

IMPLEMENTASI *DIGITAL SIGNATURE* PADA *SECURE ELECTRONIC PRESCRIPTION* MENGGUNAKAN *DIGITAL SIGNATURE ALGORITHM* BERBASIS ANDROID

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:
Hanaria Rotua Tampubolon
NIM: 155150201111171



PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2019

PENGESAHAN

IMPLEMENTASI *DIGITAL SIGNATURE* PADA *SECURE ELECTRONIC PRESCRIPTION*
MENGUNAKAN *DIGITAL SIGNATURE ALGORITHM* BERBASIS ANDROID


SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

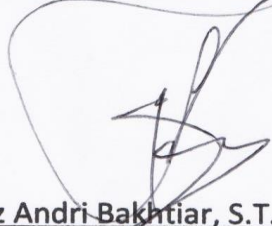
Disusun Oleh :
Hanaria Rotua Tampubolon
NIM: 155150201111171

Skripsi ini telah diuji dan dinyatakan lulus pada
23 Juli 2019
Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I


Ari Kusyanti, S.T., M.Sc
NIP: 19831228 201803 2 002

Dosen Pembimbing 2


Fariz Andri Bakhtiar, S.T., M.Kom.
NIK: 201709 840314 1 001

Mengetahui
Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T., M.T., Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar referensi.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Juli 2019



Hanaria Rotua Tampubolon

NIM: 155150201111171

PRAKATA

Puji syukur penulis panjatkan kehadiran Tuhan Yang Maha Esa atas segala rahmat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Implementasi *Digital Signature* pada *Secure Electronic Prescription* menggunakan *Digital Signature Algorithm* berbasis Android” ini. Penulisan skripsi ini diajukan untuk memenuhi salah satu syarat kelulusan guna memperoleh gelar sarjana komputer pada Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya.

Selama penulisan skripsi ini tidak lepas dari hambatan dan kesulitan, namun berkat bimbingan, dukungan, nasihat, serta kerjasama dari berbagai pihak, penulis mampu mengatasi dan akhirnya dapat menyelesaikan skripsi ini. Oleh karena itu, dalam kesempatan ini penulis dengan tulus hati mengucapkan terima kasih kepada:

1. Ibu Ari Kusyanti, S.T., M.Sc. selaku dosen pembimbing 1 serta Bapak Fariz Andri Bakhtiar, S.T., M.Kom, selaku dosen pembimbing 2, yang telah memberikan bimbingan arah penelitian, ilmu serta motivasi selama pengerjaan skripsi,
2. Bapak Agus Wahyu Widodo, S.T., M.Cs. selaku Kepala Program Studi Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya,
3. Bapak Tri Astoto Kurniawan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Informatika, Fakultas Ilmu Komputer, Universitas Brawijaya,
4. Bapak Wayan Firdaus Mahmudy, S.Si., M.T., Ph.D., Bapak Ir. Heru Nurwasito, M.Kom., Bapak Suprpto, S.T., M.T., dan Bapak Edy Santoso, S.Si., M.Kom, selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
5. Seluruh dosen dan karyawan Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya mengajarkan, membagikan ilmu yang bermanfaat, dan membantu penulis selama masa perkuliahan dan pengerjaan skripsi.
6. Teristimewa kepada kedua orang tua penulis Belman Tampubolon dan Ratna Pardede yang senantiasa memberikan motivasi, dukungan moral dan doa yang berlimpah untuk keberhasilan pengerjaan skripsi.
7. Keempat adik penulis, Apriani I.M. Tampubolon, Mei Liyanti R. Tampubolon, Juliana M. Tampubolon dan Parulian M. Tampubolon atas dukungan dan doanya.
8. Nenek penulis Dameria Manurung yang selalu mendoakan dan memberikan dukungan moral kepada penulis.
9. Sepepu penulis Astri S. Tampubolon yang selalu memberikan dukungan selama proses pengerjaan.
10. Petronella C. Habeahan yang senantiasa menjadi tempat bercerita, berkeluh kesah dan setia memberikan semangat, menghibur disaat penulis dalam suka maupun duka.

11. Seluruh teman-teman dari Futsal Putri Fakultas Ilmu Komputer, khususnya RR. Dea Annisayanti Putri, Dwi Retnoningrum, Vira Indriana yang memberikan dukungan dan doa selama perkuliahan dan masa pengerjaan skripsi.
12. Teman-teman Ferina Gurning, Intan Manalu, dan Ulianna Sibuea untuk dukungan dan kebersamaannya.
13. Teman-teman kelompok pengerjaan proyek selama perkuliahan khususnya Reza Azzubair Wijonarko yang telah memberikan semangat dan dukungan.
14. Seluruh teman-teman dari kelas K tahun 2015 yang telah memberikan dukungan motivasi dan dukungan moril selama pengerjaan skripsi.
15. Seluruh teman-teman Fakultas Ilmu Komputer, Universitas Brawijaya khususnya angkatan 2015 yang telah membantu semasa perkuliahan dengan belajar bersama di kelas dan kepanitiaan.
16. Seluruh teman-teman Ikatan Alumni SMA NEGERI 1 BALIGE di Malang yang telah memberikan dukungan dan telah berbagi ilmu pengetahuan selain Ilmu Komputer.
17. Semua pihak yang tidak dapat penulis sebutkan satu per satu yang telah memberikan bantuan kepada penulis dengan tulus ikhlas.

Semoga jasa dan kebaikan kita semua mendapat balasan dari Tuhan Yang Maha Esa. Penulis menyadari bahwa skripsi ini masih jauh dari kata sempurna. Oleh karena itu penulis sangat mengharapkan kritik dan saran guna perbaikan di masa mendatang. Akhir kata, semoga skripsi ini dapat bermanfaat bagi kita semua dan menjadi bahan masukan dalam bidang pendidikan khususnya ilmu komputer.

Malang, 23 Juli 2019

Penulis

hanaria_rotua@student.ub.ac.id

ABSTRAK

Hanaria Rotua Tampubolon, Implementasi *Digital Signature* pada *Secure Electronic Prescription* menggunakan *Digital Signature Algorithm* berbasis Android.

Pembimbing: Ari Kusyanti, S.T., M.Sc. dan Fariz Andri Bakhtiar, S.T., M.Kom.

Medication safety adalah kondisi dimana seseorang terbebas dari resiko kesehatan yang disebabkan oleh penggunaan obat. *Medication safety* mencegah kematian yang diakibatkan kesalahan penulisan resep oleh dokter. Kesalahan penulisan resep disebabkan oleh kesulitan apoteker dalam membaca tulisan dokter yang terkadang tidak standar. Kesalahan penulisan resep dapat diatasi dengan *e-prescription*, yaitu teknologi elektronik yang digunakan oleh dokter dan para medis lainnya untuk menuliskan dan mengirimkan sebuah resep. Akan tetapi pada *e-prescription* masih rentan dengan serangan pemalsuan dan penyangkalan resep. Maka pada *e-prescription* perlu diterapkannya mekanisme pengamanan untuk menjamin *integrity*, *authentication* dan *non-repudiation* yaitu *digital signature*. Penelitian ini menjelaskan perancangan, penerapan, keamanan dan kinerja *e-prescription* berbasis Android dengan memanfaatkan metode *Digital Signature Algorithm* dan SHA-1 sebagai *hash function*. Hasil penelitian ini untuk memastikan aspek integrity dilakukan dengan pengujian *brute force*, *collision attack* dan *birthday attack*. Namun, aspek *integrity* tidak terpenuhi karena *hash function* yang digunakan yaitu SHA-1 sangat rentan terhadap serangan-serangan tersebut. Untuk menjamin aspek *authentication* dan *non-repudiation* telah dilakukan pengujian dan kedua aspek tersebut sudah terpenuhi. Pembentukan dan verifikasi tanda tangan membutuhkan waktu 6,36963 ms dan 11,79276 ms. Penerapan *digital signature* pada *e-prescription* menggunakan DSA pada penelitian ini telah berhasil memenuhi aspek *authentication* dan *non-repudiation*.

Kata kunci: *Digital Signature Algorithm*, *electronic prescription*, *hash function*, *brute force*, *collision attack*, *birthday attack*, *black box*

ABSTRACT

Hanaria Rotua Tampubolon, Implementation of Digital Signature on Secure Electronic Prescription using Digital Signature Algorithm based on Android.

Supervisors: Ari Kusyanti, S.T., M.Sc. and Fariz Andri Bakhtiar, S.T., M.Kom.

Medication safety is a term to describe safety precautions from drug use. Medication safety prevents deaths that were caused by drugs that were prescribed wrongly. One of the main causes of wrongly prescribing a drug is the pharmacist not being able to read a doctor's instruction which is not standardized. Such problems can be overcome by using e-prescription, which can be utilized by doctor's and other medical staff to prescribe and send the prescription. However, there are risks of forgery and denial of the e-prescription. Therefore, a security mechanism needs to be installed on the e-prescription to guarantee integrity, authentication, and non-repudiation, which is in the form of a digital signature. This research will explain the design, implementation, security, and performance of the Android based e-prescription using the Digital Signature Algorithm method and SHA-1 as a hash function. The research results test the integrity aspect using a brute force, collision attack, and birthday attack. However, the integrity aspect could not be fulfilled, because the hash function being used which was SHA-1, was very vulnerable to the previously mentioned attack types. To ensure the authentication and non-repudiation aspect, a test was conducted and both aspects were fulfilled. The process of forming and verifying the signature required 6.36963 ms and 11,79276 ms respectively. The implementation of the digital signature on the e-prescription using DSA in this research was considered a success in the authentication and non-repudiation aspects.

Keywords: Digital Signature Algorithm, electronic prescription, hash function, brute force, collision attack, birthday attack, black box

DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
PRAKATA.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL.....	xii
DAFTAR GAMBAR	xiv
DAFTAR <i>PSEUDOCODE</i>	xvi
DAFTAR LAMPIRAN	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	3
1.4 Manfaat.....	3
1.5 Batasan Masalah	3
1.6 Sistematika Penulisan	3
BAB 2 LANDASAN KEPUSTAKAAN	5
2.1 Kajian Pustaka	5
2.2 Kriptografi	7
2.3 Kriptografi Asimetris	8
2.4 <i>Digital Signature</i>	9
2.5 <i>Digital Signature Algorithm (DSA)</i>	11
2.5.1 Parameter DSA	11
2.5.2 Pembangkitan Sepasang Kunci	12
2.5.3 Pembangkit Tanda Tangan (<i>Signing</i>).....	12
2.5.4 Verifikasi Keabsahan Tanda Tangan (<i>Verifying</i>).....	12
2.6 <i>Electronic Prescription</i>	13
2.7 Android	13

2.8 Firebase.....	14
2.9 Uji Hipotesis	14
2.10 <i>Brute Force Attack</i>	15
2.11 <i>Collision Attack</i>	15
2.12 <i>Birthday Attack</i>	15
2.13 <i>Black Box</i>	16
BAB 3 METODOLOGI PENELITIAN	17
3.1 Studi Literatur	17
3.2 Analisis Kebutuhan	17
3.2.1 Kebutuhan Perangkat Keras.....	18
3.2.2 Kebutuhan Perangkat Lunak	18
3.3 Perancangan	18
3.4 Implementasi	19
3.5 Pengujian dan Pembahasan.....	20
3.6 Kesimpulan dan Saran	20
BAB 4 Perancangan	21
4.1 Gambaran Umum Sistem.....	21
4.2 Identifikasi Aktor	21
4.3 Analisis Kebutuhan Sistem.....	22
4.3.1 Analisis Kebutuhan Fungsional	22
4.3.2 Kebutuhan Non-fungsional	24
4.4 Pemodelan Kebutuhan	25
4.4.1 <i>Use Case Diagram</i>	25
4.4.2 <i>Use Case Scenario</i>	26
4.4.3 <i>Class Diagram</i>	32
4.4.4 <i>Sequence Diagram</i>	34
4.5 Perancangan Algoritme DSA berdasarkan NIST	36
4.5.1 Pembangkit Parameter DSA.....	36
4.5.2 Pembangkit Sepasang Kunci	39
4.5.3 Pembangkit Tanda Tangan.....	39
4.5.4 Verifikasi Tanda Tangan	41
4.6 Perancangan Antarmuka	42

4.7 Perancangan Pengujian	45
4.7.1 <i>Test Vector</i>	45
4.7.2 Kinerja <i>Digital Signature Algorithm</i>	45
4.7.3 <i>Brute Force Attack</i>	46
4.7.4 <i>Collision Attack</i>	46
4.7.5 <i>Birthday Attack</i>	46
4.7.6 Pengujian Autentikasi dan <i>Non-repudiation</i>	46
4.7.7 <i>Black Box</i>	47
BAB 5 IMPLEMENTASI	49
5.1 Implementasi Algoritme	49
5.1.1 Pembangkit Parameter <i>p</i> dan <i>q</i>	49
5.1.2 Pembangkit Parameter <i>g</i>	50
5.1.3 Pembangkit Sepasang Kunci	50
5.1.4 Implementasi Pembangkit Tanda Tangan.....	51
5.1.5 Implementasi Verifikasi Tanda Tangan	52
5.2 Implementasi Sistem	52
5.2.1 Implementasi Antarmuka Beranda	52
5.2.2 Implementasi Antarmuka Daftar	53
5.2.3 Implementasi Antarmuka Masuk.....	54
5.2.4 Implementasi Antarmuka Dokter Beranda	54
5.2.5 Implementasi Antarmuka Dokter Resep.....	55
5.2.6 Implementasi Antarmuka Apoteker Beranda	56
5.2.7 Implementasi Antarmuka Apoteker Resep.....	56
BAB 6 PENGUJIAN DAN PEMBAHASAN.....	57
6.1 Parameter Pengujian	57
6.2 <i>Test Vector</i>	57
6.2.1 Tujuan Pengujian.....	57
6.2.2 Prosedur pengujian	57
6.2.3 Hasil Pengujian	58
6.3 Kinerja <i>Digital Signature Algorithm</i>	58
6.3.1 Tujuan Pengujian.....	58
6.3.2 Prosedur Pengujian	58

6.3.3 Hasil Pengujian	59
6.4 Pengujian <i>Brute Force Attack</i>	62
6.4.1 Tujuan Pengujian.....	62
6.4.2 Prosedur Pengujian	62
6.4.3 Hasil Pengujian	64
6.5 Pengujian <i>Collision Attack</i>	66
6.5.1 Tujuan Pengujian.....	66
6.5.2 Prosedur Pengujian	66
6.5.3 Hasil Pengujian	68
6.6 Pengujian <i>Birthday Attack</i>	70
6.6.1 Tujuan Pengujian.....	70
6.6.2 Prosedur Pengujian	71
6.6.3 Hasil Pengujian	72
6.7 Pengujian Autentikasi dan <i>Non-repudiation</i>	73
6.7.1 Tujuan Pengujian.....	73
6.7.2 Prosedur Pengujian	73
6.7.3 Hasil Pengujian	74
6.8 Pengujian <i>Black Box</i>	75
6.8.1 Tujuan Pengujian.....	75
6.8.2 Prosedur Pengujian	75
6.8.3 Hasil Pengujian	75
BAB 7 PENUTUP	79
7.1 Kesimpulan.....	79
7.2 Saran	80
DAFTAR REFERENSI	81
LAMPIRAN A HASIL PENGUJIAN	84

DAFTAR TABEL

Tabel 2.1 Kajian Pustaka	5
Tabel 4.1 Daftar Aktor	22
Tabel 4.2 Daftar Kebutuhan Fungsionalitas Tamu	22
Tabel 4.3 Daftar Kebutuhan Fungsionalitas Dokter	23
Tabel 4.4 Daftar Kebutuhan Fungsionalitas Apoteker	24
Tabel 4.5 Daftar Kebutuhan Non-fungsional	24
Tabel 4.6 <i>Use Case Scenario</i> Daftar.....	26
Tabel 4.7 <i>Use Case Scenario</i> Masuk.....	26
Tabel 4.8 <i>Use Case Scenario</i> Buat Kunci	27
Tabel 4.9 <i>Use Case Scenario</i> Simpan Kunci	27
Tabel 4.10 <i>Use Case Scenario</i> Lanjut.....	28
Tabel 4.11 <i>Use Case Scenario</i> Tanda Tangan	28
Tabel 4.12 <i>Use Case Scenario</i> Simpan.....	29
Tabel 4.13 <i>Use Case Scenario</i> Kirim	29
Tabel 4.14 <i>Use Case Scenario</i> Keluar	30
Tabel 4.15 <i>Use Case Scenario</i> Verifikasi.....	30
Tabel 4.16 <i>Use Case Scenario</i> Kembali.....	31
Tabel 4.17 <i>Use Case Scenario</i> Keluar	31
Tabel 4.18 Pengujian <i>Black Box</i> Kebutuhan Fungsional	47
Tabel 4.19 Pengujian <i>Black Box</i> Kebutuhan Non-fungsional	48
Tabel 6.1 Data Uji Pengukuran Waktu	59
Tabel 6.2 Waktu Proses Pembentukan Tanda Tangan dan Verifikasi	59
Tabel 6.3 Hasil Uji Normalitas Waktu Proses Tanda Tangan dan Verifikasi	61
Tabel 6.4 Hasil <i>Independent Sample t-test</i>	62
Tabel 6.5 Data Uji <i>Brute Force Attack</i>	62
Tabel 6.6 Hasil Pengujian <i>Brute Force</i>	64
Tabel 6.7 Tabel <i>Brute Force</i>	66
Tabel 6.8 Data uji <i>Collision Attack</i>	67
Tabel 29Data uji <i>Collision Attack</i> (Lanjutan)	68
Tabel 6.10 Hasil Pengujian <i>Collision Attack</i>	68

Tabel 6.11 Data Uji <i>Birthday Attack</i>	71
Tabel 6.13 Hasil Pengujian <i>Birthday Attack</i> (Lanjutan).....	73
Tabel 6.15 Hasil Pengujian Autentikasi	74
Tabel 6.16 Hasil Pengujian <i>Black Box</i> Kebutuhan Fungsional.....	75
Tabel 6.17 Hasil Pengujian <i>Black Box</i> Kebutuhan Non-fungsional	78

