipulate the application by adding a hook, i.e., replacing the orig- inal function for getting signatures with a new modified function in the applica- tion. In this research, the development of a verification method for Android ap- plications is carried out by utilizing Dex CRC and the Blake2 algorithm, which will be written in C using the Java Native Interface (JNI). The results of this study indicate that the verification method using Dex CRC and the Blake2 algorithm can effectively protect Android applications from Application Repackaging at- tacks without burdening application performance.

Keywords: *reverse engineering, application repackaging, blake2, Android protection*

DAFTAR ISI

Daftar Tabel

Daftar Gambar

BAB I
PERMASALAHAN DAN TUJUAN PENELITIAN

I.1 Latar Belakang

Android merupakan salah satu sistem operasi berbasis kernel linux yang dikembangkan oleh google yang banyak digunakan pada smarthphone dan tablet hingga saat ini. Penjualan smartphone android di prediksi mengambil sekitar 70% dari total penjualan smarthone. Hal ini menyebabkan perkembangan aplikasi android sangat pesat (Faruki, et al. 2013). App Repackaging merupakan salah satuk Teknik serangan reverse engineering yang dilakukan untuk memodifikasi atau menyisipkan beraneka ragam kode kedalam aplikasi. Dalam pengembangan aplikasi selalu ada peretas yang berusaha untuk mengeksploitasi / menyerang aplikasi yang dikembangkan dalam bentuk reverse engineering, diantaranya yaitu Application Repackaging, yang merupakan ancaman lazim dan parah dalam dunia pengembangan aplikasi android. Peretas dapat menggunakan tools reverse engineering untuk membongkar sebuah aplikasi dan menggantti, memasukkan, memodifikasi kode sumber ataupun melakukan pembelian palsu. (He, et al. 2019)

Penyerang dapat menyalahgunakan kebijakan *App Repackaging* untuk melakukan kejahatan seperti memodifikasi, membajak, ataupun menyisipkan maleware lalu membagikan ulang aplikasi tersebut melalui aplikasi Market pihak ketiga ataupun melalui website (Jeon, et al. 2021). Berdasarkan penelitian [4] yang sudah ada, 80% sampel malware diimplementasikan melalui App Repackaging. Code Obfuscation merupakan cara yang umum digunakan untuk

melindungi aplikasi dari serangan Reverse Engineering, dimana kode akan sulit dipahami oleh penyerang dan menambah waktu dan updaya untuk meretas. Akan tetapi metode tersebut dapat dilewati dengan menggunakan beberapa alat debugging dan mendapatkan kode asli dan me repack kembali aplikasi. Selain Code Obfuscation, cara ampuh untuk menghindari *App Repackaging* yaitu dengan menggunakan signature verification, dimana aplikasi tidak dapat mengirim atau menerima data dari server apabila terdeteksi terjadi perubahan signature yang merupakan indikasi bahwa aplikasi tersebut telah di modifikasi oleh orang yang tidak dikenal. Akan tetapi metode signature verification juga sudah dapat di lewati menggunakan metode hooking melalui bantuan tools *reverse engineering*, misalnya Ultima, Arm Pro, NP manager dan MT Manager akan tetapi membutuhkan waktu yang lama untuk meretasnya. Semua metode anti *reverse* yang disebutkan diatas masih memiliki beberapa celah yaitu :

1. Penambahan / Penghapusan Iklan

   Pada beberapa aplikasi atau games, biasanya terdapat iklan yang bagi beberapa orang itu mengganggu sehingga hacker membongkar aplikasi terkait kemudian menghapus ataupun menambah iklan yang terdapad pada provider iklan di AndroidManifest.xml

2. Kloning / Penggandaan

   Ketika hacker ingin menggandakan aplikasi dalam sebuah smartphone, hal ini dapat dilakukan dengan mengganti nama paket instalasi pada aplikasi tersebut.

3. *Cheat Game*

Cheat game juga dapat dimasukkan kedalam aplikasi game melalui teknik *reverse engineering*.

Berdasarkan masalah diatas, penulis akan meneliti tentang "Siganature verification berbasis Dex CRC dengan menggunakan Algoritma Blake2" yang diharapkan dapat menangani masalah *App Repackaging*.

I.2 Rumusan Masalah

Berdasarkan latar belakang diatas dapat dirumuskan permasalahan yang akan diselesaikan dalam penelitian ini yaitu :

Bagaimana meningkatkan kemampuan *signature verification* untuk mencegah *App Repackaging* ?

I.3 Tujuan Penelitian

Adapun tujuan dari penelitian ini yaitu untuk menerapkan metode *signature verification* dengan Dex CRC dan algoritma Blake2 untuk meningkatkan kemampuan *signature verification* dan menghindari serangan *App Repackaging* dan menguji performa metode yang dibuat.

I.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah tersedianya suatu metode verifikasi / pengecekan *signature* dari suatu aplikasi menggunakan Dex CRC dan algoritma Blake2 yang diharapkan dapat meningkatkan kemanan aplikasi android.

I.5 Batasan Masalah

Adapun batasan masalah pada penelitian ini adalah :

1. Penelitian ini berfokus pada pengembangan aplikasi android.

2. Metode yang diusulkan diterapkan hanya pada tingkat *source code*, bukan pada aplikasi android yang sudah di *compile*.