DAFTAR GAMBAR

Gambar 2.1 Diagram Proses Enkripsi dan Dekripsi Kriptografi Asimetris	9
Gambar 2.2 Proses Penandatanganan dan Verifikasi	10
Gambar 3.1 Diagram Alir Metodologi Penelitian	17
Gambar 3.2 Arsitektur Sistem	19
Gambar 4.1 <i>Use Case Diagram Electronic Prescription</i>	25
Gambar 4.2 <i>Class Diagram Electronic Prescription</i>	33
Gambar 4.3 <i>Sequence Diagram</i> Buat Kunci	34
Gambar 4.4 <i>Sequence Diagram</i> Tanda Tangan	35
Gambar 4.5 <i>Sequence Diagram</i> Verifikasi	35
Gambar 4.6 Diagram Alir Algoritme DSA	36
Gambar 4.7 Diagram Alir Pembangkit Parameter p dan q	37
Gambar 4.8 Diagram Alir Pembangkit parameter g	38
Gambar 4.9 Diagram Alir Pembangkit Sepasang Kunci	39
Gambar 4.10 Diagram Alir Pembangkit k dan k^1	40
Gambar 4.11 Diagram Alir Pembangkit Tanda Tangan	40
Gambar 4.12 Diagram Alir Verifikasi Tanda Tangan	41
Gambar 4.13 Perancangan Antarmuka Beranda	42
Gambar 4.14 Perancangan Antarmuka Daftar	42
Gambar 4.15 Perancangan Antarmuka Masuk	43
Gambar 4.16 Perancangan Antarmuka Dokter Beranda	43
Gambar 4.17 Perancangan Antarmuka Dokter Resep	44
Gambar 4.18 Perancangan Antarmuka Apoteker Beranda	44
Gambar 4.19 Perancangan Antarmuka Apoteker Resep	45
Gambar 5.1 Implementasi Antarmuka Beranda	53
Gambar 5.2 Implementasi Antarmuka Daftar	53
Gambar 5.3 Implementasi Antarmuka Masuk	54
Gambar 5.4 Implementasi Antarmuka Dokter Beranda	54
Gambar 5.5 Implementasi Antarmuka Dokter Resep (a)	55
Gambar 5.6 Implementasi Antarmuka Apoteker Beranda	56
Gambar 5.7 Implementasi Antarmuka Apoteker Resep	56

Gambar 6.1 Hubungan Proses Pembentukan Tanda Tangan dan Verifikasi 61

DAFTAR PSEUDOCODE

<i>Pseudocode 5.2 Implementasi Membangkitkan p dan q</i>	50
<i>Pseudocode 5.3 Implementasi Pembangkit Parameter g</i>	50
<i>Pseudocode 5.4 Implementasi Pembangkit Sepasang Kunci</i>	51
<i>Pseudocode 5.5 Pembangkit k dan k'</i>	51
<i>Pseudocode 5.6 Implementasi Pembangkit Tanda Tangan</i>	52
<i>Pseudocode 5.7 Implementasi Verifikasi</i>	52

DAFTAR LAMPIRAN

LAMPIRAN A HASIL PENGUJIAN	Error! Bookmark not defined.
A.1 Hasil Pengujian <i>Brute Force</i>	84
A.2 Hasil Pengujian <i>Collision Attack</i>	86
A.3 Hasil Pengujian <i>Bithday Attack</i>	90

BAB 1 PENDAHULUAN

1.1 Latar Belakang

Dewasa ini, teknologi jaringan komputer berkembang dengan sangat pesat sedemikian hingga diterapkan dalam bidang kesehatan dan disebut sebagai *medication safety*, yakni kebebasan dari bahaya yang dapat dicegah dengan penggunaan obat (ISMP Canada, 2007). *Medication safety* menghindari terjadinya kesalahan pengobatan (*medication error*). Kesalahan pengobatan diakibatkan kesalahan interaksi antara manusia dan sistem komputer pada proses pengobatan (meresepkan, mengeluarkan dan memberikan obat) sehingga menghasilkan akibat yang tidak diinginkan dan berpotensi membahayakan pasien (Mukhopadhyay & Lohani, 2017) dan contoh kesalahan pengobatan adalah kesalahan penulisan resep (*prescription errors*) (Minuz & Velo, 2009).

Kesalahan penulisan resep mengakibatkan 7000 kematian dan peningkatan biaya mordibitas dan mortalitas yang mencapai \$77 miliar per tahun (IOM, 2006). Dalam penelitian (Susanti, 2013) disimpulkan bahwa kesalahan penulisan resep disebabkan oleh beberapa hal seperti terjadinya kesalahan dosis obat dan tidak ditulisnya bentuk kesediaan obat. Selain itu, kesalahan penulisan resep juga disebabkan oleh kesulitan apoteker dalam membaca resep dari dokter. Menurut IOM dalam (IOM, 2006), kesalahan penulisan resep dapat diatasi oleh *e-prescription*, yakni teknologi elektronik yang memungkinkan dokter dan praktisi medis lainnya untuk menulis resep elektronik dan mengirimkannya langsung ke komputer apotek yang dikehendaki yang tergabung dalam jaringan *e-prescribing* (Coustasse dkk., 2014).

E-prescription tidak sempurna karena rentan mengalami serangan pemalsuan dan penyangkalan resep (Noviyanto & Nugroho, 2013). Seseorang yang berniat jahat dapat memalsukan sebuah resep dengan menyadap resep yang sah dan memasukkan informasi apapun pada resep itu. Contoh lain, tanpa fitur pengamanan yang mumpuni, identitas pasien dalam suatu resep bisa dipalsukan sehingga resep dapat berisi informasi sesuka orang yang berniat jahat. Sebutan dari tindakan ini adalah pemalsuan resep. Pemalsuan resep tersebut marak terjadi di apotek-apotek di Indonesia karena apoteker kesulitan membedakan antara resep asli dengan resep palsu. Selain itu, masalah pada penyangkalan resep juga terjadi sebagai contoh pada kasus kematian salah satu penyanyi terkenal. Seorang dokter yang merawatnya menyangkal dan mengaku bahwa dirinya tidak pernah menuliskan resep obat kepada pasiennya (Sentosa, 2011). Apoteker sebenarnya dapat memastikan keaslian (integritas) dan *non-repudiation* resep dengan menghubungi kontak dokter yang tercantum pada resep, tetapi tidak semua resep berisi informasi kontak dokter yang bisa dihubungi apoteker. Dengan kata lain, perlu ada mekanisme pengamanan untuk menjamin integritas dan *non-repudiation* pada sebuah resep yang dihasilkan sistem *e-prescription*.

Beberapa penelitian telah dilakukan dalam menyelesaikan masalah penyangkalan dan pemalsuan resep. Salah satunya adalah penggunaan

otentikasi dua faktor dalam *e-prescription* (Ohio Board Pharmacy, 2018). Untuk proses autentikasi tersebut, *National Institute of Standards and Technology* mengusulkan penerapan *digital signature* (NIST, 2013). *Digital signature* adalah suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan atau *signer*. Salah satu metode yang digunakan dalam *digital signature* adalah *Digital Signature Algorithm* (DSA). *Digital Signature Algorithm* (DSA) merupakan salah satu algoritme dalam kriptografi asimetris yang memanfaatkan kunci publik untuk autentikasi, pengamanan data dan perangkat *non-repudiation*. DSA dirancang untuk mencegah dan menjaga data dari serangan luar atau *attacker* yang diasumsikan tidak mengetahui kunci privat *signer* yang digunakan untuk membangkitkan *digital signature*. Dibandingkan metode lain seperti *Rivest Shamir Adleman* (RSA) dan *Elliptical Curve Digital Algorithm* (ECDSA), DSA memiliki kelebihan berupa proses pembuatan kunci, penandatanganan sebuah data menggunakan waktu yang lebih cepat (Sivaraman K., 2006). DSA juga memiliki kekurangan yaitu proses verifikasi lebih lama dibandingkan dengan RSA.

Sistem yang diusulkan dalam penelitian, meski bekerja dengan baik, kurang praktis diterapkan karena penandatanganan dan verifikasi dilakukan dalam komputer. Penggunaan komputer yang statis membuat sistem kurang nyaman diakses pengguna karena komputer tidak dapat diakses di mana saja. Di sisi lain, terdapat *smartphone* yang bermobilitas tinggi sehingga pengguna dapat menggunakan *smartphone* di mana pun. *Smartphone* saat ini telah dilengkapi dengan spesifikasi yang cukup untuk menangani proses penandatanganan dan verifikasi.

Berdasarkan penjelasan di atas diusulkan implementasi *digital signature* pada *Secure Electronic Prescription* menggunakan DSA untuk menjamin integritas data, *non-repudiation*, dan masalah autentikasi. Sistem yang dibangun terdiri atas *smartphone* Android dan *database* Firebase. *Smartphone* digunakan oleh entitas-entitas untuk melakukan penulisan resep serta penandatanganan pada resep tersebut dan untuk mengakses resep serta melakukan verifikasi terhadap resep tersebut. Entitas-entitas pengguna sistem yang dimaksud adalah apoteker, dan dokter. Kemudian, Firebase digunakan untuk menyimpan resep serta *digital signature* penulis resep tersebut dan juga informasi masuk (*login*) entitas-entitas pengguna sistem.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah dibahas, maka rumusan masalah yang menjadi pokok penelitian yaitu:

1. Bagaimana implementasi *digital signature* menggunakan DSA pada *Secure Electronic Prescription* (SEP) berbasis Android?
2. Bagaimana kinerja *digital signature* menggunakan DSA yang telah diimplementasikan?
3. Bagaimana keamanan *digital signature* menggunakan DSA yang telah diimplementasikan?

1.3 Tujuan

Tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut:

1. Mengimplementasikan *digital signature* menggunakan DSA pada SEP berbasis Android.
2. Mengetahui kinerja *digital signature* menggunakan DSA yang telah diimplementasikan.
3. Mengetahui keamanan *digital signature* menggunakan DSA yang telah diimplementasikan.

1.4 Manfaat

Manfaat yang diharapkan dari penelitian Implementasi *digital signature* pada *Secure Electronic Prescription* menggunakan DSA berbasis Android sebagai berikut:

1. Pengguna dapat melakukan pengiriman informasi dan dapat diakses oleh orang yang bersangkutan untuk informasi tersebut.
2. Pengguna dapat mencegah pengaksesan informasi dari orang-orang yang tidak sah untuk menghindari upaya penyalahgunaan resep selama proses pengiriman resep dari dokter kepada apoteker.
3. Pengguna dapat melakukan pengiriman informasi agar tidak terjadi lagi hal-hal buruk seperti salah memberikan obat kepada pasien dikarenakan tulisan dokter tidak terbaca.
4. Dapat mencegah anti penyangkalan pada saat pemberian resep oleh dokter, jika suatu saat terjadi resep yang diberikan dokter tidak sesuai dengan pasien.
5. Untuk penelitian selanjutnya, penelitian ini dapat menjadi landasan dasar dan gambaran umum untuk mempermudah dalam pengembangan sistem ini.

1.5 Batasan Masalah

Agar pembahasan penelitian ini lebih terarah dan tidak adanya menyimpang dari yang telah dirumuskan, batasan masalah dalam penelitian ini adalah:

1. Algoritme DSA dengan panjang kunci privat 1024-bit dan kunci publik 160-bit yang digunakan dalam penelitian ini.
2. Android 5.1 (*Lolypop*) adalah sistem operasi yang digunakan.
3. Algoritme SHA-1 adalah *hash function* yang digunakan.
4. Bahasa pemrograman Java yang digunakan untuk implementasi sistem ini.

1.6 Sistematika Penulisan

Sistematika pembahasan dan penyusunan laporan penelitian ini ditujukan untuk menggambarkan dan menguraikan secara garis besar yang meliputi beberapa bab, sebagai berikut:

BAB I : PENDAHULUAN

Pada bab ini menjelaskan mengenai informasi umum latar belakang masalah yang diteliti, perumusan masalah yang menjadi permasalahan untuk dipecahkan, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika pembahasan.

BAB II : LANDASAN KEPUSTAKAAN

Pada bab ini menjelaskan mengenai dasar-dasar teori yang berhubungan dengan perumusan masalah sebagai acuan untuk membahas permasalahan pada penelitian nantinya. Dasar-dasar teori tersebut dikutip dari beberapa buku, jurnal, *website* dari lembaga resmi yang dapat dipertanggungjawabkan.

BAB III : METODOLOGI PENELITIAN

Pada bab ini, menjelaskan metode yang dilakukan dalam pelaksanaan penelitian diantaranya yaitu studi literatur, perancangan, implementasi, pengujian dan analisis, serta penarikan kesimpulan.

BAB IV : PERANCANGAN

Pada bab ini menjelaskan tentang gambaran umum dari perancangan implementasi *digital signature* pada *Secure Electronic Prescription* menggunakan DSA berbasis Android yang diimplementasikan nantinya.

BAB V : IMPLEMENTASI

Pada bab ini menjelaskan proses penerapan dari rancangan yang sudah dituliskan pada bab sebelumnya.

BAB VI : PENGUJIAN DAN PEMBAHASAN

Pada bab ini menjelaskan tentang pengujian dan pembahasan dari kinerja sistem yang telah dibangun, apakah sistem yang dibangun dapat menjawab masalah yang telah dirumuskan pada bab sebelumnya.

BAB VII : PENUTUP

Pada bab ini menjelaskan kesimpulan setelah dilakukannya penelitian dan memberikan saran untuk penelitian berikutnya.

BAB 2 LANDASAN KEPUSTAKAAN

Pada bab landasan kepastakaan menjelaskan tentang studi literatur yang mendukung penelitian ini dan dilanjutkan pada subbab tentang dasar-dasar teori yang digunakan sebagai pedoman teoritis dalam pelaksanaan penelitian ini.

2.1 Kajian Pustaka

Kajian pustaka yang digunakan adalah penelitian yang telah dilakukan sebelumnya tentang DSA. Kajian pustaka dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kajian Pustaka

No.	Nama Penulis, Tahun, Judul	Persamaan	Perbedaan	
			Penelitian Terdahulu	Rencana Penelitian
1	<i>National Institute of Standards and Technology, 2013, "Digital Signature Standard (DSS)".</i>	Metode yang digunakan DSA.	Perancangan parameter, pasangan kunci, penandatanganan, dan verifikasi pada DSA.	Menerapkan DSA pada objek <i>e-prescription</i> .
2	K. Ramya, dan K. Suganya, 2013, <i>"Design and Implementation of Digital Signature"</i> .	Metode yang digunakan adalah DSA.	Perancangan parameter, pasangan kunci, penandatanganan, dan verifikasi pada DSA.	Menerapkan DSA pada objek <i>e-prescription</i> .
3	Mohamad Ali Sadikin & Septia Ulfa Sunaringtyas, 2016, <i>"Implementing Digital Signature for the Secure Electronic Prescription Using QR-code Base on Android Smartphone"</i> .	Penggunaan objek yaitu <i>Electronic Prescription</i> .	Mengembangkan <i>digital signature</i> menggunakan algoritme RSA.	Memanfaatkan DSA pada pengembangan <i>digital signature</i> .

National Institute of Standard (NIST) telah mengembangkan DSA dari tahun 1998-2013. Pengembangan yang dilakukan dan tertulis pada FIPS PUB 186-4 dengan judul *Digital Signature Standard (DSS)* halaman 15-21 yaitu perancangan parameter, pasangan kunci, penandatanganan dan verifikasi menggunakan DSA. Penelitian ini juga dilakukan untuk merancang pemilihan ukuran parameter dan *hash function* yang digunakan (NIST, 2013). Persamaan penelitian saat ini dengan sebelumnya adalah penggunaan hasil perancangan parameter, pasangan kunci, penandatanganan dan verifikasi menggunakan metode DSA. Perbedaan penelitian ini dengan sebelumnya adalah penggunaan DSA pada objek yang berbeda. Pada penelitian ini, objek adalah *e-prescription*.

Penelitian oleh K. Ramya dan K. Suganya dengan judul *Design and Implementation of Digital Signature* telah menyimpulkan bahwa banyak teknologi keamanan menggunakan tanda tangan digital. Sebagai contoh *Microsoft* dapat digunakan untuk menandatangani secara digital program perangkat lunak, melindungi program-program tersebut ketika didistribusikan di internet. Demikian juga *digital signature* dapat digunakan untuk menandatangani pesan secara digital untuk memastikan integritas komunikasi (Suganya & Ramya, 2013). Persamaan penelitian ini dengan sebelumnya adalah penggunaan perancangan parameter dan kunci dengan metode DSA dan perbedaan dengan penelitian sebelumnya adalah menerapkannya pada objek *e-prescription*. Penerapan metode ini pada *e-prescription* untuk mengatasi masalah-masalah yang sudah dijelaskan pada bab sebelumnya.

Pada penelitian oleh Sadikin & Sunaringtyas, Sistem *Secure Electronic Prescription* telah berhasil dikembangkan dengan menggunakan *digital signature* yaitu berupa QR-code berbasis Android. Penelitian ini memiliki konsep kriptografi berfungsi untuk memastikan bahwa data resep dan QR-code yang berisikan rincian informasi resep hanya dikirimkan untuk apoteker yang sudah ditentukan oleh dokter atau untuk mencegah masalah *cybercrime* seperti pencurian, modifikasi, dan akses yang tidak sah (Sunaringtyas & Sadikin, 2016). Algoritme yang digunakan adalah RSA 2048-bit dan diterapkan menggunakan bahasa pemrograman Java dan sistem berbasis Android. Penelitian ini berfokus pada penggabungan layanan yang diberikan oleh sistem yaitu kerahasiaan, autentikasi dan *non-repudiation*.

Persamaan penelitian saat ini dengan sebelumnya adalah dengan memanfaatkan penggunaan *digital signature* pada objek *electronic prescription* untuk memberikan layanan autentikasi dan *non-repudiation*. Namun perbedaan penelitian ini dengan sebelumnya adalah menggunakan algoritme RSA 2048-bit sedangkan penelitian ini menggunakan DSA karena pada saat pembuatan *digital signature* dan mendekripsikan sebuah data waktu yang digunakan lebih cepat dibandingkan penggunaan algoritme RSA.

2.2 Kriptografi

Kriptografi adalah suatu teknik yang digunakan untuk mengacak sebuah pesan agar tidak diketahui arti dari pesan tersebut. Kriptografi adalah ilmu pengetahuan dan seni yang berfungsi untuk menjaga pesan-pesan agar tetap aman (*secure*) (Schneier, 1996). Kriptografi memiliki prinsip dasar yaitu menyembunyikan informasi dari pesan dengan berbagai cara sehingga orang-orang yang tidak berhak atau tidak memiliki akses tidak dapat mengetahui isi dari pesan tersebut. Pada peradaban Mesir dan Romawi telah diterapkan konsep kriptografi dalam keseharian. Sistem keamanan mencakup lima aspek, yaitu (Widiyanto, 2007):

1. *Privacy/Confidentiality*

Aspek *privacy* atau *confidentiality* merupakan usaha yang dilakukan dengan tujuan melindungi informasi dari beberapa pihak yang tidak memiliki hak untuk mengakses informasi yang dilindungi. *Privacy* merupakan data yang bersifat rahasia, sedangkan *confidentiality* adalah data yang ditujukan kepada pihak lain dengan maksud dan tujuan tertentu.

2. *Integrity*

Integrity atau integritas lebih mengutamakan bahwa suatu informasi hanya dapat diubah dengan adanya izin dari pemilik informasi. Suatu informasi jika mengalami perubahan dari informasi aslinya maka aspek integritasnya tidak tercapai, sebagai contoh pada suatu informasi terdapat virus, sehingga informasi tersebut dapat dikatakan tidak utuh lagi.

3. *Authentication*

Authentication merupakan aspek yang mengutamakan bahwa informasi benar-benar asli dan orang-orang yang akan mengakses dan memberikan informasi itu benar-benar yang dituju atau dengan kata lain *server* yang ditujukan adalah *server* yang asli.

4. *Availability*

Availability berfokus pada ketersediaan dari informasi. Informasi yang tersedia hanya dapat diakses oleh orang yang berhak saja.

5. *Access Control*

Access control untuk mengatur pihak yang dapat mengakses informasi. Misalnya seorang *admin* memiliki hak akses penuh terhadap sebuah komputer, tetapi hal ini tidak berlaku bagi *account guest* ataupun *limited account* lainnya.

Pembakuan penulisan pada kriptografi dapat dituliskan dalam fungsi matematika. Fungsi-fungsi mendasar dalam kriptografi adalah enkripsi dan dekripsi. Menurut penelitian sebelumnya terdapat komponen-komponen yang digunakan untuk mencapai tujuan kriptografi. Ada tujuh komponen yang digunakan, yaitu (Ariyus, 2006):

1. Enkripsi (*Encryption*)

Enkripsi bertujuan untuk mengamankan informasi pada saat pengiriman agar informasi tersebut terjaga kerahasiaannya. Informasi tersebut diubah dalam

bentuk serangkaian kode yang sulit untuk dibaca dan diartikan. Enkripsi dapat diartikan sebagai *cipher* atau kode.

2. Dekripsi (*Decryption*)

Dekripsi yaitu suatu proses untuk mengubah informasi dalam bentuk kode yang sulit diartikan (pesan yang dienkripsi) ke dalam bentuk informasi asli. Dekripsi dapat diartikan sebagai *decipher*.

3. Kunci

Kunci yang dihasilkan berfungsi untuk melakukan enkripsi dan dekripsi. Kunci terdiri dari dua yaitu kunci publik pada umumnya diketahui oleh banyak orang dan kunci privat yang hanya diketahui orang-orang tertentu saja.

4. *Plaintext*

Plaintext (cleartext) yaitu pesan asli yang dituliskan dan dapat dibaca seperti biasanya. *Plaintext* diubah menjadi *ciphertext* dengan menggunakan algoritme kriptografi.

5. Pesan

Pesan merupakan isi dari sebuah informasi yang dikirimkan oleh pengirim kepada penerima melalui saluran komunikasi data, kurir dan sebagainya yang disimpan dalam media penyimpanan seperti memori, kertas dan sebagainya.

6. *Ciphertext*

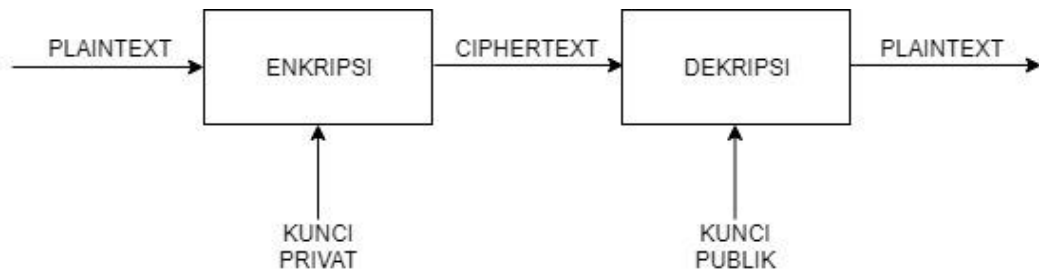
Ciphertext adalah hasil dari proses enkripsi pesan dari pesan asli ke dalam bentuk kode atau karakter yang tidak bisa dibaca karena tidak memiliki arti atau makna.

7. Kriptanilis (*Cryptanalysis*)

Kriptanilis merupakan suatu ilmu yang menganalisis dan memecahkan *ciphertext* ke dalam bentuk *plaintext* tanpa mengetahui kunci yang digunakan. Orang yang melakukan kriptanilis disebut *cryptanalysis* atau kriptanalisis.

2.3 Kriptografi Asimetris

Umumnya selain menggunakan fungsi tertentu dalam melakukan enkripsi dan dekripsi, sering kali fungsi itu diberi parameter tambahan yang disebut dengan istilah kunci yang dibagi menjadi kunci simetris dan asimetris. Penelitian sebelumnya yang dikemukakan oleh Whitfield Diffie dan Martin Hellman menghasilkan sebuah temuan yaitu teknik enkripsi asimetris atau sering disebut dengan kriptografi asimetris atau kriptografi kunci publik (W. Diffie & M. Hellman, 1976). Kriptografi asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk enkripsi dan satunya lagi digunakan untuk dekripsi. Kunci yang digunakan untuk mendekripsi (kunci publik) pesan dapat diketahui oleh siapapun, sedangkan kunci yang digunakan untuk mengenkripsi (kunci privat) pesan hanya diketahui satu orang saja yaitu penerima pesan.



Gambar 2.1 Diagram Proses Enkripsi dan Dekripsi Kriptografi Asimetris

Sumber: Ana Wahyuni (2011)

Pada Gambar 2.1 adalah diagram proses enkripsi dan dekripsi kriptografi asimetris. *Plaintext* dienkripsi menggunakan kunci privat sehingga menghasilkan *ciphertext* dan *ciphertext* didekripsi menggunakan kunci publik dan menghasilkan informasi ke bentuk semula (*plaintext*). Dari penjelasan tersebut dapat diketahui bahwa kunci yang digunakan saat proses enkripsi dan dekripsi berbeda. Kriptografi asimetris ini memiliki kelebihan dan kelemahan. Kelebihannya adalah keamanan dalam pendistribusian kunci lebih baik, untuk perubahan kunci publik dan kunci privat jarang dilakukan, masalah manajemen kunci yang lebih baik karena jumlah kunci lebih sedikit. Sedangkan kelemahannya adalah tidak adanya jaminan bahwa kunci publik benar-benar aman, kunci yang digunakan untuk tingkat keamanan yang sama lebih panjang dibandingkan dengan kriptografi simetris, kecepatannya lebih rendah dibandingkan dengan kriptografi simetris. Contoh algoritme terkenal yang menggunakan kunci asimetris adalah RSA dan DSA.

2.4 Digital Signature

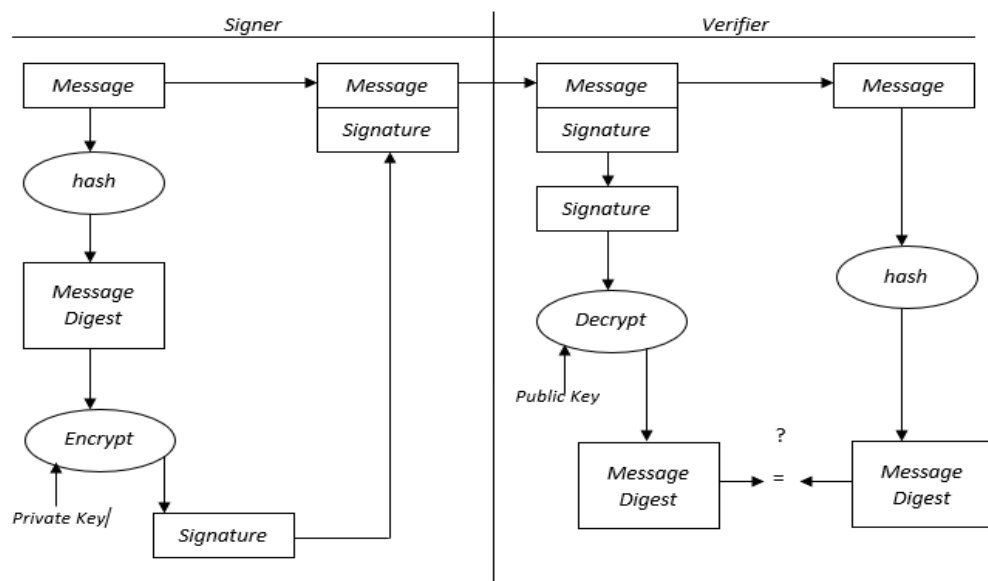
Digital signature bukanlah tanda tangan manual yang digitalisasi dengan alat pemindaian tetapi suatu nilai kriptografis yang bergantung pada pesan dan pengirim pesan (Munir, 2005). Keuntungan menggunakan *digital signature* yaitu dapat menjamin integritas data, dapat membuktikan keabsahan pengirim (asal pesan) dan *non-repudiation*.

Pemberian tanda tangan dengan cara mengenkripsi mempunyai dua fungsi yaitu kerahasiaan dan autentikasi pesan. Namun dalam beberapa kasus untuk kerahasiaan pesan tidak diperlukan proses enkripsi sehingga hanya membutuhkan autentikasi pesan saja. Pemberian *digital signature* dengan menggunakan fungsi *hash* hanya dapat menggunakan kriptografi asimetris; karena skema *digital signature* berbasis sistem kunci publik yang dapat menyelesaikan masalah *non-repudiation* (baik penerima dan pengirim pesan mempunyai pasangan kunci masing-masing). Beberapa aspek yang dapat dijamin jika suatu pesan telah diberi *digital signature* yaitu:

1. *Integrity*. *Digital signature* dapat memastikan bahwa pesan yang dikirimkan itu masih utuh atau tidak dimodifikasi pada saat pengiriman pesan tersebut karena pada pesan asli tidak dilakukan enkripsi sehingga pesan tersebut dapat dibaca semua pihak. Pesan tersebut dikatakan utuh atau tidak dimodifikasi dengan cara membandingkan *message digest* dari pesan asli dan *plaintext*

dari hasil verifikasi *digital signature*. Jika hasilnya sama maka dokumen telah terjamin keasliannya.

2. *Authentication*. *Digital signature* dibentuk dengan cara mengenkripsi *message digest* dari pesan asli menggunakan kunci privat pengirim. Hasil dari enkripsi *message digest* hanya dapat didekripsi dengan menggunakan kunci publik pasangan dari kunci privat. Maka jika hasil verifikasi membuktikan bahwa *message digest* sama dengan hasil dekripsi tersebut, maka dapat ditarik kesimpulan pengirim adalah benar-benar yang memiliki kunci privat.
3. *Non-repudiation*. Jika *digital signature* telah terbukti ditandatangani menggunakan kunci privat tertentu, maka orang yang menulis pesan tersebut tidak menyangkal bahwa dialah yang menulis pesan tersebut, dengan kata lain orang yang menulis pesan itu harus bertanggung jawab terhadap apa yang dia tuliskan.



Gambar 2.2 Proses Penandatanganan dan Verifikasi

Sumber: Ana Wahyuni (2011)

Gambar 2.2 menjelaskan bahwa pengirim membuat sebuah pesan dan hendak dikirimkan kepada penerima. Lalu pengirim mengubah pesan yang telah dibuat menjadi *message digest* dengan menggunakan fungsi *hash* dan mengenkripsi *message digest* menggunakan kunci privat yang dimiliki oleh pengirim. Maka hasil enkripsi tersebut merupakan *digital signature*. Hasil enkripsi (*digital signature*) dapat dilampirkan ke pesan asli atau dapat dikirim secara terpisah dalam waktu bersamaan, lalu dikirimkan kepada penerima. Proses yang dilakukan pengirim adalah proses penandatanganan (*signer*).

Penerima mendapatkan pesan beserta *digital signature*-nya yang dikirimkan oleh pengirim. Lalu penerima membagi dua bagian yaitu pesan asli dan *digital signature*-nya. Penerima menggunakan fungsi *hash* dari pesan asli dan memperoleh *message digest* dari pesan tersebut. Penerima melakukan dekripsi pada *digital signature* menggunakan kunci publik milik pengirim dan setelah melakukan dekripsi memperoleh *plaintext*. Pengirim membandingkan *plaintext* dan *message digest* tersebut, apabila hasilnya sama maka pesan tersebut benar-

benar ditulis oleh pengirim, karena hanya pengirim yang tau kunci privat miliknya dan pesan tersebut tidak dimodifikasi oleh siapapun. Proses yang dilakukan oleh penerima adalah proses verifikasi (verification).

Penelitian yang dilakukan oleh NIST (*The National of Standart and Technology*) mengumumkan standard untuk *digital signature* terdiri dari dua komponen (NIST, 2013) yaitu :

1. DSA sebagai algoritme *digital signature* (pemberian tanda tangan pada pesan).
2. *Hash function* yang disebut *Secure Hash Algorithm* (SHA) untuk membangkitkan *message digest* dari pesan.

2.5 Digital Signature Algorithm (DSA)

Digital Signature Algorithm merupakan anggota algoritme kriptografi asimetrik atau kriptografi kunci publik. Penggunaan DSA difokuskan untuk menjamin integritas dan autentikasi bukan *confidentiality*. DSA mempunyai dua fungsi utama yaitu pembentukan tanda tangan (*signature generation*) dan pemeriksaan keabsahan tanda tangan (*signature verification*). DSA menggunakan dua buah kunci yaitu kunci privat digunakan untuk pembentukan tanda tangan dan kunci publik digunakan untuk verifikasi tanda tangan. DSA menggunakan fungsi *hash Secure Hash Algorithm* (SHA) untuk mengubah pesan menjadi intisari pesan yang berukuran 160-bit. DSA dan algoritme *digital signature* lainnya mempunyai tiga proses utama yaitu pembangkitan pasangan kunci (*Key Pair Generation*), pembangkitan *digital signature* (*Digital Signature Generation*) dan verifikasi *digital signature* (*Digital Signature Verification*).

2.5.1 Parameter DSA

DSA dikembangkan dari algoritme *National Institute of Standards and Technology* (NIST). DSA dikompilasi menggunakan seperangkat parameter domain, data yang akan ditandatangani dan fungsi *hash* (NIST, 2013). Parameter didefinisikan sebagai berikut:

1. p adalah bilangan prima, dengan $2^{L-1} < p < 2^L$ dan L adalah panjang bit p .
2. q merupakan faktor prima dari $(p - 1)$, dengan syarat $2^{N-1} < q < 2^N$ dan N adalah panjang bit q .
3. g adalah generator dari subkelompok urutan q dalam kelompok multiplikasi $FG(p)$, sehingga $1 < g < p$.
4. x merupakan kunci privat yang harus tetap rahasia. x adalah bilangan bulat acak atau pseudo yang dihasilkan secara acak, sehingga $0 < x < q$. x berada dalam kisaran $[1, q - 1]$.
5. y merupakan kunci publik, untuk mendapatkan nilai y dapat menggunakan persamaan 2.1.
$$y = g^x \bmod p \quad (2.1)$$
6. k merupakan nomor rahasia yang unik untuk setiap pesan. k adalah bilangan bulat acak atau pseudo yang dihasilkan secara acak, sehingga $0 < k < q$. k berada dalam kisaran $[1, q - 1]$.

2.5.2 Pembangkitan Sepasang Kunci

Adapun prosedur pembangkitan sepasang kunci adalah:

1. Bilangan prima p dan q dipilih sesuai dengan persamaan 2.2.

$$(p - 1) \bmod q = 0 \quad (2.2)$$

2. Nilai g dihitung menggunakan persamaan 2.3 yang dalam hal ini $1 < h < p - 1$ dan $h^{(p-1)/q} \bmod p > 1$.

$$g = h^{(p-1)/q} \bmod p \quad (2.3)$$

3. Nilai x dicari dengan ketentuan $0 < x < q$ dan x merupakan kunci privat.
4. Nilai y dicari dengan ketentuan persamaan 2.4 dan y merupakan kunci publik.

$$y = g^x \bmod p \quad (2.4)$$

Prosedur di atas menghasilkan kunci publik dinyatakan sebagai (p, q, g, y) dan untuk kunci privat dinyatakan sebagai (p, q, g, x) .

2.5.3 Pembangkit Tanda Tangan (*Signing*)

Prosedur pembangkitan tanda tangan (*signing*) adalah sebagai berikut:

1. Pesan m diubah menjadi intisari pesan dengan fungsi *hash* SHA, H .
2. Bilangan acak ditentukan sesuai dengan $k < q$.
3. Tanda tangan dari pesan m adalah bilangan r dan s . Menghitung r dan s menggunakan persamaan 2.5, 2.6 dan 2.7.

$$r = (g^k \bmod p) \bmod q \quad (2.5)$$

$$z = \text{the leftmost } \min(N, \text{outlen}) \text{ bits of Hash}(M) \quad (2.6)$$

$$s = (k^{-1} (z + xr)) \bmod q \quad (2.7)$$

4. Pesan m dikirim beserta tanda tangan r dan s .

2.5.4 Verifikasi Keabsahan Tanda Tangan (*Verifying*)

Prosedur verifikasi keabsahan tanda tangan adalah sebagai berikut:

1. Nilai w , U_1 , U_2 , dan v dihitung menggunakan persamaan 2.8, 2.9, 2.10, 2.11, dan 2.12.

$$w = s'^{-1} \bmod q \quad (2.8)$$

$$z = \text{the leftmost } \min(N, \text{outlen}) \text{ bits of Hash}(M') \quad (2.9)$$

$$U_1 = (zw) \bmod q \quad (2.10)$$

$$U_2 = ((r')w) \bmod q \quad (2.11)$$

$$v = (((g^{u_1}(y^{u_2}) \bmod p) \bmod q \quad (2.12)$$

2. Tanda tangan sah apabila memenuhi persamaan 2.13 yang memiliki arti bahwa pesan masih asli dan dikirimkan oleh pengirim yang benar.

$$v = r \quad (2.13)$$

2.6 Electronic Prescription

E-prescribing adalah teknologi elektronik yang memungkinkan dokter dan praktisi medis lainnya untuk menulis resep elektronik (*e-prescription*) dan mengirimkannya ke komputer apotek yang dikehendaki yang tergabung dalam jaringan *e-prescribing* langsung dari tempat praktik dokter atau tempat perawatan (Coustasse dkk., 2014). *E-prescribing* menawarkan beberapa keuntungan yaitu sebagai berikut:

1. Peningkatan keselamatan pasien dan kualitas perawatan pasien
E-prescribing mengurangi kesalahan pengobatan karena kesalahan pembacaan tulisan tangan pada resep tertulis.
2. Kenyamanan pasien
E-prescribing menghemat waktu dengan menghindari perjalanan terpisah ke apotek untuk meminta resep dan menunggu di apotek untuk resep yang harus diisi.
3. Keamanan
E-prescribing lebih aman daripada resep kertas. Resep kertas sering terjadi kesalahan transkripsi dan merupakan target untuk pencurian dan gangguan. *E-prescribing* memastikan bahwa transaksi resep yang dikendalikan ditransmisikan secara aman dan terenkripsi ke penerima yang dituju.