I.6 Sistematika penulisan

Untuk memberikan gambaran singkat mengenai isi tulisan secara keseluruhan, maka akan diuraikan beberapa tahapan dari penilisan secara sistematis, yaitu :

BAB I PENDAHULUAN

Bab ini berisi penjelasan tentang latar belakang yang menjabarkan alasan dilakukannya penelitian terkait *Android signature verification* berdasarkan peluang penelitian dan uraian penelitian awal tentang *Android signature verification* yang dilakukan, terkait rumusan masalah, tujuan, manfaat, ruang lingkup serta sistematika penulisan penelitian dibahas pada bagian ini.

BAB II TINJAUAN PUSTAKA

Bab ini memuat tinjauan teori dan konsep dasar dari penelitian yang akan dilakukan berhubungan dengan *reverse engineering, app repackaging, algoritma blake2, java, JNI* dan beberapa landasan teori lainnya. Diuraikan pula tentang tinjauan Pustaka yang merupakan penjelasan tentang hasil-hasil penelitian sebelumnya yang berkaitan dengan penelitian yang dilakukan. Dalam bab ini juga diuraikan tentang kerangka pemikiran yang merupakan penjelasan tentang kerangka berpikir untuk memecahkan masalah yang sedang diteliti, termasuk menguraikan objek penelitian *state of the art* dari beberapa penelitian terkait

BAB III METODOLOGI PENELITIAN

Bab ini berisi tentang tahapan penelitian, waktu dan lokasi penelitian, instrumen penelitian, tahap persiapan, gambaran umum sistem, scenario pengujian, dan analisis perform.

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi hasil dari penelitian ini yaitu hasil evaluasi verifikasi *signature* menggunakan *Dex CRC* dan algoritma *Blake2* pada aplikasi android terhadap serangan *reverse engineering* serta evaluasi performa dari metode tersebut.

BAB V KESIMPULAN DAN SARAN

Bab in merupakan bab akhir yang berisi kesimpulan, implikasi, keterbatasan penelitian yang telah dilakukan, serta saran untuk penelitian selanjutnya.

BAB II
TINJAUAN PUSTAKA

2.1 Kajian Pustaka

2.1.1 Reverse Engineering

Reverse engineering merupakan proses pengekstrakan alur kerja, pengetahuan, atau garis besar desain dari apapun yang telah dikerjakan. Konsep ini telah ada sejak lama sebelum computer modern di temukan bahkan mungkin berasal dari era revolusi industry. *Reverse engineering* mirip dengan penelitian ilmiah dimana peneliti mencoba menyusun cetak biru (*blueprint*) dari atom atau pikiran manusia. *Reverse engineering* biasanya dilakukan untuk mencari atau mendapatkan bahkan menambah pengetahuan, ide ataupun pilosofi desain Ketika beberapa informasi itu tidak tersedia di dalam sebuah kasus tertentu. Didalam beberapa kasus juga informasi yang dimiliki seseorang tidak akan di bagikan ke orang lain. Dikasus lain juga biasanya karena informasi itu hilang atau rusak.

2.1.2 Software Reverse Engineering

Software merupakan sekumpulan program yang dirancang untuk menjalankan fungsi tertentu, dan *software reverse engineering* adalah bagaimana kita melihat kedalam *software* tersebut*, software reverse engineering* merupakan proses virtual yang hanya melibatkan CPU dan pikiran manusia. *Software reverse engineering* membutuhkan kombinasi kemampuan dan pemahaman tentang komputer dan pemahaman pengembangan aplikasi / *software,* akan tetapi seperti kebanyakan subjek, prasyarat yang paling penting adalah rasa penasaran yang sangat kuat dan keinginan untuk belajar. *Software reverse engineering* menggabungkan beberapa keahlian : pemecahan kode, pemecahan teka-teki, pemrograman dan analisis logika.

Berdasarkan perkembangan teknologi, pengetahuan yang didapatkan dari *reverse engineering* dapat digunakan untuk menggunakan kembali suatu objek, analisis keamanan maupun mengetahui cara kerja sesuatu. Salah satu alat dalam *software reverse enginnering* adalah *disassembler.* Alat ini membaca kode biner dan menampilkan setiap instruksi sebagai text yang dapat dipahami. Alat ini biasanya digunakan oleh cracker dan mendapatkan hak akses kedalam suatu system atau

menyebabkan kerusakan yang lain. (Cyril, et al. 2020).

Mode peretasan dengan cara *reverse engineering* kemudian memasukkan kode berbahaya lalu mengemasnya kembali merupakan sebagian besar penyebab peretasan aplikasi Android (Lee and Lim 2014).

Tahapan *software reverse engineering* :

1. Mengumpulkan informasi.

   Mengumpulkan semua kemungkinan informasi tentang program. Sumber-sumber informasi termasuk *source code,* desain, dan dokumentasi untuk pemanggilan *system* dan rute external.
2. Memeriksa informasi.

   Pada tahap ini, seseorang yang melakukan pemulihan memeriksa informasi agar terbiasa dengan sistem suatu *system* dan komponen nya.
3. *Review.*

   Tinjau / *review* informasi atau desain yang telah dipulihkan dan serta verifikasi bahwa hasil pemulihan sudah benar. (Byrne 1991)