2.7 Android

Android adalah sistem operasi seluler yang dikembangkan oleh Google, yang terutama ditujukan untuk perangkat seluler seperti *smartphone* (Jakimoski & Lazareska, 2017). Android merupakan salah satu sistem operasi yang paling berkembang, karena Android menggunakan bahasa pemrograman Java serta kelebihanannya sebagai *software* yang menggunakan basis kode komputer yang bisa didistribusikan secara terbuka (*open source*) sehingga pengguna dapat membuat aplikasi baru didalamnya. Android digunakan dalam penelitian ini karena mulai tahun 2008 sampai sekarang, Android telah mengalami banyak peningkatan yang secara bertahap meningkatkan sistem operasinya dengan menambahkan fitur-fitur baru dan memperbaiki kesalahan di versi sebelumnya. Pada tahun 2017 pengguna aktif Android perbulan lebih dari 2 miliar yang mengalami peningkatan dari tahun sebelumnya yaitu sekitar 1,5 miliar pengguna aktif. Terdapatnya fitur seperti *browser*, MMS, SMS, GPS dan lain-lain maka sangat memudahkan penggunaannya untuk mendapatkan informasi, mengetahui posisi, serta juga komunikasi antar para pengguna. Arsitektur Android adalah sebagai berikut:

1. *Applications* dan *Widgets* bekerja pada saat proses download dan instalasi.
2. *Application Frameworks* digunakan pada saat pembuatan aplikasi ataupun pengembangan aplikasi pada Android.
3. *Libraries* berisi fitur-fitur Android yang digunakan oleh pengembang aplikasi untuk menjalankan aplikasi yang dibuat.
4. *Android Run Time* adalah aplikasi yang sudah dibangun dijalankan dengan implementasi dari linux.

5. Linux Kernel merupakan bagian dari inti sistem operasi Android dan berfungsi sebagai pengatur mekanisme penyimpanan, komunikasi dengan perangkat Android lainnya.

2.8 Firebase

Firebase adalah layanan yang diberikan untuk memudahkan para pengembang untuk mengembangkan aplikasinya (Firebase, 2012). Layanan ini diberikan agar para pengembang hanya fokus pada pengembangan aplikasinya. Pada *firebase* data disinkronisasikan secara *realtime* kepada setiap klien yang terhubung dan disimpan dalam bentuk JSON. Pada penelitian ini *firebase* berfungsi sebagai penyimpanan objek kunci, pengguna dan resep. *Firebase* digunakan dalam penelitian ini karena memiliki beberapa keunggulan dari *firebase* yaitu sebagai berikut:

1. *Realtime database*
Data disinkronisasikan secara *realtime* yaitu jika setiap data berubah, maka perangkat yang terhubung menerima *update* dalam waktu yang cepat.
2. *Authentication*
Firebase memberikan pelayanan untuk autentikasi berupa email atau sandi, penyediaan pihak ketiga seperti Google atau Facebook, atau dapat menggunakan sistem akun yang sudah ada.
3. *Cloud Storage*
Penyimpanan dan membagikan gambar, audio, video, atau konten lain yang dibuat pengguna secara mudah dengan penyimpanan objek yang andal, sederhana, dan hemat biaya yang dikembangkan untuk skala Google.

2.9 Uji Hipotesis

Uji hipotesis adalah suatu metode pengambilan keputusan untuk menguji hipotesis yang ditegakkan yang berlandaskan dari analisis data yang dilakukan melalui percobaan terkontrol maupun tidak terkontrol atau observasi (Endra, 2017). Kesimpulan yang dapat ditarik dapat menggunakan rumus sebagai berikut:

- H_0 diterima jika mendapatkan nilai signifikansi lebih besar dari 0,05 dan
- H_0 tidak diterima jika mendapatkan nilai signifikansi lebih kecil dari 0,05.

Keterangan:

H_0 : Hipotesis utama.

Pengujian yang dapat dilakukan untuk menentukan apakah terdapat perbedaan dalam pengujian adalah sebagai berikut:

1. *Independent Sample T-Test*
Independent Sample T-Test merupakan uji statistik inferensial yang menentukan apakah ada perbedaan yang signifikan secara statistik antara rata-rata dalam dua kelompok yang tidak terkait (Laerd statistics, 2018). Kelompok yang tidak terkait juga disebut dengan kelompok independen. Metode ini menggunakan subjek yang digunakan harus secara acak dilakukan pada kedua kelompok.
2. *Oneway Anova*

Oneway Anova digunakan untuk menentukan ada atau tidaknya perbedaan yang signifikan secara statistik antara rata-rata dua atau lebih kelompok independen (tidak terkait) (Laerd statistics, 2018). Metode ini menggunakan subjek yang digunakan harus secara acak dilakukan pada masing-masing kelompok.

2.10 Brute Force Attack

Brute Force adalah upaya untuk menemukan kunci dengan secara sistematis mencoba setiap kombinasi huruf, angka dan simbol yang mungkin sampai menemukan kombinasi yang benar dan berfungsi (Kumar & Sowmya, 2017). Seorang penyerang selalu dapat menemukan kata sandi melalui serangan *Brute force*, tetapi dapat memakan waktu bertahun-tahun untuk menemukannya. Bergantung pada panjang dan kerumitan kunci, mungkin ada triliunan kombinasi yang memungkinkan.

Peretas melancarkan serangan *Brute Force* menggunakan *tools* yang tersedia secara luas yang memanfaatkan *wordlist* dan *smart rule* yang sudah diatur sedemikian rupa dan otomatis untuk menebak kunci tersebut. Meskipun serangan seperti itu mudah dideteksi, serangan ini tidak begitu mudah untuk dicegah.

2.11 Collision Attack

Collision attack adalah upaya untuk menemukan dua pesan dari fungsi *hash* yang menghasilkan hasil *hash* yang sama (Markov dkk, 2017). Jika dua pesan yang berbeda menghasilkan hasil *hash* yang sama disebut dengan *Collision*. Secara praktis, *Collision attack* dapat dieksploitasi. Jika penyerang menawarkan unduhan file dan menunjukkan *hash* untuk membuktikan integritas file, penyerang bisa mengganti unduhan file untuk file berbeda yang memiliki nilai *hash* yang sama, dan orang yang mengunduhnya tidak akan dapat mengetahui perbedaannya. File akan tampak valid karena memiliki *hash* yang sama dengan file yang seharusnya.

2.12 Birthday Attack

Birthday attack adalah jenis serangan kriptografi yang mengeksploitasi matematika dalam teori probabilitas (Kohno & Bellare, 2004). Serangan ini dapat digunakan untuk menyalahgunakan komunikasi antara dua pihak atau lebih. *Birthday attack* adalah jenis serangan yang hampir mirip dengan *collision attack*. *Birthday attack* merupakan jenis dari serangan kriptografi yang menggunakan matematika dibalik *birthday paradox* yang menyatakan bahwa pada grup dipilih secara acak 23 orang, maka terdapat 50% lebih peluang minimal 2 orang mempunyai ulang tahun yang sama. Keberhasilan serangan ini sangat tergantung pada kemungkinan tumbukan yang lebih tinggi yang ditemukan antara upaya serangan acak dan tingkat permutasi yang tetap.

2.13 *Black Box*

Black Box didefinisikan sebagai teknik pengujian di mana fungsionalitas *Application Under Test* (AUT) diuji tanpa melihat struktur kode internal, detail implementasi dan pengetahuan jalur internal perangkat lunak. Jenis pengujian ini didasarkan sepenuhnya pada persyaratan dan spesifikasi perangkat lunak. Dalam pengujian *Black Box* fokus pada masukan dan keluaran dari sistem perangkat lunak tanpa peduli tentang pengetahuan internal dari program perangkat lunak.

Jenis pengujian *Black Box* terdiri dari beberapa yaitu:

1. Pengujian Fungsional. Jenis pengujian *Black Box* ini terkait dengan persyaratan fungsional suatu sistem dan dilakukan oleh penguji perangkat lunak.
2. Pengujian Non-Fungsional. Jenis pengujian *Black Box* ini tidak terkait dengan pengujian fungsionalitas tertentu, tetapi persyaratan non-fungsional seperti *performance*, *usability* dan *scalability*.
3. Pengujian Regresi. Pengujian regresi dilakukan setelah perbaikan kode, peningkatan atau pemeliharaan sistem lainnya untuk memeriksa kode baru tidak memengaruhi kode yang ada.

BAB 3 METODOLOGI PENELITIAN

Bab metodologi penelitian menjelaskan langkah-langkah yang dilakukan untuk memecahkan rumusan masalah yang sudah ditentukan. Langkah-langkah tersebut dapat dilihat pada Gambar 3.1.



Gambar 3.1 Diagram Alir Metodologi Penelitian

Diagram alir metode penelitian adalah studi literatur, perancangan, implementasi, pengujian dan pembahasan, kesimpulan dan saran.

3.1 Studi Literatur

Studi literatur dilakukan untuk mempelajari teori dasar yang digunakan sebagai penunjang dan referensi penelitian ini dilakukan. Studi literatur ini bersumber dari penelitian-penelitian sebelumnya, buku, *website*, jurnal, artikel atau dokumen yang relevan dengan topik dan permasalahan pada penelitian ini. Informasi yang didapat dari studi literatur ini dapat dijadikan rujukan untuk memperkuat argumen yang ada. Studi literatur pada penelitian ini adalah *electronic prescription* (*e-prescription*), *digital signature*, DSA dan Android.

3.2 Analisis Kebutuhan

Tahap analisis kebutuhan membahas tentang kebutuhan-kebutuhan yang harus dipenuhi dalam suatu penelitian. Dalam melakukan penelitian ini ada dua kebutuhan yaitu kebutuhan perangkat lunak dan kebutuhan perangkat keras. Tahap analisis kebutuhan ini sangat penting dalam penerapan sistem karena kebutuhan-kebutuhan ini adalah sebagai dasar penerapan, agar apa yang dibuat sesuai dengan yang dibutuhkan. Analisis kebutuhan membuat kegiatan perancangan lebih efektif.

3.2.1 Kebutuhan Perangkat Keras

Perangkat keras yang dibutuhkan dalam penelitian ini adalah komputer dan *smartphone* Android. *Smartphone* Android tahun 2017 digunakan lebih dari 2 miliar pengguna *smartphone* atau setara dengan 1/3 penduduk didunia. Atas dasar ini, digunakan *smartphone* Android didalam sistem agar sistem dikembangkan bisa bermanfaat ke banyak orang. *Smartphone* Android sebagai alat yang digunakan oleh dokter dan apoteker untuk menghasilkan kunci publik dan kunci privat, melakukan tanda tangan dan melakukan verifikasi. *Smartphone* ini bekerja menggunakan Android versi 5.1 (*Lolypop*) karena pada versi ini setiap interaksi yang dilakukan lebih nyata, konten merespon sentuhan pengguna secara intuitif (Android, 2014).

Perangkat keras kedua adalah komputer yang digunakan oleh peneliti dalam melakukan penerapan sistem berbasis Android sehingga berdasarkan dokumentasi Android Developers, memiliki spesifikasi sebagai berikut: memiliki RAM 3GB, tetapi direkomendasikan menggunakan RAM 8GB, memiliki minimal 2GB ruang *disk* yang tersedia, disarankan 4GB (500GB untuk IDE + 1,5 GB untuk Android SDK), dan resolusi layar minimum 1280 x 800 (Android Developers, 2018). Komputer yang memiliki spesifikasi tersebut digunakan dalam penelitian ini karena mempermudah dalam penerapan *code* sistem sehingga prosesnya dapat berjalan dengan cepat.

3.2.2 Kebutuhan Perangkat Lunak

Perangkat lunak pada komputer yang dibutuhkan dalam penelitian ini yaitu menggunakan sistem operasi windows 7/8/10 (32 atau 64-bit), SDK Java 8, mendukung bahasa pemrograman Java sebagai bahasa pemrograman untuk menuliskan kode program dan memiliki Android Studio sebagai media pembuatan kode, Ubuntu 18.04 LTS sebagai media dalam melakukan pengujian dan SPSS 23 sebagai media untuk menghitung uji statistik. Android Studio juga terintegrasi dengan Android *Software Development Kit* (SDK) untuk *deploy* ke perangkat Android. Kebutuhan perangkat lunak lainnya adalah bahasa pemrograman Java merupakan bahasa pemrograman utama yang digunakan berisi file kode sumber Java, termasuk bahasa XML yaitu untuk membuat tampilan atau *layout*.

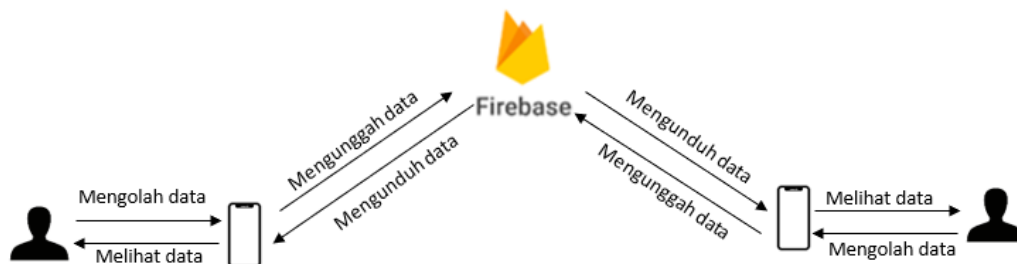
3.3 Perancangan

Perancangan pada penelitian berisi perancangan alur sistem dan perancangan pengujian. Perancangan alur sistem ini menjelaskan rincian dari rencana penerapan sistem, sedangkan perancangan pengujian menjelaskan rincian dari percobaan mekanisme penerapan sistem. Perancangan alur sistem sebagai berikut: pertama proses daftar dan masuk. Pengguna pertama kali melakukan proses daftar untuk memiliki akun dengan cara mengisi beberapa data tentang diri sendiri lalu disimpan ke *database*. Setelah memiliki akun lalu dapat masuk menggunakan nama pengguna dan kata sandi yang dipakai pada saat proses pendaftaran sebelumnya. Lalu sistem memverifikasi apakah nama pengguna dan kata sandi yang dimasukkan oleh pengguna. Jika nama pengguna dan kata sandi

benar maka pengguna dapat menggunakan sistem dan jika nama pengguna dan kata sandi salah maka pengguna menerima pemberitahuan dan pengguna tidak dapat melanjutkan proses berikutnya pada aplikasi.

Berikutnya adalah proses menghasilkan kunci DSA. Pengguna menghasilkan secara acak kunci privat dan kunci publik dengan cara pengguna menekan tombol buat kunci pada sistem kemudian pengguna menekan tombol simpan kunci untuk menyimpan kunci privat dan kunci publik ke *database* dan menampilkan pemberitahuan bahwa kunci tersebut berhasil disimpan. Proses yang ketiga adalah penandatanganan. Pengguna melakukan proses penandatanganan pada resep menggunakan kunci privat. Pengguna mengisi nama pasien, id pasien, resep, dan aturan pemakaian obat tersebut pada kotak resep. Setelah selesai menuliskan maka pengguna menekan tombol tanda tangan dan simpan maka resep tersebut disimpan dalam *database*. Kemudian dokter mengirimkan resep yang telah ditandatangani kepada apoteker untuk verifikasi.

Proses yang terakhir adalah verifikasi. Pengguna melakukan daftar atau masuk pada sistem. Apoteker melakukan verifikasi menggunakan kunci publik. Jika proses verifikasi berhasil, sistem menampilkan pesan resep valid. Arsitektur dari sistem yang dibangun dapat dilihat pada Gambar 3.2.



Gambar 3.2 Arsitektur Sistem

Arsitektur sistem terdiri dari dua aktor yaitu dokter dan apoteker, menggunakan dua perangkat *smartphone* dan menyimpan data pada *firebase*. Interaksi yang dapat dilakukan pada sistem ini adalah mengolah, melihat, mengunggah, dan mengunduh data.

3.4 Implementasi

Implementasi adalah proses pembuatan sistem sesuai dengan perancangan yang sudah disusun sebelumnya dengan merujuk pada studi literatur yang sudah dibahas sebelumnya. Langkah pertama pada penelitian ini yaitu proses implementasi daftar dan masuk untuk dapat melanjutkan proses selanjutnya pada sistem. Proses yang kedua adalah implementasi menghasilkan kunci DSA untuk menghasilkan kunci privat dan kunci publik pada sistem kemudian disimpan didalam *database*. Proses ketiga adalah pengimplementasian penandatanganan pada sistem dan dikirimkan ke apoteker. Dan yang terakhir adalah proses verifikasi

untuk menghasilkan resep yang dituliskan oleh dokter tadi kepada apoteker dengan cara verifikasi menggunakan kunci privat tersebut.

3.5 Pengujian dan Pembahasan

Pada tahap pengujian dilakukan bertujuan untuk mengetahui apakah hasil implementasinya sesuai dengan rancangan dan dapat menjawab rumusan masalah yang sudah ditetapkan. Pengujian yang dilakukan pada sistem ada enam metode pengujian yaitu pengujian *test vector*, kinerja DSA, *brute force*, *collision attack*, *birthday attack*, pengujian autentikasi dan *non-repudiation* dan *black box*. Pengujian dilakukan agar hasil akhir sistem nantinya benar-benar sudah teruji dan meminimalisir kesalahan yang mungkin terjadi di kemudian hari.

3.6 Kesimpulan dan Saran

Metode penelitian ini menghasilkan kesimpulan berdasarkan rumusan-rumusan masalah pada penelitian ini. Kesimpulan tersebut merupakan rangkuman dari pengimplementasian yang sudah dilakukan dan hasil kinerja sistem yang diimplementasikan. Metode ini juga berisikan saran-saran dari peneliti yang ditujukan ke penelitian selanjutnya yang mengembangkan sistem ini, agar penelitian selanjutnya dapat mengatasi kekurangan-kekurangan pada penelitian ini dan pengembangan dapat berjalan dengan baik.

BAB 4 PERANCANGAN

Bab perancangan menjelaskan tentang perancangan sistem yang digunakan dalam penelitian implementasi *digital signature* pada *secure electronic prescription* menggunakan DSA berbasis Android.