2.1.3 Cyclic Redundacy Check (CRC)

Cyclic Redundancy Check, atau sering disebut CRC merupakan teknik untuk mendeteksi kesalahan dalam data digital, tetapi tidak untuk melakukan koreksi ketika terjadi kesalahan. CRC digunakan terutama dalam proses transmisi data. Dalam metode CRC, beberapa pengecekan bit disebut *checksum* yang ditambahkan kedalam pesan saat transmisikan. Penerima dapat menentukan apakah bit yang diterima sesuai dengan data atau tidak, untuk memastikan terjadi kesalahan atau tidak pada saat proses transmisi data. Jika terjadi kesalahan, penerima mengirimkan kembali pesan negatif (*negative acknowledgemen*) kepada pengirim dan meminta agar pesan tersebut dikirim ulang. Teknik ini terkadang juga digunakan pada perangkat penyimpanan data seperti *Disk Drive.* Dalam situasi ini setiap blok pada *disk* akan memiliki pengecekan bit, dan perangkat keras secara otomatis memulai pembacaan ulang blok ketika ada kesalahan yang terdeteksi, atau melaporkan kesalahan ke perangkat lunak. (Alrkiyan 2021)

CRC didasarkan pada aritmatika polinomial, khususnya pada komputasi sisa bagi satu polinomial di GF(2) oleh yang lain. Metode ini seperti memperlakukan pesan sebagai bilangan biner yang sangat besarm dan menghitung sisanya dengan membaginya dengan bilangan prima yang cukup besar secara intuitif yang menghasilkan *checksum* yang andal. Polinomial pada GF(2) adalah polinomial pada variable tunggal x yang koefisiennya 0 atau 1. Penjumlahan dan pengurangan dilakukan dengan menggunakan modulo 2, keduanya sama dengan eksklusif atau operator. Misalnya, jumlah polinomial

$$x^3 + x + 1 \text{ dan}$$

$$x^4 + x^3 + x^2 + x$$

adalah $x^4 + x^2 + 1$. Polynomial ini biasanya tidak ditulis dengan tanda minus, namun juga bisa karena koefisien -1 ekuivalen dengan koefisien 1. Perkalian polynomial semacam itu sangatlah mudah. Produk dari satu koefisien dengan yang lain sama dengan kombinasinya dengan logika dan hasil operator, dan hasil kali parsial dijumlahkan secara eksklusif atau perkalian tidak diperlukan untuk menghitung *checksum* CRC. Pembagian polinomial pada GF(2) dapat dilakukan dengan cara yang hamper sama seperti pembagian panjang polinomial bilangan bulat.

Gambar 2.1 menunjukkan impementasi dasar dari CRC-32 pada perangkat lunak. Protokol CRC-32 menginisialisasi register CRC, kemudian mentransmisikan setiap bit yang paling tidak signifikan pertama kali dan melengkapi *checksum.* Diasumsikan bahwa pesan tersebut terdiri dari jumlah bit yang integral. Program ini menggunakan shift kiri. Hal ini memerlukan pembalikan setiap bit pesan dan memposisikannya di ujung kiri register 32 bit yang dilambangkan dengan “bit” pada program.

```
unsigned int crc32(unsigned char *message) {
   int i, j;
   unsigned int byte, crc;

   i = 0;
   crc = 0xFFFFFFFF;
   while (message[i] != 0) {
      byte = message[i];            // Get next byte.
      byte = reverse(byte);         // 32-bit reversal.
      for (j = 0; j <= 7; j++) {    // Do eight times.
         if ((int)(crc ^ byte) < 0)
              crc = (crc << 1) ^ 0x04C11DB7;
         else crc = crc << 1;
         byte = byte << 1;          // Ready next msg bit.
      }
      i = i + 1;
   }
   return reverse(~crc);
}
```

Gambar 1 *Algoritma CRC-32*

Kode yang ada pada Gambar 1 adalah contoh CRC32 (Peterson 2020).

2.1.4 Android

2.1.4.1 Sistem Operasi Android

Android adalah *software* untuk perangkat mobile yang mencakup sistem operasi, *middleware* dan aplikasi kunci. Pengembangan aplikasi pada platform Android pada umumnya menggunakan bahasa pemrogaman Java. Serangkaaian aplikasi inti Android antara lain klien email, program SMS, kalender, peta, browser, kontak dan lain-lain. Dengan menyediakan sebuah platform pengembangan yang terbuka, pengembang aplikasi Android menawarkan kemampuan untuk membangun aplikasi yang sangat kaya dan inovatif. Pengembang bebas untuk mengambil keuntungan dari perangkat keras, akses informasi, lokasi, menjalankan layanan di latarbelakang, mengatur alaram, menambahkan pemberitahuan ke status bar, dan banyak lagi (Binus 2022).

Android bergantung pada versi linux 2.6 untuk layanan system inti seperti keamanan, manajemen memori, manajemen proses, *network stack,* dan model driver. Kernel juga bertindak sebagai lapisan abstraksi antara *hardware* dan seluruh *software*.

2.1.4.2 Sejarah Sistem Operasi Android

Android adalah sistem operasi untuk telepon seluler yang berbasis linux. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri untuk digunakan oleh bermacam *smartphone.* Awalnya, *google* Inc. membeli Android Inc., pendatang baru yang membuat piranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah open Handset Alliance, konsorsium dari 34 perusahaan piranti keras, piranti lunak dan telekomunikasi, termasuk Google, HTC, Intel, Motrola, Qualcomm, T-mobile dan Nvidia. Terdapat dua jenis distributor system operasi android. Pertama yang mendapat dukungan penuh dari Google atau *Google Mail Service (GMS)* dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan languns Google atau dikenal sebagai *Open Handset Distribution (OHD).*