4.1 Gambaran Umum Sistem

Secure Eletronic Prescription merupakan sebuah aplikasi yang didesain untuk membantu proses pengiriman resep dari dokter kepada apoteker dengan menerapkan konsep *digital signature* menggunakan DSA. Aplikasi ini diimplementasikan menggunakan bahasa pemrograman Java dan berbasis Android.

Langkah awal yang dilakukan dalam pengoperasian sistem adalah daftar dan masuk. Jika pengguna belum memiliki akun maka terlebih dahulu melakukan pendaftaran dengan cara mengisi beberapa data seperti nama, nama pengguna, kata sandi, jenis kelamin, NIP, nomor telepon, dan profesi. Jika pengguna sudah memiliki akun maka pengguna dapat masuk dengan menggunakan nama pengguna dan kata sandi. Jika pengguna yang masuk ke dalam sistem adalah dokter, maka sistem menampilkan beranda dokter. Kemudian jika dokter tidak memiliki pasangan kunci yaitu kunci publik dan kunci privat, maka sistem menunjukkan opsi untuk membuat kunci. Kunci publik dan privat yang dihasilkan sistem sesuai dengan standar NIST.

Kunci publik dan kunci privat yang dihasilkan oleh sistem disimpan pada *firebase*, kemudian dokter menekan tombol lanjut. Sistem menampilkan halaman dokter resep. Dokter mengisi beberapa data yaitu nama pasien, ID pasien, *email*, dan menulis resep pada form yang sudah disediakan oleh sistem. Resep tersebut ditandatangani menggunakan kunci privat dokter dengan cara menekan tombol tanda tangan. Resep yang telah dituliskan oleh dokter disimpan ke dalam *firebase* dengan menggunakan tombol simpan dan dikirimkan kepada apoteker dengan menggunakan tombol kirim.

Kemudian resep diverifikasi apoteker dengan *smartphone* miliknya. *Smartphone* kemudian mengakses kunci publik yang telah disimpan dalam *database* kunci publik untuk proses verifikasi. Jika pesan itu asli, maka ada notifikasi resep valid dan sebaliknya, jika tidak asli maka ada notifikasi resep tidak valid.

4.2 Identifikasi Aktor

Proses implementasi sistem ini, dilakukannya identifikasi aktor yang berguna untuk memberikan gambaran siapa saja yang menggunakan sistem yang dibangun. Daftar aktor dapat dilihat pada Tabel 4.1.

Tabel 4.1 Daftar Aktor

Nama Aktor	Deskripsi Aktor
Tamu	Aktor yang menggunakan sistem namun memiliki hak terbatas dalam pengoperasian sistem.
Dokter	Aktor yang menggunakan sistem untuk melakukan proses penandatanganan.
Apoteker	Aktor yang menggunakan sistem untuk melakukan proses verifikasi.

Sistem ini terdiri dari tiga aktor yaitu Tamu pengguna sistem yang melakukan akses terbatas, dokter yang melakukan proses penandatanganan dan apoteker melakukan proses verifikasi.

4.3 Analisis Kebutuhan Sistem

Analisis kebutuhan pada sistem ini menggunakan aturan penomoran yang diambil dari nama sistem, kebutuhan sistem, aktor, dan nomor kebutuhan. Untuk nama sistem menggunakan inisial nama sistem itu sendiri, seperti *secure electronic prescription* (SEP). Setelah menggunakan nama sistem dan dilanjutkan dengan kebutuhan sistem kemudian untuk kebutuhan fungsional menggunakan kode F dan kebutuhan non-fungsional menggunakan huruf NF. Dilanjut dengan aktor, jika aktornya adalah dokter maka menggunakan kode D, jika aktornya adalah apoteker maka menggunakan kode A dan jika aktornya adalah tamu maka menggunakan kode T dan yang terakhir adalah penomoran kebutuhan.

4.3.1 Analisis Kebutuhan Fungsional

Kebutuhan fungsional merupakan kebutuhan yang mencakup segala sesuatu proses-proses yang dilakukan oleh sistem. Kebutuhan fungsional pada sistem *secure electronic prescription* dapat dilihat pada Tabel 4.2.

Tabel 4.2 Daftar Kebutuhan Fungsionalitas Tamu

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SEP-F-T-01	Daftar	Sistem harus mampu memberikan layanan untuk melakukan pendaftaran dan mendapatkan akses terhadap fungsionalitas dokter dan apoteker yang tersedia pada sistem. Pada saat pendaftaran informasi yang dibutuhkan adalah nama, <i>username</i> , kata sandi, jenis kelamin, NIP, nomor telepon, alamat dan profesi.

Tabel 4.2 Daftar Kebutuhan Fungsionalitas Tamu (lanjutan)

2	SEP-F-T-02	Masuk	Sistem harus mampu memberikan layanan untuk melakukan masuk untuk otentikasi dengan memasukkan nama pengguna dan kata sandi.
---	------------	-------	--

Tamu memiliki dua kebutuhan fungsional yaitu daftar dan masuk. Tamu wajib melakukan kedua fungsional ini agar dapat menggunakan sistem secara tidak terbatas. Daftar kebutuhan fungsionalitas Dokter dapat dilihat pada Tabel 4.3.

Tabel 4.3 Daftar Kebutuhan Fungsionalitas Dokter

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SEP-F-D-01	Buat Kunci	Sistem harus mampu memberikan layanan kepada dokter untuk menghasilkan kunci publik dan kunci privat.
2	SEP-F-D-02	Simpan Kunci	Sistem harus mampu menyimpan kunci publik dan kunci privat ke dalam <i>firebase</i> .
3	SEP-F-D-03	Lanjut	Sistem harus mampu memberikan layanan kepada dokter untuk pergi ke halaman selanjutnya.
4	SEP-F-D-04	Tanda Tangan	Sistem harus mampu memberikan tanda tangan menggunakan kunci privat pada resep yang dituliskan dokter.
5	SEP-F-D-05	Simpan	Sistem harus mampu menyimpan resep yang telah ditandatangani ke dalam <i>firebase</i> .
6	SEP-F-D-06	Kirim	Sistem harus mampu mengirim resep yang telah ditandatangani kepada apoteker.
7	SEP-F-D-07	Keluar	Sistem harus mampu memberikan layanan keluar dari sistem untuk pengguna.

Dokter memiliki tujuh kebutuhan fungsional yaitu buat kunci, simpan kunci, lanjut, tanda tangan, simpan, kirim dan keluar. Ketujuh fungsional tersebut digunakan oleh dokter untuk proses penandatanganan resep. Daftar kebutuhan fungsionalitas apoteker dapat dilihat pada Tabel 4.4.

Tabel 4.4 Daftar Kebutuhan Fungsionalitas Apoteker

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SEP-F-A-01	Verifikasi	Sistem harus mampu memverifikasi resep menggunakan kunci publik.
2	SEP-F-A-02	Kembali	Sistem harus mampu memberikan layanan untuk kembali ke halaman apoteker beranda.
3	SEP-F-A-03	Keluar	Sistem harus mampu memberikan layanan keluar dari sistem untuk pengguna.

Seorang apoteker memiliki tiga fungsional yaitu verifikasi, kembali dan keluar. Ketiga fungsi tersebut digunakan oleh apoteker untuk proses verifikasi resep yang telah ditandatangani oleh dokter.

4.3.2 Kebutuhan Non-fungsional

Kebutuhan non-fungsional merupakan kebutuhan yang menentukan kepuasan dari aktor baik dokter maupun apoteker dalam penggunaan sistem ini. Kebutuhan ini tidak diperlukan oleh masing-masing aktor secara langsung namun mempengaruhi pandangan aktor terhadap sistem. Kebutuhan non-fungsional pada sistem dapat dilihat pada Tabel 4.5.

Tabel 4.5 Daftar Kebutuhan Non-fungsional

No	Kode Fungsi	Nama Fungsi	Deskripsi
1	SEP-NF-01	<i>Usability</i>	Rancangan antarmuka mudah dimengerti oleh pengguna dalam pengoperasian sistem.
2	SEP-NF-02	<i>Avaibility</i>	Sistem dapat digunakan selama 24 jam.

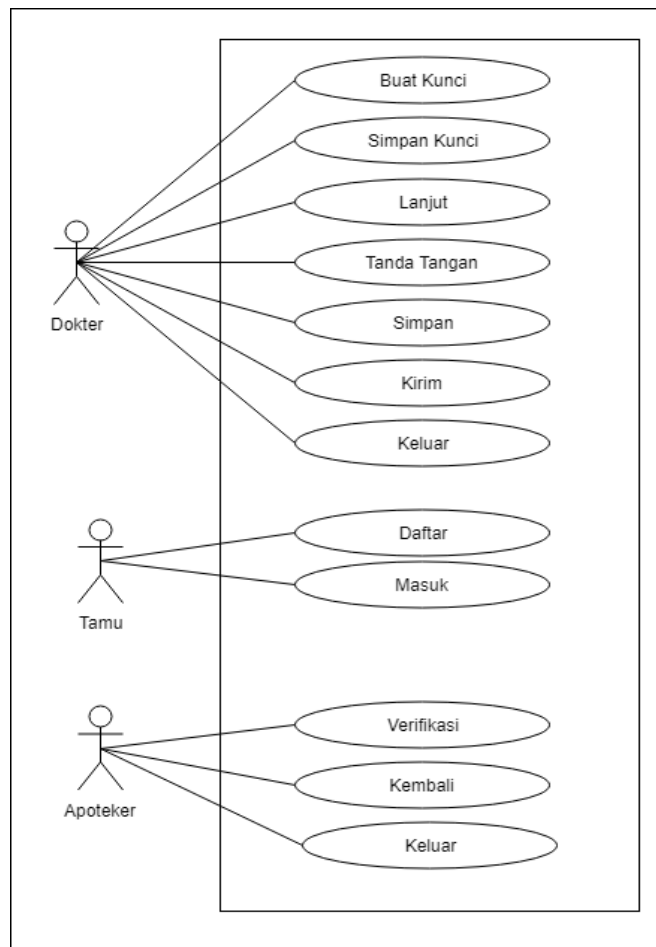
Sistem yang dibangun memiliki dua kebutuhan non-fungsional yaitu *usability* yaitu antarmuka sistem dirancang dengan mudah agar pengguna cepat mengerti pada saat penggunaan sistem inidan *avaibility* yaitu sistem dapat diakses selama 24 jam.

4.4 Pemodelan Kebutuhan

Pada bagian pemodelan kebutuhan menjelaskan tentang pemodelan kebutuhan fungsional yang telah diidentifikasi sebelumnya yaitu *use case diagram*, *use case scenario*, *sequence diagram*, *class diagram*.

4.4.1 Use Case Diagram

Use case diagram merupakan diagram yang menggambarkan aktifitas apa saja yang dilakukan aktor pada sistem yang dilihat dari luar sistem. Kebutuhan fungsional yang sudah diidentifikasi sebelumnya digambarkan pada *use case diagram*. *Use case diagram* sistem ini dapat dilihat pada Gambar 4.1.



Gambar 4.1 Use Case Diagram Electronic Prescription

Use case diagram digambarkan bahwa Tamu melakukan interaksi pada sistem yaitu daftar dan masuk. Dokter melakukan interaksi pada sistem yaitu buat kunci, simpan kunci, lanjut, tanda tangan, simpan, kirim, dan keluar. Apoteker melakukan interaksi yaitu verifikasi, kembali dan keluar.

4.4.2 Use Case Scenario

Use case scenario merupakan pemodelan kebutuhan yang menjelaskan lebih detail dari setiap kebutuhan fungsional sistem dalam bentuk tabel. *Use case scenario* daftar (SEP-F-T-01) dapat dilihat pada Tabel 4.6.

Tabel 4.6 Use Case Scenario Daftar

Kode Fungsi	SEP-F-T-01
Nama Use Case	Daftar
Deskripsi	Sistem harus mampu memberikan layanan untuk melakukan pendaftaran dan mendapatkan akses terhadap fungsionalitas dokter dan apoteker yang tersedia pada sistem.
Aktor	Tamu
Pra-Kondisi	Tamu berada dihalaman Daftar.
Tindakan	<ol style="list-style-type: none">1. Tamu memasukan data seperti nama, nama pengguna, kata sandi, jenis kelamin, NIP, nomor telepon, alamat dan profesi (dokter atau apoteker) pada halaman Daftar.2. Tamu menekan tombol daftar yang berada dibawah form pendaftaran.3. Sistem melakukan penyimpanan data pada <i>firebase</i>.
Alternatif	<ol style="list-style-type: none">1. Sistem akan menampilkan pemberitahuan jika salah satu dari data tidak diisi. Misalnya nomor telepon tidak diisi secara lengkap.2. Sistem akan menampilkan pemberitahuan jika nama pengguna yang dimasukkan sudah digunakan oleh pengguna lain.
Post-Kondisi	Penyimpanan data Tamu berhasil, menampilkan pesan “Berhasil mendaftar. Silahkan masuk” dan diarahkan ke halaman utama.

Tabel 4.6 adalah langkah-langkah melakukan daftar yang dilakukan oleh Tamu. *Use case scenario* masuk (SEP-F-T-02) dapat dilihat pada Tabel 4.7.

Tabel 4.7 Use Case Scenario Masuk

Kode Fungsi	SEP-F-G-02
Nama Use Case	Masuk
Deskripsi	Sistem harus mampu memberikan layanan untuk melakukan masuk untuk otentikasi dengan memasukkan nama pengguna dan kata sandi.

Tabel 4.7 Use Case Scenario Masuk (lanjutan)

Aktor	Tamu
Pra-Kondisi	Tamu berada pada halaman masuk
Tindakan	1. Tamu memasukkan data diri yaitu nama pengguna dan kata sandi 2. Tamu menekan tombol masuk. 3. Sistem melakukan otentikasi pengguna dengan data yang telah disimpan pada <i>firebase</i> .
Alternatif	1. Sistem akan menampilkan pemberitahuan masuk gagal jika data tidak diisi dengan lengkap. 4. Sistem akan menampilkan pemberitahuan gagal masuk jika data yang dimasukkan tidak terdaftar pada <i>firebase</i> .
Post-Kondisi	Otentikasi berhasil menampilkan pesan “Berhasil masuk” dan akan diarahkan ke halaman dokter beranda jika dokter yang melakukan proses masuk dan diarahkan ke halaman apoteker beranda jika apoteker yang masuk.

Tabel 4.7 adalah langkah-langkah melakukan masuk yang dilakukan oleh Tamu. *Use case scenario* buat kunci (SEP-G-D-01) dapat dilihat pada Tabel 4.8.

Tabel 4.8 Use Case Scenario Buat Kunci

Kode Fungsi	SEP-F-D-01
Nama Use Case	Buat Kunci
Deskripsi	Sistem harus mampu memberikan layanan kepada dokter untuk menghasilkan kunci publik dan kunci privat.
Aktor	Dokter
Pra-Kondisi	Dokter berada pada halaman dokter beranda.
Tindakan	Dokter menekan tombol buat kunci.
Alternatif	-
Post-Kondisi	Kunci publik dan kunci privat berhasil dibuat dan kedua kunci ditampilkan pada halaman dokter beranda.

Tabel 4.8 adalah langkah-langkah melakukan buat kunci yang dilakukan oleh dokter. *Use case scenario* simpan kunci (SEP-F-D-02) dapat dilihat pada Tabel 4.9.

Tabel 4.9 Use Case Scenario Simpan Kunci

Kode Fungsi	SEP-F-D-02
Nama Use Case	Simpan Kunci

Tabel 4.9 Use Case Scenario Simpan Kunci (Lanjutan)

Deskripsi	Sistem harus mampu menyimpan kunci publik dan kunci privat ke dalam <i>firebase</i> .
Aktor	Dokter
Pra-kondisi	Dokter telah mendapatkan kunci publik dan kunci privat.
Tindakan	Dokter menekan tombol simpan kunci.
Alternatif	Sistem akan menampilkan pesan “Gagal menyimpan. Kunci belum ada.” jika kunci publik dan kunci privat belum tersedia.
Post-kondisi	Kunci publik dan kunci privat berhasil disimpan pada <i>firebase</i> .

Tabel 4.9 adalah langkah-langkah melakukan penyimpanan kunci ke dalam *database* yang dilakukan oleh dokter. *Use case scenario* Lanjut (SEP-F-D-03) dapat dilihat pada Tabel 4.10.

Tabel 4.10 Use Case Scanario Lanjut

Kode Fungsi	SEP-F-D-03
Nama <i>Use Case</i>	Lanjut
Deskripsi	Sistem harus mampu memberikan layanan kepada dokter untuk melanjutkan ke halaman selanjutnya.
Aktor	Dokter
Pra-kondisi	Dokter telah mendapatkan kunci publik dan kunci privat.
Tindakan	Dokter menekan tombol lanjut
Alternatif	Sistem akan menampilkan pesan “Gagal lanjut. Kunci belum ada.” jika Kunci publik dan kunci privat belum tersedia.
Post-kondisi	Sistem telah menampilkan halaman dokter resep.

Tabel 4.10 adalah langkah-langkah melakukan untuk pergi ke halaman selanjutnya yang dilakukan oleh dokter. *Use case scenario* tanda tangan (SEP-F-D-04) dapat dilihat pada Tabel 4.11.

Tabel 4.11 Use Case Scenario Tanda Tangan

Kode Fungsi	SEP-F-D-04
Nama <i>Use Case</i>	Tanda Tangan

Tabel 4.11 Use Case Scenario Tanda Tangan (Lanjutan)

Deskripsi	Sistem harus mampu memberikan tanda tangan menggunakan kunci privat pada resep yang dituliskan dokter.
Aktor	Dokter
Pra-kondisi	Dokter telah menuliskan resep pada form resep.
Tindakan	Dokter menekan tombol Tanda tangan.
Alternatif	Sistem akan menampilkan pesan “Isi semua form” jika form yang tersedia tidak diisi oleh Dokter.
Post-kondisi	Resep yang telah dituliskan dokter telah berhasil ditandatangani.

Tabel 4.11 adalah langkah-langkah melakukan proses penandatanganan resep yang dilakukan oleh dokter. *Use case scenario* simpan (SEP-F-G-05) dapat dilihat pada Tabel 4.12.

Tabel 4.12 Use Case Scenario Simpan

Tabel Kode Fungsi	SEP-F-D-05
Nama <i>Use Case</i>	Simpan
Deskripsi	Sistem harus mampu menyimpan resep yang telah ditandatangani ke dalam <i>database</i> .
Aktor	Dokter
Pra-Kondisi	Dokter telah menuliskan resep pada form yang tersedia.
Tindakan	Dokter menekan tombol simpan.
Alternatif	Sistem akan menampilkan pesan “Resep belum ditandatangani” jika dokter belum menandatangani resep yang sudah dibuat.
Post-kondisi	Resep berhasil disimpan didalam <i>firebase</i> .

Tabel 4.12 adalah langkah-langkah melakukan penyimpanan resep ke dalam *database* yang dilakukan oleh dokter. *Use case* kirim (SEP-F-G-06) dapat dilihat pada Tabel 4.13.

Tabel 4.13 Use Case Scenario Kirim

Kode Fungsi	SEP-F-D-06
Nama <i>Use Case</i>	Kirim
Deskripsi	Sistem harus mampu mengirim resep yang telah ditandatangani kepada apoteker.

Tabel 4.13 Use Case Scenario Kirim (Lanjutan)

Aktor	Dokter
Pra-kondisi	Dokter telah mendapatkan resep yang telah ditanda tangani.
Tindakan	Dokter menekan tombol Kirim.
Alternatif	Sistem akan menampilkan pesan “Resep belum ditandatangani” jika dokter belum menandatangani sehingga tidak dapat mengirim resep tersebut.
Post-kondisi	Resep berhasil dikirimkan.

Tabel 4.13 adalah langkah-langkah melakukan pengiriman resep kepada apoteker yang dilakukan oleh dokter. *Use case* keluar (SEP-F-G-07) dapat dilihat pada Tabel 4.14.

Tabel 4.14 Use Case Scenario Keluar

Kode Fungsi	SEP-F-D-07
Nama <i>Use Case</i>	Keluar
Deskripsi	Sistem harus mampu memberikan layanan keluar untuk pengguna dapat keluar dari sistem.
Aktor	Dokter
Pra-kondisi	Dokter telah selesai mengoperasikan sistem.
Tindakan	Dokter menekan tombol Keluar.
Alternatif	-
Post-kondisi	Telah berhasil keluar dari sistem dan diarahkan ke halaman beranda.

Tabel 4.14 adalah langkah-langkah untuk keluar dari sistem yang dilakukan oleh dokter. *Use case scenario* verifikasi (SEP-F-A-01) dapat dilihat pada Tabel 4.15.

Tabel 4.15 Use Case Scenario Verifikasi

Kode Fungsi	SEP-F-A-01
Nama <i>Use Case</i>	Verifikasi
Deskripsi	Sistem harus mampu melakukan verifikasi resep menggunakan kunci publik.
Aktor	Apoteker
Pra-kondisi	Apoteker masuk dalam sistem dan telah menerima resep yang telah ditandatangani dan kunci publik dari dokter.

Tabel 4.15 Use Case Scenario Verifikasi (Lanjutan)

Tindakan	Menekan tombol Verifikasi.
Alternatif	-
Post-kondisi	Mendapatkan hasil verifikasi <i>resep</i> .

Tabel 4.15 adalah langkah-langkah melakukan verifikasi terhadap resep yang ditandatangani yang dilakukan oleh apoteker. *Use case scenario* kembali (SEP-F-A-02) dapat dilihat pada Tabel 4.16.

Tabel 4.16 Use Case Scenario Kembali

Kode Fungsi	SEP-F-A-02
Nama <i>Use Case</i>	Kembali
Deskripsi	Sistem harus mampu melakukan kembali ke halaman apoteker beranda.
Aktor	Apoteker
Pra-kondisi	Apoteker berada pada halaman apoteker resep.
Tindakan	Menekan tombol kembali.
Alternatif	-
Post-kondisi	Apoteker kembali ke halaman apoteker beranda.

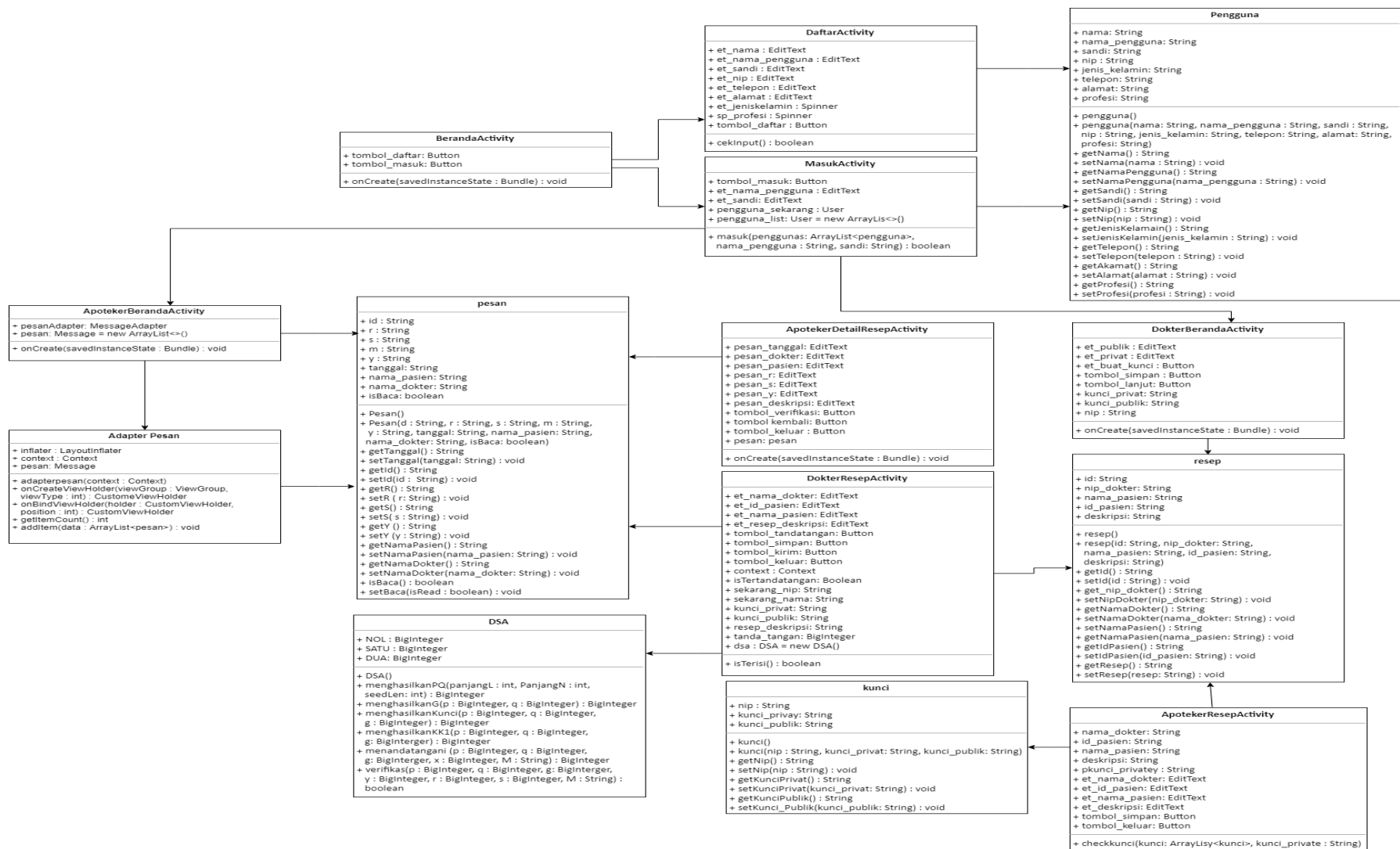
Tabel 4.16 adalah langkah-langkah melakukan kembali ke halaman apoteker beranda. *Use case scenario* keluar (SEP-F-A-03) dapat dilihat pada Tabel 4.17.

Tabel 4.17 Use Case Scenario Keluar

Kode Fungsi	SEP-F-A-03
Nama <i>Use Case</i>	Keluar
Deskripsi	Sistem harus mampu memberikan layanan keluar untuk pengguna dapat keluar dari sistem.
Aktor	Apoteker
Pra-kondisi	Apoteker telah selesai mengoperasikan sistem.
Tindakan	Apoteker menekan tombol Keluar.
Alternatif	-
Post-tindakan	Apoteker berhasil keluar dari sistem dan diarahkan ke halaman beranda.

4.4.3 Class Diagram

Class diagram adalah jenis diagram stuktur statis yang menggambarkan struktur sistem dengan menunjukkan kelas sistem, atributnya, operasi (atau metode) dan hubungan antar objek. Perancangan *class diagram* sistem ini dapat dilihat pada Gambar 4.2.



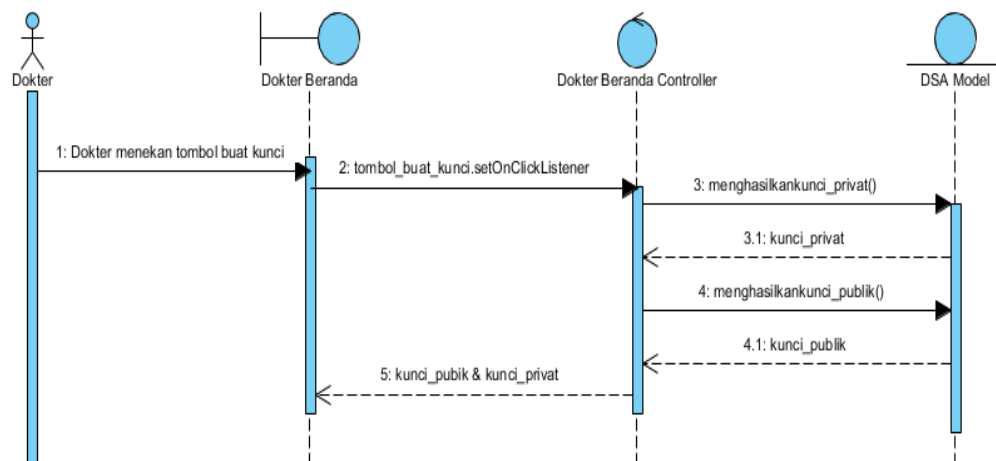
Gambar 4.2 class *diagram* terdiri dari empat model yaitu kunci, pengguna, pesan, dan resep. Terdapat sembilan *controller* yaitu beranda *activity*, daftar *activity*, masuk *activity*, dokter beranda *activity*, apoteker beranda *activity*, pesan adapter, dokter resep *activity*, apoteker resep *activity*, dan apoteker resep.

4.4.4 Sequence Diagram

Sequence diagram merupakan suatu diagram yang menggambarkan interaksi antar objek secara merinci bagaimana operasi dilakukan. *Sequence diagram* fokus pada waktu dan menunjukkan urutan interaksi secara visual dengan menggunakan sumbu vertikal *diagram* untuk mewakili waktu pesan apa yang dikirim. *Sequence diagram* pada sistem sebagai berikut:

1. Sequence Diagram Buat Kunci

Proses pembuatan kunci dilakukan oleh dokter. Proses ini dapat digambarkan pada *sequence diagram* buat kunci dapat dilihat pada Gambar 4.3.

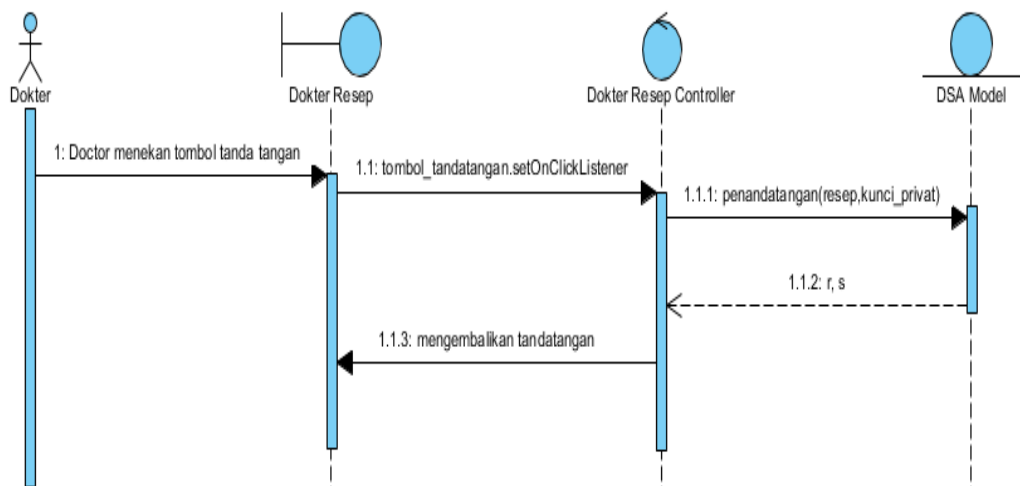


Gambar 4.3 Sequence Diagram Buat Kunci

Proses menghasilkan kunci seorang dokter menggunakan fungsi buat kunci untuk menghasilkan kunci publik dan kunci privat. Gambar 4.3 terdapat tiga objek yang saling berinteraksi yaitu dokter beranda, dokter beranda *controller* dan DSA model untuk menjalankan fungsi buat kunci.

2. Sequence Diagram Tanda Tangan

Proses tanda tangan dilakukan oleh dokter. Proses ini dapat digambarkan pada *sequence diagram* tanda tangan dapat dilihat pada Gambar 4.4.

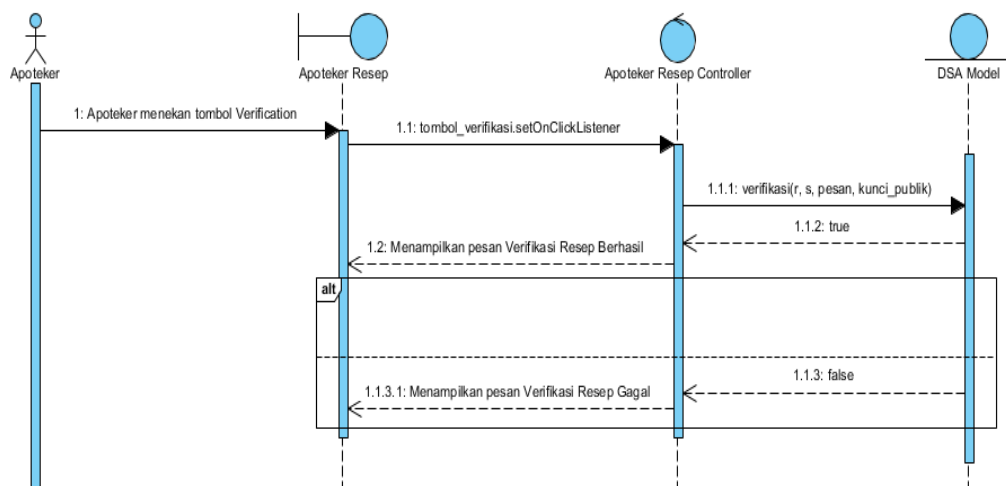


Gambar 4.4 Sequence Diagram Tanda Tangan

Proses penandatanganan resep seorang dokter menggunakan fungsi tanda tangan. Gambar 4.4 terdapat tiga objek yang saling berinteraksi yaitu dokter resep, dokter resep *controller* dan DSA model.

3. Sequence Diagram Verifikasi

Proses verifikasi dilakukan oleh apoteker. Proses ini dapat digambarkan pada *sequence diagram* verifikasi dapat dilihat pada Gambar 4.5.

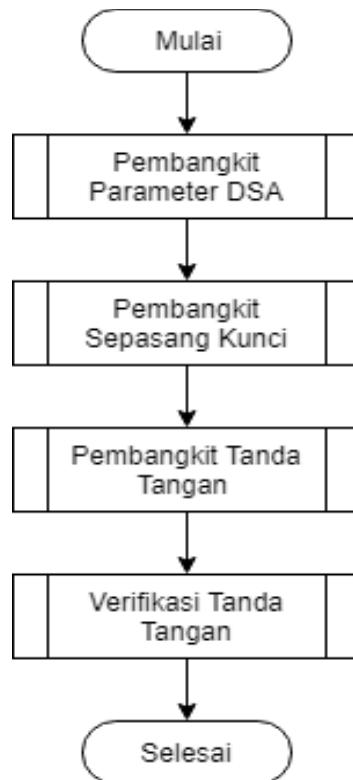


Gambar 4.5 Sequence Diagram Verifikasi

Proses verifikasi resep yang dikirimkan oleh dokter kepada apoteker dapat menggunakan fungsi verifikasi. Gambar 4.5 terdapat tiga objek yang saling berinteraksi yaitu apoteker resep, apoteker resep *controller* dan DSA Model.

4.5 Perancangan Algoritme DSA berdasarkan NIST

DSA dikembangkan dari algoritme *National Institute of Standards and Technology* (NIST). DSA dikompilasi menggunakan seperangkat parameter domain, data yang akan ditandatangani dan fungsi *hash* (NIST, 2013). Tahap perancangan algoritme dapat digambarkan dengan menggunakan diagram alir. Perancangan algoritme secara umum dapat dilihat pada Gambar 4.6.

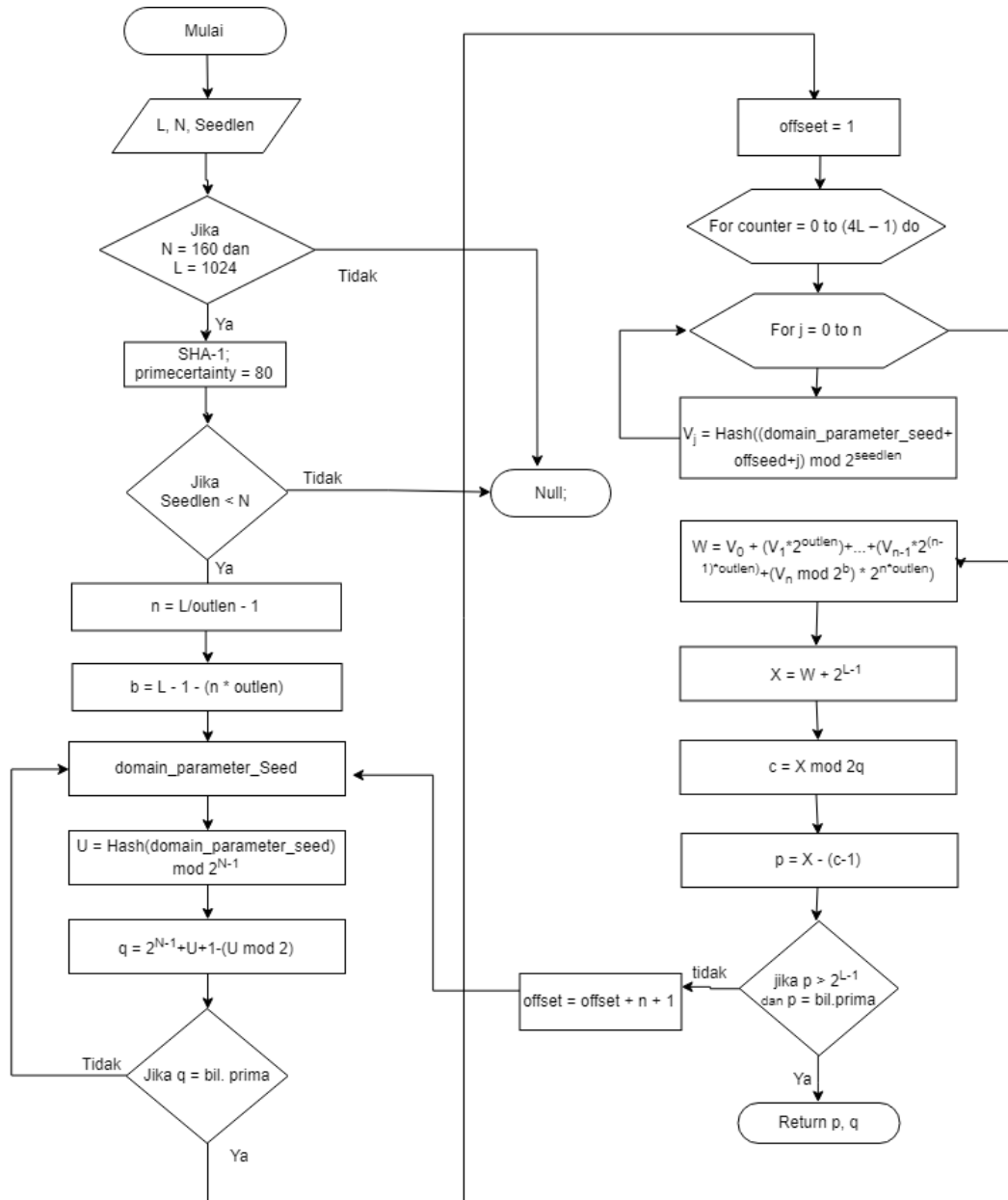


Gambar 4.6 Diagram Alir Algoritme DSA

Langkah-langkah algoritme dimulai dengan pembangkitan parameter DSA, pembangkitan sepasang kunci, penandatanganan berdasarkan parameter dan kunci yang dibangkitkan tadi dan yang terakhir adalah melakukan verifikasi tanda tangan.