2007-2008 : Produk Awal : Sekitar September 2007 sebuah studi melaporkan bahwa Google mengajukan hak paten aplikaasai telepon seluler yang akhirnya Google mengenalkan Nexus One, salah satu jenis telepon pintar yang menggunakan Android pada sistem operasinya. Telepon seluler ini diproduksi oleh HTC corporation dan tersedida du pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja Android ARM Holdings, Atheros Communications, diproduksi oleh Asustek Computer Inc, Garmin Ltd, Softbank, Sony Ericsson, Toshiba Corp, dan Vodafone Group Plc. Seiring pembentukan *Open Handset Alliance*, OHA mengumumkan produk perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel Linux 2.6. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Android Versi 1.5 (*Cupcake*) : Pada pertengahan Mei 2009, Google kembali merilis telepon seluler dengan menggunakan Android dan SDK (*Software Development Kit*) dengan versi 1.5 (*Cupcake*). Terdapat beberapa pembaruan termasuk juga penambahan beberapa fitur dalam seluler versi ini yakni kemampuan merekam

dan menonton video dengan modus kamera, mengunggah video ke Youtube dan gambar ke Picasa langsung dari telepon, dukungan Bluetooth A2DP, kemampuan terhubung secara otomatis ke headset Bluetooth, animasi layar, dan keyboard pada layar yang dapat disesuaikan dengan sistem. *Donut* (versi 1.6) dirilis pada September dengan menampilkan proses pencarian yang lebih baik dibanding sebelumnya, penggunaan baterai indikator dan kontrol applet VPN. Fitur lainnya adalah galeri yang memungkinkan pengguna untuk memilih foto yang akan dihapus; kamera, camcorder dan galeri yang dintegrasikan; CDMA / EVDO, 802.1x, VPN, Gestures, dan Text-to-speech engine; kemampuan dial kontak; teknologi text to change speech (tidak tersedia pada semua ponsel; pengadaan resolusi VWGA.

Android versi 2.0/2.1 (*Eclair*) : Pada 3 Desember 2009 kembali diluncurkan ponsel Android dengan versi 2.0/2.1 (*Eclair*), perubahan yang dilakukan adalah pengoptimalan *hardware*, peningkatan Google Maps 3.1.2, perubahan UI dengan browser baru dan dukungan HTML5, daftar kontak yang baru, dukungan flash untuk kamera 3,2 MP, digital Zoom, dan Bluetooth 2.1. Untuk bergerak cepat dalam persaingan perangkat generasi berikut, Google melakukan investasi dengan mengadakan kompetisi aplikasi mobile terbaik (*killer apps* – aplikasi unggulan). Kompetisi ini berhadiah $25,000 bagi setiap pengembang aplikasi terpilih. Kompetisi diadakan selama dua tahap yang tiap tahapnya dipilih 50 aplikasi terbaik. Dengan semakin berkembangnya dan semakin bertambahnya jumlah handset Android, semakin banyak pihak ketiga yang berminat untuk menyalurkan aplikasi mereka kepada sistem operasi Android. Aplikasi terkenal yang diubah ke dalam sistem operasi Android adalah Shazam, Backgrounds, dan WeatherBug. Sistem operasi Android dalam situs internet juga dianggap penting untuk menciptakan aplikasi Android asli, contohnya oleh MySpace dan Facebook.

Android 2.2 (*Froyo/Frozen Yoghurt*) : Fitur-fitur yang ditambahkan pada android versi ini diantaranya adalah fitur Wifi hotspot portable, fitur auto update pada android market, bisa memasang aplikasi pada SD card, dan mampu membuat kinerja aplikasi menjadi lebih cepat 2 hingga 5 kali dari sebelumnya. Terdapat juga fitur untuk meningkatkan keamanan ponsel seperti penggunaan PIN dan kata sandi, fungsionalitas

tethering USB, dan juga fitur play store yang mampu memperbaharui aplikasi secara otomatis.

Android 2.3 (*Gingerbread*) : Gingerbread membawa perubahan yang sangat drastis pada Android seperti perubahan tampilan yang terkesan lebih modern pada ponsel. Perubahan pada antarmuka tidak hanya mempercantik tampilannya, tetapi juga membuat antarmuka semakin terlihat simple dan cepat. Perubahan lain yang hadir pada Android versi ini adalah peningkatan resolusi untuk mendukung ukuran layar yang sangat besar, penambahan fungsi NFC, peningkatan fungsi copy-paste, support format video VP8 dan WebM, berbagai efek audio baru seperti reverb, equalization, bass boost,dll, dan mendukung jumlah kamera yang lebih dari satu. Ponsel pertama yang membawa sistem operasi Gingerbread merupakan ponsel Nexus S yang dikembangkan oleh Samsung. Beberapa fitur lain yang ditambahkan pada sistem operasi Gingerbread versi Android 2.3.3 adalah adanya fitur voice chat dan video chat menggunakan Google Talk, perubahan versi keamanan SSL yang digunakan, dan juga beberapa ponsel yang mampu menggunakan Google Wallet.

Android 3.0/3.1 (*Honeycomb*) : Android ini merupakan OS yang dibuat khusus untuk mengoptimalkan penggunaan tablet dengan membuat beberapa fungsi yang hanya dapat digunakan pada tablet saja. Perubahan signifikan yang terlihat merupakan pemanfaatannya tombol nonfisik untuk navigasi dibanding tombol fisik yang sebelumnya digunakan seperti tombol home dan back. Pada versi ini, ditambahkan juga sistem bar yang mampu digunakan untuk melihat notifikasi, bar aksi yang dapat ditarik dari atas untuk melihat opsi dan segala hal yang sampai sekarang terdapat pada bar aksi pada layar atas. Android ini juga support dengan multi-processor serta memberikan perubahan warna temapada smartphone dan memberikan tampilan preview pada widget. Fitur lain yang tak kalah berguna adalah fitur untuk mengenkripsi semua data pengguna untuk meningkatkan keamanan ponsel.