4.5.1 Pembangkit Parameter DSA

Tahap pembangkit parameter DSA dilakukan untuk menghasilkan parameter p, q . Perancangan algoritme pembangkit parameter p, q dapat dilihat pada Gambar 4.7.



Gambar 4.7 Diagram Alir Pembangkit Parameter p dan q

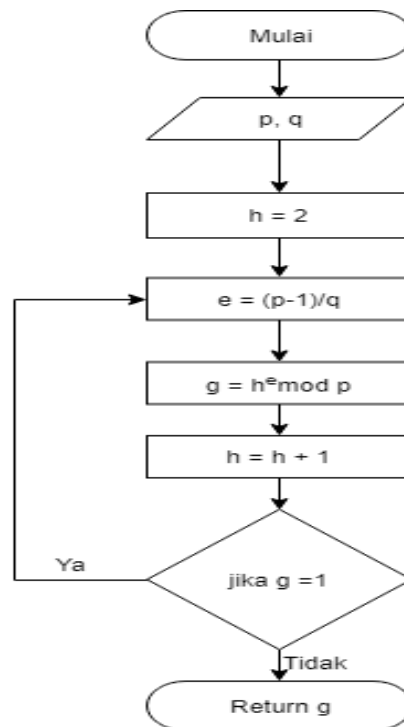
Gambar 4.7 dapat dijelaskan sebagai berikut:

1. Input $L, N, Seedlen$
2. Pastikan pasangan (L, N) ada dalam daftar pasangan (L, N) yang dapat diterima. Jika pasangan tidak ada dalam daftar, maka mengembalikan NULL. Daftar pasangan (L, N) yaitu: jika $L = 1024$ $N = 160$
3. Jika $L = 1024$ menggunakan *primecertainty* = 80 dan jika $N = 160$ menggunakan SHA-1
4. Jika $(seedlen < N)$, maka mengembalikan NULL.
5. $n = L/outlen - 1$
6. $b = L - 1 - (n * outlen)$
7. Mendapatkan nilai *domain_parameter_seed* dari nilai *arbitrary sequence*
8. $U = \text{Hash}(\text{domain_parameter_seed}) \bmod 2^{N-1}$.

9. $q = 2^{N-1} + U + 1 - (U \bmod 2)$.
10. Uji apakah q bilangan prima. Jika q tidak bilangan prima, maka kembali ke langkah 5. Jika iya lanjut ke langkah 11.
11. $offset = 1$.
12. For $counter = 0$ to $(4L - 1)$ do
 - 11.1 For $j = 0$ to n do

$$V_j = \text{Hash}((domain_parameter_seed + offset + j) \bmod 2^{seedlen}).$$
 - 11.2 $W = V_0 + (V_1 * 2^{outlen}) + \dots + (V_{n-1} * 2^{(n-1)*outlen}) + ((V_n \bmod 2^b) * 2^{n*outlen})$
 - 11.3 $X = W + 2^{L-1}$.
 - 11.4 $c = X \bmod 2q$.
 - 11.5 $p = X - (c - 1)$.
 - 11.6 Jika $(p < 2^{L-1})$, lalu lanjut kelangkah 11.7. Menguji apakah p bilangan prima. jika p ditentukan sebagai prima, maka mengembalikan VALID dan nilai-nilai p, q dan (opsional) nilai-nilai $domain_parameter_seed$ dan $counter$.
 - 11.7 $offset = offset + n + 1$.
12. Lanjut kelangkah 5.

Tahap selanjutnya adalah menghasilkan generator g . Perancangan algoritme pembangkit generator g dapat digambarkan pada Gambar 4.8.



Gambar 4.8 Diagram Alir Pembangkit parameter g

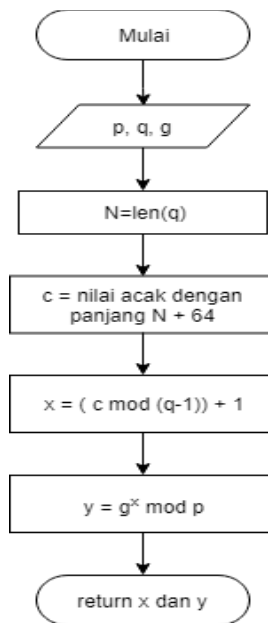
Gambar 4.8 dapat dijelaskan sebagai berikut:

1. Input p dan q .
2. $h = 2$.

3. $e = (p - 1)/q$.
4. $g = h^e \bmod p$.
5. $h = h + 1$.
6. Jika $(g = 1)$, lalu lanjut kelangkah 2.
7. Mengembalikan g .

4.5.2 Pembangkit Sepasang Kunci

Tahap pembangkit sepasang kunci dilakukan untuk menghasilkan nilai x dan y . Perancangan algoritme pembangkit sepasang kunci dapat dilihat pada Gambar 4.9.



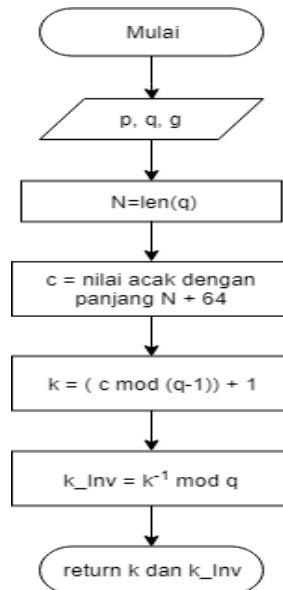
Gambar 4.9 Diagram Alir Pembangkit Sepasang Kunci

Gambar 4.9 dapat dijelaskan sebagai berikut:

1. Input p, q , dan g
2. $N = \text{panjang bit } q$
3. $C = \text{nilai acak dengan panjang } N + 64$
4. $x = (c \bmod (q - 1)) + 1$.
5. $y = g^x \bmod p$.
6. Mengembalikan x dan y .

4.5.3 Pembangkit Tanda Tangan

Tahap pembangkit tanda tangan dijelaskan proses pembangkit tanda tangan dan fungsi ini memberikan keluaran r , dan s . Sebelum menghasilkan tanda tangan terlebih dahulu dilakukan pembangkitan nomor rahasia per-pesan atau dinotasikan dengan k dan k^{-1} . Perancangan algoritme pembangkit nomor rahasia per-pesan dapat dilihat pada Gambar 4.10.

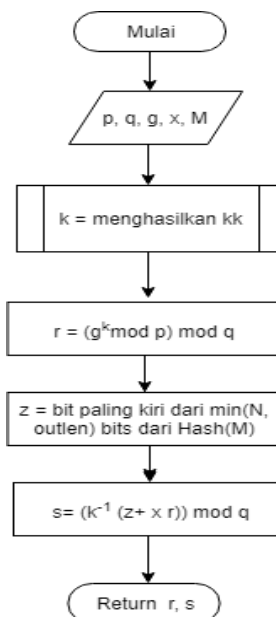


Gambar 4.10 Diagram Alir Pembangkit k dan k^{-1}

Gambar 4.10 dapat dijelaskan sebagai berikut:

1. Input p, q, g
2. $N = \text{panjang bit } q$
3. $c = \text{nilai acak dengan panjang } N + 64$
4. $k = (c \bmod (q - 1)) + 1$.
5. $K_Inv = k^{-1} \bmod q$
6. Kembalikan k dan k_Inv .

Perancangan algoritme pembangkit tanda tangan dapat dilihat pada Gambar 4.11.



Gambar 4.11 Diagram Alir Pembangkit Tanda Tangan

Gambar 4.11 dapat dijelaskan sebagai berikut:

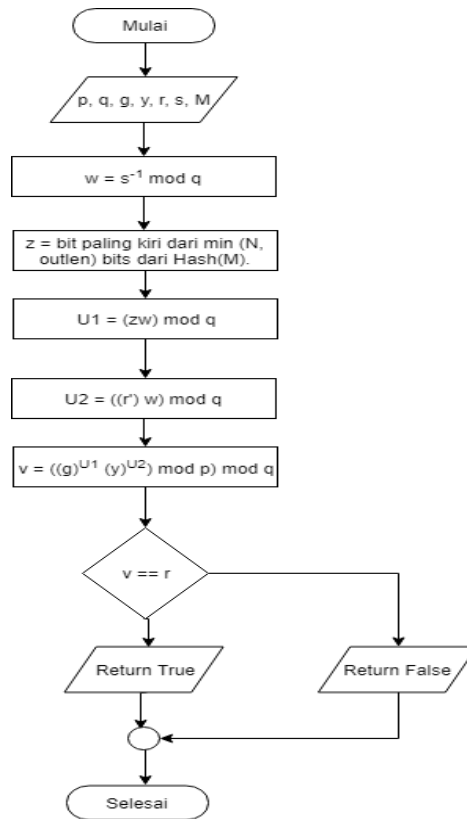
1. Masukan dari proses ini adalah p, q, g, x, M .
2. Menentukan bilangan k .
3. Setelah menemukan bilangan k maka menentukan nilai r dengan rumus sebagai berikut:

$$r = (g^k \bmod p) \bmod q$$
4. Menentukan nilai z sebagai berikut

$$z = \text{the leftmost } \min(N, \text{outlen}) \text{ bits of Hash}(M)$$
5. Menentukan nilai s dengan menggunakan rumus sebagai berikut $s = (k^{-1}(z + xr)) \bmod q$.
6. Keluaran dari proses ini adalah nilai r dan s .

4.5.4 Verifikasi Tanda Tangan

Tahap verifikasi tanda tangan dijelaskan proses verifikasi tanda tangan dengan menghasilkan nilai *true* atau *false*. Perancangan algoritme pembangkit sepasang kunci dapat dilihat pada Gambar 4.12.



Gambar 4.12 Diagram Alir Verifikasi Tanda Tangan

Gambar 4.12 dapat dijelaskan sebagai berikut:

1. Input p, q, g, y, r, s, M
2. Menentukan nilai w dengan rumus sebagai berikut: $w = s^{-1} \bmod q$
3. Menentukan nilai z dengan rumus sebagai berikut:

$$z = \text{the leftmost } \min(N, \text{outlen}) \text{ bits of Hash}(M')$$
4. Menentukan nilai U_1 menggunakan rumus sebagai berikut:

$$U_1 = (zw) \bmod q$$

5. Menentukan nilai U_2 menggunakan rumus sebagai berikut:

$$U_2 = ((r')w) \bmod q$$

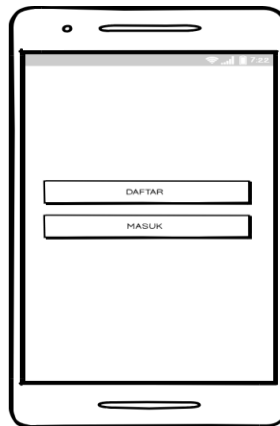
6. Menentukan nilai v menggunakan rumus sebagai berikut:

$$v = (((g^{u_1})(y)^{u_2}) \bmod p) \bmod q$$

7. Jika nilai $v == r$ maka verifikasi berhasil, dan jika nilai v tidak sama dengan r maka verifikasi gagal.
8. Output dari proses ini adalah *true* atau *false*.

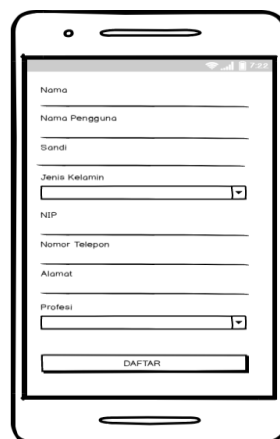
4.6 Perancangan Antarmuka

Tahap perancangan antarmuka terdiri dari halaman-halaman dari aplikasi dengan bentuk *low level design*. Tampilan rancangan antarmuka dari aplikasi *electronic prescription* sebagai berikut:



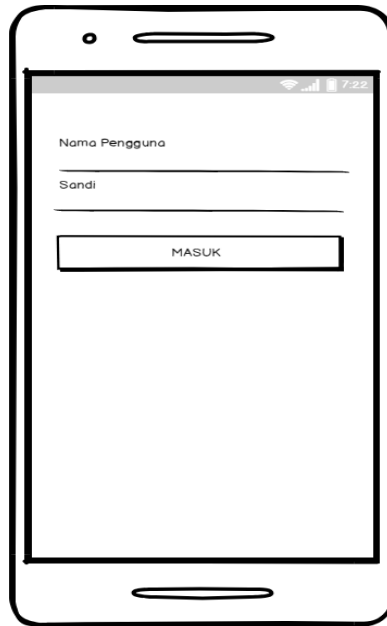
Gambar 4.13 Perancangan Antarmuka Beranda

Rancangan antarmuka beranda dapat dilihat pada Gambar 4.13. Halaman ini terdiri dari dua yaitu daftar dan masuk. Fungsi daftar dan masuk digunakan oleh Tamu untuk melakukan pendaftaran dan masuk ke dalam aplikasi.



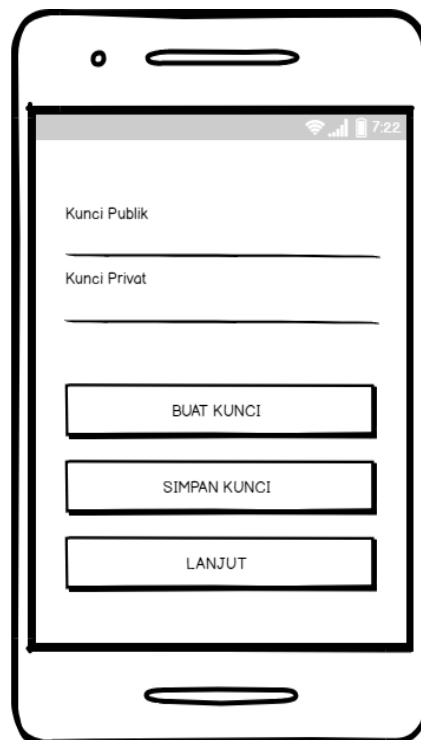
Gambar 4.14 Perancangan Antarmuka Daftar

Rancangan antarmuka daftar dapat dilihat pada Gambar 4.14. Tamu mengisi data seperti nama, nama pengguna, kata sandi, jenis kelamin, NIP, nomor telepon, alamat dan profesi pada halaman ini.



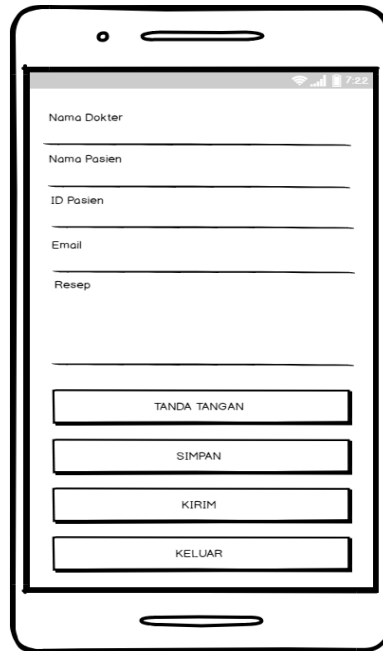
Gambar 4.15 Perancangan Antarmuka Masuk

Rancangan antarmuka Masuk dapat dilihat pada Gambar 4.15. Tamu terlebih dahulu mengisi nama pengguna dan kata sandi untuk dapat melakukan aktivitas pada sistem.



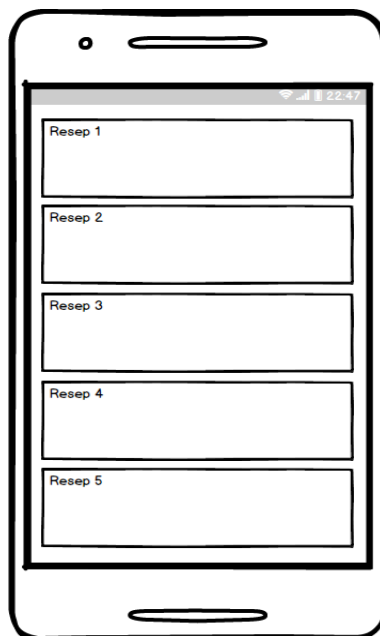
Gambar 4.16 Perancangan Antarmuka Dokter Beranda

Rancangan antarmuka dokter beranda dapat dilihat pada Gambar 4.16. Halaman ini terdiri dari tiga yaitu buat kunci, simpan kunci dan lanjut. Aktivitas dalam halaman ini hanya dapat digunakan oleh Dokter.