Android 0 (*Ice Cream Sandwich*) : Beberapa fitur baru yang terdapat pada Android versi ini adalah membuka kunci dengan face recognition, adanya pemantauan serta control dari penggunaan data seluler, fitur pencarian email secara offline, dan berbagi informasi dengan NFC.

Android 4.1/4.2/4.3 (*Jelly Bean*) : Pada Android versi ini mulai terdapat fitur Google Now yang berfungsi mempermudah pencarian yang terdapat pada home screen, Fitur photo sphere, dan berbagai widget terbaru.

Android 4.4 (*KitKat*) : Versi Android ini memberikan optimalisasi yang baik yang bahkan dapat mendukung perangkat smartphone yang spesifikasinya cukup rendah saat itu. Sehingga pada masanya, versi Android ini menjadi jenis Android favorit. Versi Android ini menjadi fondasi dibuatnya smartphone flagship yang memiliki spesifikasi yang cukup bagus. Versi ini juga sudah mulai mendukung mode 64-bit, sehingga memungkinkan penggunaan ram diatas 3GB.

Android 6.0 (*Marshmallow*) : Versi android ini mulai memperkenalkan sistem sensor sidik jari sebagai bentuk keamanan dan sistem pembayaran pada google play. Untuk saat ini.

Android 7.0 (Nougat) : OS Android *Nougat* masih digunakkan pada beberapa smartphone jadul. Versi ini mengalami perubahan pada interface dan juga meng-implementasikan mode split screen, sehingga 2 aplikasi dapat tampil dalam 1 layar.

Android 8.0 & 8.1 (*Oreo*) : *Oreo* menjadi OS android umum karena banyak digunakkan dan bahkan mungkin anda juga pakai. Versi ini menjadi versi kedua setelah KitKat yang menggunakan brand cemilan manis terkenal. OS Android ini membuat pengguna dapat mendapatkan pembaharuan lebih cepat.

Android 9.0 (*Pie*) : OS android *Pie* memberikan navigasi *gesture* yang merubah bentuk tombol *Home*, *Back*, dan *Recent Apps*. Versi ini juga memberikan beberapa fitur yang berguna seperti pengaturan kecerahan, *notification setting*, dan *screenshooting*.

Android 10 (Q) : Versi Android ini masih terbatas pada beberapa perangkat Android saja. Salah satu fitur dalam Android 10 adalah *Dark Mode* yang dipercaya dapat mengurangi daya penggunaan baterai. Dan juga tema penggunaan nama-nama makanan manis sudah dihentikan dari versi Android ini.

Android 11 : Sistem keamanan pada Android 11 lebih canggih dibanding versi versi sebelumnya. OS android 11 juga memungkinkan untuk screen record tanpa aplikasi eksternal, dan juga *screenshot* Panjang.

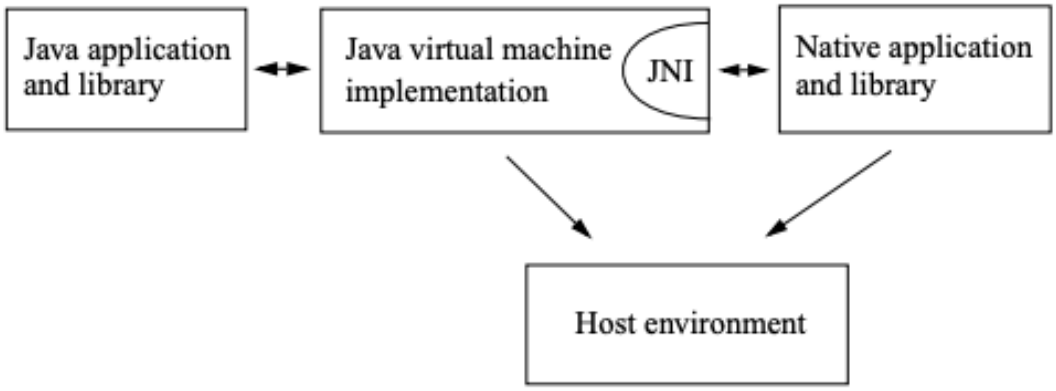
Android 12 : OS ini masih dalam tahap pengembangan dan dikenal sebagai Android 12 beta preview. Versi ini memperkenalkan berbagai fitur baru yang belum ada pada pendahulunya, seperti *One-Handed Mode*, *Customize Phone*, dan *Privary Security Enhancement*. Peningkatan fitur privasi menjadi mungkin dengan diadakannya *Privacy Dashboard* yang berisi data pribadi sepeti akses lokasi, penggunaan kamera dan mikrofon selama 24 jam terakhir. Permintaan terhadap ponsel pintar semakin meningkat dari tahun ke tahun dan perusahaan pembuatnya akan terus bersaing mengembangkan sistem operasi untuk ponsel. Android sebagai sistem operasi memiliki pangsa pasar 71.52% tergolong besar dan diminati oleh banyak orang. Oleh karena itu, Google harus mempertimbangkan untuk menciptakan sistem operasi yang semakin aman, karena tantangan keamanan akan semakin besar seiring meningkatnya permintaan terhadap sistem operasi Android. Bersamaan dengan itu, berbagai fitur baru yang mendukung kenyamanan pengguna juga harus ditingkatkan agar Android semakin diminati pengguna karena kenyamanan dan keamanannya dan dengan demikian dapat bersaing dengan merk lain (Ubaya 2014).

2.1.5 *Java Native Interface (JNI)*

Java Native Interface (*JNI*) adalah fitur canggih dari platform Java. Aplikasi yang menggunakan JNI dapat memasukkan kode *native* yang ditulis dalam Bahasa pemrogaman seperti Bahasa C dan C++, serta kode yang ditulis dalam Bahasa pemrograman Java. JNI memungkinkan *programmer* untuk memanfaatkan kekuatan platform Java, tanpa harus meninggalkan investasi mereka pada kode lama. Karena JNI adalah bagian dari platform Java, *programmer* dapat mengatasi masalah yang interoperabilitas sekalipun dan diharapkan solusi mereka berfungsi dengan semua implementasi pada platform Java.

Disaat platform Java disebarkan diatas *Host Environment,* hal ini memungkinkan aplikasi Java bekerja erat dengan kode *native* yang ditulis dalam Bahasa pemrograman lain. Programmer sudah mulai mengadopsi platform Java untuk membanguun aplikasi yang secara tradisional ditulis dalam C dan C++. Namun, investasi yang ada pada kode lama, aplikasi Java akan jalan berdampingan dengan C dan C++ selama bertahun-tahun. JNI adalah fitur canggih yang akan memungkinkan

programmer memanfaatkan Java, namun tetap menggunakan kode yang ditulis dalam Bahasa lain. Sebagai implementasi dari mesin virtual Java, JNI adalah antarmuka dua arah yang memungkinkan aplikasi Java untuk memanggil kode *native* dan sebaliknya, Gambar 2 mengilustrasikan peran JNI tersdebut.



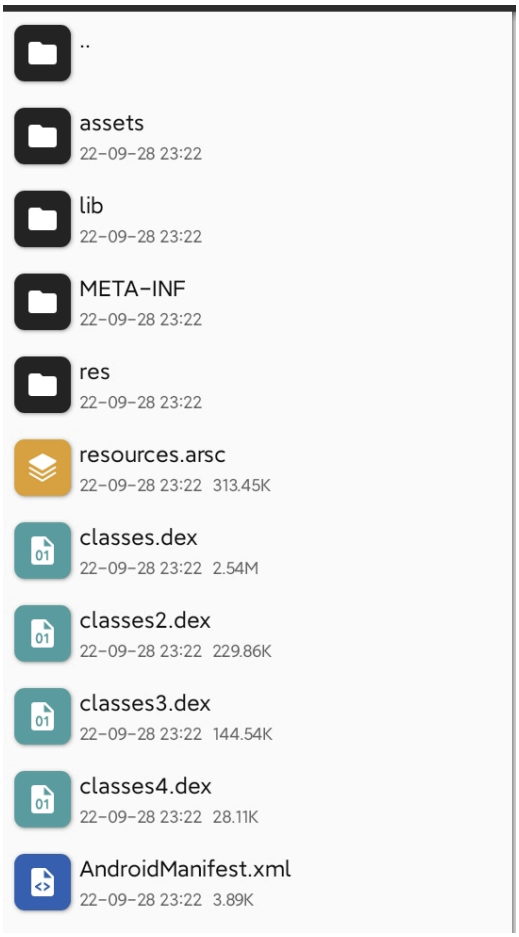
Gambar 2 *Peran JNI pada platfrorm Java*

JNI dirancang uuntuk menangani situasi ketika *programmer* perlu menggabungkan aplikasi Java dengan kode *native.* Sebagaai antarmuka dua arah, JNI dapat mengatasi masalah tersebut.

1. Programmer dapat menggunakan JNI untuk menulis metode asli yang memungkinkan aplikasi Java melakukan pemanggilan fungsi yang diimpleementasikan pada *library*. Aplikasi Java memanggil metode asli dengan cara yang sama seperti mereka memanggil metode dalam Bahasa pemrograman Java. Namun dibalik layar, metode asli diimplementasaikan dalam Bahasa lain yang berada di dalam *library.*
2. JNI mendukung antarmuka pemanggilan yang memungkinkan *programmer* menyemaatkan implementasi mesin virtual Java kedalam aplikai *native.* Aplikasi *native* bisa di tautkan dengan *native library* yang mengimplementasikan mesin virtual Java, dan kemudian gunakan antaarmuka pemanggilan untuk mengeksekusi komponen perangkat lunak yang ditulis dalam Bahasa pemrogramaan Java (Liang 1999).

2.1.6 *Android Package* (APK)

Android Package (APK) adalah format berkas yang digunakan untuk mendistribusikan dan memasang software dan middleware ke ponsel dengan sistem operasi Android, mirip dengan paket MSI pada Windows atau Deb pada OS Debian. Berikut ini gambar komponen-komponen yang terdapat dalam file berjenis APK.



Gambar 3 *Konten APK*

- Folder ***Assets*** berisi file-file pembantu yang dibutuhkan oleh aplikasi android seperti *font,* konfigurasi, dll.
- Folder **lib** berisi kumpulan *shared library* atau *native library.*
- Folder **META-INF** berisi informasi-informasi *manifest* dan metadata lainnya yang berkaitan dengan aplikasi.

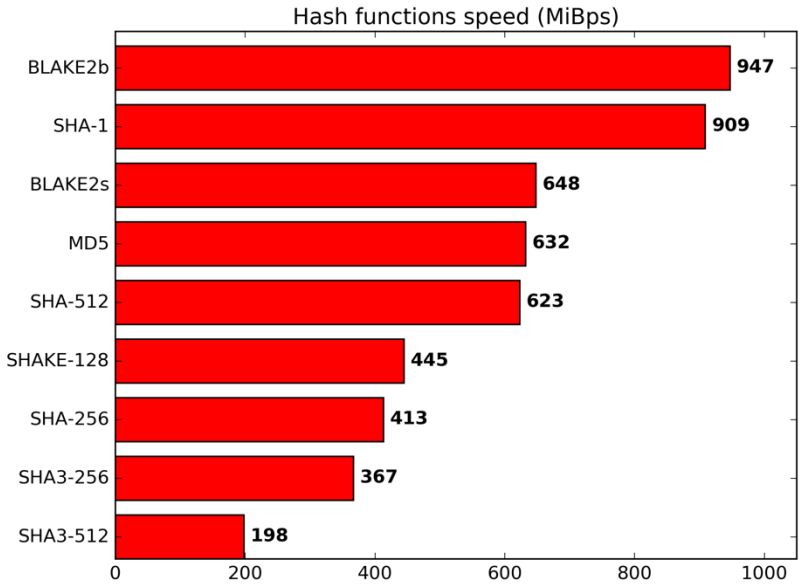
- Folder **res** berisi file-file yang digunakan sebagai layout pada aplikasi android, seperti .xml maupun file .png atau .jpg dan lain sebagainya.
- File **resources.arsc** berisi kumpulan nilai string yang di kompilasi menjadi satu file arsc.
- File **classes*.dex** berisi hasil kompilasi dari file java.
- File **AndroidManifest.xml** berisi penjelasan informasi penting aplikasi yang pertama kali akan dibaca oleh *installer,* seperti informasi Sistem Operasi maksimal yang dapat dijalankan, nama aplikasi, versi aplikasi dan lain-lain.

2.1.7 Algoritma Blake2

Blake2 adalah fungsi hash kriptografi yang leebih cepat dari MD5, SHA-1, SHA-2 dan SHA-3, namun setidaknya sama amannya dengan standar terbaru SHA-3. Blake2 telah diadopsi oleh banyak proyek karena kecepatan tinggi, keamanaan, dan kesederhanaannya, Blake2 ditentukan dalam RFC 7963 (BLAKE2 — fast secure hashing n.d.). Blake2 memiliki dua algoritma utama :

1. Blake2b : dioptimalkan untuk platform 64 Bit yang mendukung ARM dan menghasailkan digit dengan ukuran berapapun antara 1 sampai 65 byte.
2. Blake2s : dioptimalkan untuk platform 8 hingga 32-bit, dan menghasilkan digit dengan ukuran antara 1 hingga 32 byte.

Kedunya dirancang untuk menawarkan kemanan serupa dengan fungsi ideal yang memproduksi digit dengan panjang yang sama. Masing-masing portable untuk CPU apapun, tetapi bisa sampai dua kali lebih cepat bila digunakaan pada ukuran CPU yang telah dioptimakan; misalnya, pada Tegra 2 Blake2 diharapkan lebih cepat sekitar dua kali lipat cepat.



Gambar 4 *Kecepatan fungsi Hash (MiBps)*

Sumber : https://www.blake2.net/skylake.png

2.2 Kajian Literatur

2.2.1 State of the art penelitin

Tabel 1 *State of the art*

No	Judul, Nama, Tahun terbit dan Penerbit Karya Ilmiah	Permsalahan	Metode Penyelesian	Kinerja
1	Signature Verification Based on Android Dex CRC Using Blake2 Algorithm	*Repackaged and Cloned Android apps*	Menggunakan verifikasi signature berdasarkan dex crc dengan algoritma blake2	Kinerja topik yang diusulkan dapat melindungi aplikasi android dari serangan reverse engineering (App repackaging, Cloning, dll)
2	Judul : DroidOlytics : Robust Feature Signature for Repackaged Android apps on Official and Third Party android markets Penulis : Parves Faruki, et al Tahun : 2013 Penerbit : IEEE	*Repackaged Android apps*	Memanfaatkan signature aplikasi android sebagai fitur untuk analisis menggunakan pendekatan statistic terhadap aplikasi yang di repackage	Kinerja metode yang diusulkan dapat mendeteksi varian maleware yang belum deiketahui pada keluarga yang sama
3	Judul : Exploiting Binary-level Code Virtualizatoin to Protect Android Application Against App Repackaging	*Repackaged Android apps*	Menggunakan obfuskasi pada code native (JNI) dengan mengginakan virtualisasi mesin ARM	Kinerja dari metode ini menyebabkan penyerang membutuhkan waktu yang lama untuk dapat melewati

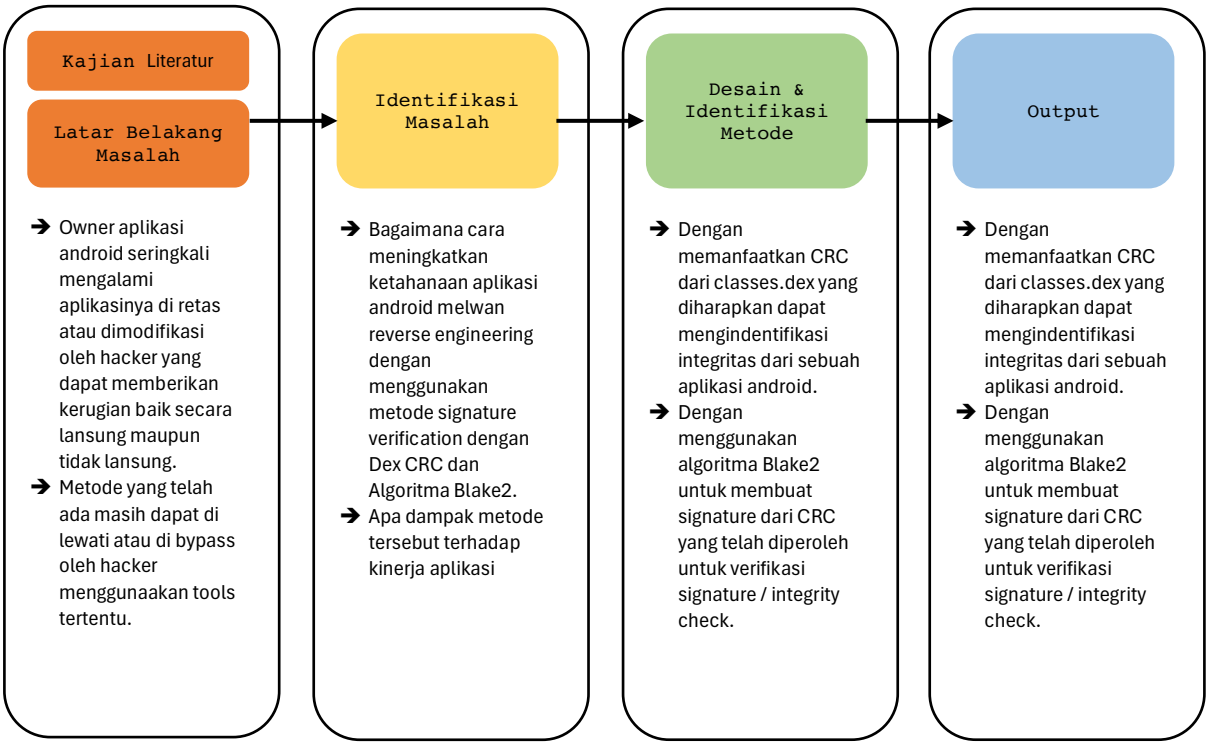
	Penulis : Zhongkai He, et al (He, et al. 2019) Tahun : 2019 Penerbit : IEEE			obfuskasi dan dapat melakukan serangan.
4	Judul : Automated Multi-Layered Bytecode Generation for Preventing Sensitive Information Leaks fron Android Applications Penulis : Geochang Jeon, et al (Geochang, et al. 2021) Tahun : 2021 Penerbit : IEEE	Pencurian Informasi pada aplikasi Android	Metode yang diusulkan yaitu dengan memisahkan kode yang mengandung data sensitive kemudian mengobfuskasi nya.	Metode ini dapat melindungi dari pencurian informasi dengan multi-layer bytecode
5	Judul : RomaDroid : A Robust and efficient Techique for Detecting Android App Clones Using a tree	*App Cloning*	Metode yang diusulkan menggunakan analisis similairity antara masing-masing signature dengan	RomaDroid dapat mendeteksi secara akurat aplikasi yang di gandakan bahkan apabila code telah di obfuskasi

	Structure and Components of Each App's Manifest File Penulis : Byoungchul Kim, et al (Kim, et al. 2019) Tahun : 2019 Penerbit : IEEE		algoritma Longest Common Subsequence (LCS)	
6	Judul : Control Flow Obfuscation Based Protection Method for Android Applications Penulis : Yong Peng, et al (Peng, et al. 2017) Tahun : 2017 Penerbit : IEEE	*Malicious Code Injection*	Menggunakan cotrol flow obfuscatin	Metode yang disusulkan memiliki tingkat compatible yang tinggi dan berhasil memperkuat obfuskasi.
7	Judul : Resilient User-Side Android Application Repakaging and Tampering Detection Using	*App Repackaging*	Menggunakan Cryptographically obfuscated logic bombs	Teknik ini efektif, efisien dan tahan terhadap berbagai Teknik analisis bom termasuk fuzzing, eksekusi

	Cryptographically Obfuscated Logic Bombs (Qiang Zeng, et al, 2021) Penerbit : IEEE			simbolik dan program slicing.
8	DIVILAR: Diversifying Intermediate Language for Anti-Repackaging on Android Platform (Whu Zhou, Zhi Wang, et al. 2014)	*App Repackaging*	Menggunakan VM-based protection	Divilar berhasil memperkuat aplikasi android dari beberapa seerangan reverse engineering akan tetapi menambah beban kerja aplikasi
9	An Android Application Protection Scheme against Dynamic Reverse Engineering Attacks (Kyeonghwan Lim, Younsik Jeong Et al, 2016)	*Reverse Engineering*	Pada penelitian ini penulis menggunakan metode Stub Dex untuk memisahkan kode yang rawan dari dex utama	Penelitian ini menghasilkan skema proteksi untuk mendeteksi perangkat root serta Dex Encryption.
10	Efficient Code Obfuscation for Android (Aleksandrina Kovacheva, 2013)	*Reverse Engineering*	Pada penelitian ini penulis menggabungkaan metode stub dex dan obfuscate yang di implementasaikan pada classes.dex	Metode yang digunakan dapat mengobfuskasi nama class menjadi Unicode dan mengenkripsi string selama runtime.

2.3 Kerangkaa Pikir

Tujuan kerangka pemikiran adalah untuk mengetahui posisi dan keunikan penelitian yang akan dilaksanakan. Kerangka pikir dapat dilihat pada Gambar 5 di bawah ini yang menjelaskan mengenai alur penelitian yang akan dilakukan.



Gambar 5 *Kerangka Pikir*