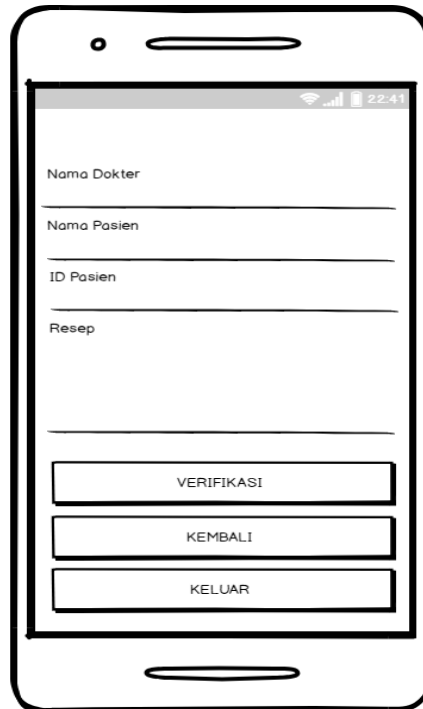
Gambar 4.17 Perancangan Antarmuka Dokter Resep

Rancangan antarmuka dokter resep dapat dilihat pada Gambar 4.17. Halaman ini terdiri dari empat yaitu tanda tangan, simpan, kirim dan keluar. Dokter menuliskan resep dan memberikan tanda tangan pada halaman ini.



Gambar 4.18 Perancangan Antarmuka Apoteker Beranda

Rancangan antarmuka apoteker beranda dapat dilihat pada Gambar 4.18. Halaman ini menampilkan resep yang dikirimkan dokter kepada apoteker.



Gambar 4.19 Perancangan Antarmuka Apoteker Resep

Rancangan antarmuka apoteker resep dapat dilihat pada Gambar 4.19. Halaman ini terdiri dari tiga fungsi yaitu verifikasi, kembali dan keluar. Halaman ini hanya dapat diakses oleh apoteker.

4.7 Perancangan Pengujian

Pengujian pada penelitian ini menggunakan beberapa metode yaitu pengujian *test vector*, kinerja DSA, *brute force*, *collision attack*, *birthday attack*, pengujian autentikasi dan *non-repudiation* dan *black box*.

4.7.1 Test Vector

Test vector dilakukan untuk melihat kebenaran DSA yang diimplementasikan sudah sesuai dengan ketentuan, di mana akan merujuk pada *test vector* oleh NIST. Tanda tangan yang dihasilkan pada DSA yang digunakan pada penelitian ini akan dibandingkan dengan *test vector* tersebut. Prosedur untuk pengujian ini dilakukan dengan memasukkan nilai p, q, g, k, x, y dan pesan pada program DSA menggunakan format *decimal* lalu menjalankan program DSA. Kemudian mencocokkan nilai tanda tangan yang dihasilkan program dengan *test vector*. Jika hasilnya sama maka DSA yang diimplementasikan sudah sesuai dengan ketentuan dan jika berbeda maka DSA yang diimplementasikan tidak sesuai dengan ketentuan.

4.7.2 Kinerja Digital Signature Algorithm

Pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan algoritme DSA dalam pembentukan dan verifikasi tanda tangan. Kemudian membandingkan kedua hasilnya untuk mengetahui apakah terjadi perbedaan yang signifikan antara

kedua proses tersebut. Prosedur pengujian ini dilakukan dengan cara menjalankan 30 kali program DSA dengan kombinasi kunci privat dan kunci publik dengan pesan yang sama. Setelah mendapatkan waktu proses tanda tangan dan verifikasi maka akan dilakukan uji statistik yaitu normalitas dan mengetahui apakah terdapat perbedaan yang signifikan pada 30 hasil tersebut.

4.7.3 Brute Force Attack

Pengujian ini dilakukan untuk mengetahui tingkat ketahanan dan kekuatan DSA dari serangan *Brute Force*. Mekanisme yang akan dilakukan adalah untuk menemukan kunci privat yang dihasilkan DSA. Pengujian ini dilakukan dengan cara menyiapkan pesan yang akan dicari kunci privatnya. Kemudian menjalankan program *brute force* sebanyak 100.000 kali percobaan. Jika kunci privat yang dihasilkan oleh *Brute Force* sama dengan kunci privat sebenarnya maka serangan yang dilakukan berhasil dan jika nilai kedua kunci berbeda maka akan dinilai gagal.

4.7.4 Collision Attack

Pengujian ini dilakukan untuk mengetahui tingkat ketahanan dan keamanan *hash function* yang digunakan dari serangan *collision*. Mekanisme yang akan dilakukan adalah untuk menemukan pada dua pesan yang berbeda namun memiliki nilai *hash* yang sama. Pengujian ini dilakukan dengan cara menyiapkan dua pesan (pesan asli dan pesan palsu) dalam bentuk pdf. Kemudian menjalankan program *collision attack* pada kedua pesan tersebut. Jika mendapatkan nilai *hash* yang sama pada kedua pesan tersebut maka serangan berhasil dan jika tidak dapat menemukan nilai *hash* yang sama maka serangan gagal.

4.7.5 Birthday Attack

Pengujian ini dilakukan untuk mengetahui tingkat ketahanan dan keamanan *hash function* yang digunakan dari *birthday attack*. Tujuan pengujian ini dilakukan adalah untuk menemukan kolisi atau beberapa pesan berbeda yang memiliki nilai *hash* yang sama. Pengujian ini dilakukan dengan cara menyiapkan pesan yang akan dicari nilai *hash* yang sama. Kemudian menjalankan program *birthday attack* pada pesan tersebut. Pada pengujian ini dilakukan pada 6 karakter pertama pada pesan, karena semakin banyak karakter yang akan dicari kolisinya maka semakin lama waktu yang dibutuhkan. Jika mendapatkan kolisi maka serangan berhasil dan jika tidak mendapatkan kolisi maka serangan gagal.

4.7.6 Pengujian Autentikasi dan Non-repudiation

Pengujian autentikasi dan *non-repudiation* dilakukan untuk memeriksa keabsahan pengirim pesan. Pengujian ini dilakukan dengan menjalankan program DSA dan membangkitkan dua pasangan kunci yaitu pasangan kunci pertama (A) dan pasangan kunci kedua (B). Pengujian dilakukan pertama kali dengan menandatangani pesan dengan menggunakan kunci privat A dan diverifikasi menggunakan kunci publik A. Pengujian selanjutnya dilakukan dengan menandatangani pesan yang sama pada pengujian pertama dengan menggunakan kunci privat A dan diverifikasi menggunakan kunci publik B.

4.7.7 Black Box

Pengujian *Black Box* dilakukan untuk mengetahui apakah sistem yang sudah dibangun sudah sesuai dengan kebutuhan fungsional dan non-fungsional yang sudah dirancang sebelumnya. Pengujian ini dilakukan dengan menggunakan *Use Case Scenario* yang sudah dirancang sebelumnya. Jika pengujian sudah sesuai dengan *Use Case scenario* maka pengujian valid dan jika tidak sesuai maka pengujiannya tidak valid. Langkah-langkah dari tiap pengujian kebutuhan dapat dilihat pada Tabel 4.18 dan Tabel 4.19.

Tabel 4.18 Pengujian *Black Box* Kebutuhan Fungsional

Nama Kasus Uji	Tindakan
Daftar	<ol style="list-style-type: none">1. Tamu mengakses halaman daftar.2. Tamu memasukkan data seperti nama, nama pengguna, kata sandi, jenis kelamin, NIP, nomor telepon, alamat dan profesi pada halaman Daftar.3. Tamu menekan tombol daftar yang berada dibawah form pendaftaran.4. Sistem melakukan penyimpanan data pada database.
Masuk	<ol style="list-style-type: none">1. Tamu berada pada halaman masuk.2. Tamu memasukkan data diri yaitu nama pengguna dan kata sandi.3. Tamu menekan tombol masuk.4. Sistem melakukan otentikasi pengguna dengan data yang telah disimpan pada <i>firebase</i>.
Buat Kunci	<ol style="list-style-type: none">1. Dokter berada pada halaman dokter beranda.2. Dokter menekan tombol buat kunci.
Simpan Kunci	<ol style="list-style-type: none">1. Dokter telah mendapatkan kunci publik dan kunci privat.2. Dokter menekan tombol simpan kunci.
Lanjut	<ol style="list-style-type: none">1. Dokter telah mendapatkan kunci publik dan kunci privat.2. Dokter menekan tombol lanjut
Lanjut	<ol style="list-style-type: none">1. Dokter telah mendapatkan kunci publik dan kunci privat.2. Dokter menekan tombol lanjut
Tanda Tangan	<ol style="list-style-type: none">1. Dokter telah menuliskan resep pada form resep.2. Dokter menekan tombol Tanda tangan.
Simpan	<ol style="list-style-type: none">1. Dokter telah menuliskan resep pada form yang tersedia.2. Dokter menekan tombol simpan.

Tabel 4.18 Pengujian *Black Box* Kebutuhan Fungsional (Lanjutan)

Nama Kasus Uji	Tindakan
Kirim	1. Dokter telah mendapatkan resep yang telah ditanda tangani. 2. Dokter menekan tombol Kirim.
Keluar	Dokter menekan tombol keluar.
Verifikasi	1. Apoteker masuk dalam sistem dan telah menerima resep yang telah ditandatangani dan kunci publik dari dokter. 2. Menekan tombol Verifikasi.
Keluar	Apoteker menekan tombol keluar.

Tabel 4.19 Pengujian *Black Box* Kebutuhan Non-fungsional

Nama Kasus Uji	Tindakan
<i>Usability</i>	Pengguna menggunakan sistem dan melakukan proses yang ada pada sistem.
<i>Avaibility</i>	Pengguna melakukan pengoperasian sistem kapan saja.

BAB 5 IMPLEMENTASI

Pada bab implementasi membahas tentang implementasi algoritme dan sistem yang telah dirancang sebelumnya pada Bab Perancangan. Pada subbagian implementasi algoritme menjelaskan tentang *pseudocode* program yang digunakan yaitu DSA. Subbagian implementasi sistem menjelaskan tentang antarmuka sistem.

5.1 Implementasi Algoritme

Subbagian implementasi algoritme menjelaskan tentang *pseudocode* program dari perancangan algoritme yang telah dibuat pada bab perancangan yaitu diagram alir pembangkit parameter p dan q , pembangkit parameter g , pembangkit kunci privat x dan kunci publik y , pembangkit k dan k^{-1} , pembangkit tanda tangan, verifikasi tanda tangan.

5.1.1 Pembangkit Parameter p dan q

DSA memberikan syarat bahwa pasangan *privat* dan kunci publik yang digunakan untuk pembuatan tanda tangan digital dan verifikasi dihasilkan dari serangkaian parameter domain tertentu. Parameter ini bersifat publik. Parameter domain untuk DSA adalah bilangan bulat p , q , dan g . Implementasi *pseudocode* program menghasilkanPQ untuk melakukan proses pembangkitan parameter p dan q dapat dilihat pada *pseudocode* 5.1.

Algoritme 1: menghasilkanPQ	
1	Input:
2	panjangL
3	panjangN
4	seedLen
5	Output:
6	hasilP
7	hasilQ
8	Proses:
9	Cek pasangan panjangL dan panjangN apakah ada dilist
10	pasangan
11	Cek panjangL untuk menentukan primeCertainty
12	Mendeklarasikan MessageDigest objekHash = null
13	try
14	Mencetak hasil error exception apabila pada proses try
15	terdapat error
16	If (seedLen < panjangN) then null
17	Mendeklarasikan outlen = panjangObjekHash * 8
18	Mendeklarasikan n = [panjangL/outlen]-1
19	Mendeklarasikan b = L - 1 - (n*outlen)
20	Mendeklarasikan domain_parameter_seed
21	Mendeklarasikan U = hash(domain_parameter_seed) mod
22	2^panjangN-1
23	Mendeklarasikan hasilQ = 2^panjangN-1 + U + 1 - (U mod 2)
24	Cek apakah hasilQ bilangan prima
25	If hasilQ tidak bilangan prima then kembali keproses
26	domain_parameter_seed
27	Mendeklarasikan offset = 1
28	for counter = 0 to (4*panjangL-1)do
29	for j=0 to n do
30	mendeklarasikan Vj =
31	Hash((domain_parameter_seed+offset+j) mod 2^seedlen)

32	mendeklarasikan $W = V_0 + (V_1 * 2^{\text{outlen}}) + \dots + (V_{n-1} * 2^{(n-1)*\text{outlen}}) + ((V_n \bmod 2^b) * 2^n * \text{outlen})$
33	
34	mendeklarasikan $X = W + 2^{\text{panjangL}-1}$
35	mendeklarasikan $c = X \bmod 2^{\text{hasilQ}}$
36	mendeklarasikan $\text{hasilP} = X - (c-1)$
37	if $(p < 2^{\text{panjangL}-1})$ then $\text{offset} = \text{offset} + n + 1$
38	cek hasilP apakah bilangan prima then mengembalikan
39	nilai hasilP dan hasilQ
40	end for
41	end for

Pseudocode 5.1 Implementasi Membangkitkan p dan q

Pseudocode 5.1 mengembalikan nilai hasilP dan hasilQ yang dibentuk dari panjangL , panjangN , dan seedlen .

5.1.2 Pembangkit Parameter g

Implementasi pseudocode program menghasilkan G untuk melakukan proses pembangkitan parameter g dapat dilihat pada pseudocode 5.2.

Algoritme 2: menghasilkanG	
1	Input:
2	hasilP , bilangan prima P
3	hasilQ , bilangan prima Q
4	Output:
5	hasilG
6	Proses:
7	Mendapatkan nilai $e = (p-1)/q$
8	Mendeklarasikan nilai $h = \text{DUA}$
9	Mendeklatasikan nilai $\text{hasilG} = \text{SATU}$
10	do
11	mendapatkan nilai $\text{hasilG} = h^e \bmod p$
12	nilai $h = h + 1$
13	while $(\text{hasilG} = 1)$
14	Mengembalikan nilai hasilG
15	end while

Pseudocode 5.2 Implementasi Pembangkit Parameter g

Pseudocode 5.2 mengembalikan nilai hasilG yang dibentuk dari hasilP , dan hasilQ .

5.1.3 Pembangkit Sepasang Kunci

Pasangan kunci privat x dan publik key y tersebut dihasilkan dari parameter p, q dan g . Implementasi pseudocode program menghasilkan kunci untuk melakukan proses pembangkitan sepasang kunci x dan y dapat dilihat pada Pseudocode 5.3.

Algoritme 3: memenghasilkanKunci	
1	Input:
2	hasilP , bilangan prima P
3	hasilQ , bilangan prima Q
4	hasilG
5	Output:
6	hasilX
7	hasilY
8	Proses:
9	Mendeklarasikan $N = \text{panjang bit } q$
10	Mendapatkan nilai c secara acak dengan $c = \text{panjangP} + 64$
11	Mendapatkan nilai $\text{hasilX} = (c \bmod (q-1)) + 1$

12	Mendapatkan nilai hasilY = $g^x \bmod p$
13	Mendeklarasikan array hasilXY yang menyimpan nilai hasilX
14	dan hasilY
15	Mengembalikan hasilXY

Pseudocode 5.3 Implementasi Pembangkit Sepasang Kunci

Pseudocode 5.3 mengembalikan nilai hasilX, hasilY yang dibentuk dari hasilP, hasilQ dan hasilG.

5.1.4 Implementasi Pembangkit Tanda Tangan

Setiap penandatanganan menggunakan pasangan kunci yaitu kunci privat x dan kunci publik y . Sebelum melakukan penandatanganan terlebih dahulu sistem menghasilkan nomor rahasia per-pesan k . Implementasi *pseudocode* program menghasilkan $kk1$ untuk melakukan proses pembangkitan k dan k' dapat dilihat pada *Pseudocode 5.4*.

Algoritme 4: menghasilkanKK1	
1	Input:
2	hasilP, bilangan prima P
3	hasilQ, bilangan prima Q
4	hasilG
5	Output:
6	hasil K
7	hasil K_Inv
8	Proses:
9	Mendeklarasikan N = panjang bit q
10	Mendapatkan nilai c secara acak dengan $c = \text{panjangP} + 64$
11	Mendapatkan nilai hasilK = $(c \bmod (q-1)) + 1$
12	Mendapatkan nilai hasilK_Inv = invers(k, q)
13	Mendeklarasikan array hasilKK yang menyimpan nilai hasilK
14	dan hasilK_Inv.
15	Mengembalikan hasilKK

Pseudocode 5.4 Pembangkit k dan k'

Pseudocode 5.4 mengembalikan nilai hasilK, hasilK_Inv yang dibentuk dari hasilP, hasilQ dan hasilG.

Implementasi *pseudocode* program menandatangani untuk melakukan proses pembangkitan tanda tangan dapat dilihat pada *Pseudocode 5.5*.

Algoritme 5: menandatangani	
1	Input:
2	hasilP, bilangan prima P
3	hasilQ, bilangan prima Q
4	hasilG
5	hasilX
6	M
7	Output:
8	hasilR, hasilS
9	Proses:
10	Mendeklarasikan array kk dengan hasil kembalian method
11	menghasilkanKK1
12	Mendeklarasikan k, dengan kk indeks ke 0 adalah k
13	Mendeklarasikan hasilR dengan $\text{hasilr} = (g^k \bmod p) \bmod q$
14	try
15	Mendapatkan nilai hasilz = bit paling kiri dari hash(M)
16	Mendeklarasikan k_inv dengan kk indeks ke 1 adalah k_inv
17	Mendeklarasikan hasilS = $(k^{-1}(z+xr)) \bmod q$

18	Mendeklarasikan array hasil Tanda Tangan yang menyimpan
19	hasilR dan hasilS
20	Mengembalikan nilai hasil Tanda Tangan
21	Catch
22	Mencetak hasil error exception apabila pada proses try terdapat
23	error dan mengembalikan nilai null

Pseudocode 5.5 Implementasi Pembangkit Tanda Tangan

Pseudocode 5.5 mengembalikan nilai hasilR, hasilS yang dibentuk dari hasilP, hasilQ, hasilG, hasilX dan M.

5.1.5 Implementasi Verifikasi Tanda Tangan

Implementasi *pseudocode* verifikasi untuk melakukan verifikasi pada pesan yang sudah ditandatangani dapat dilihat pada *Pseudocode 5.6*.

Algoritme 6: verifikasi	
1	Input:
2	hasilP, bilangan prima P
3	hasilQ, bilangan prima Q
4	hasilG
5	hasilY
6	hasilR
7	hasilS
8	M
9	Output:
10	True atau False
11	Proses:
12	Mendeklarasikan $w = s' \bmod q$
13	Try
14	Mendapatkan nilai hasilz = bit paling kiri dari hash(M)
15	Mendeklarasikan $U1 = (zw) \bmod q$
16	Mendeklarasikan $U2 = (rw) \bmod q$
17	Mendeklarasikan $v = ((g^{U1}y^{U2}) \bmod p) \bmod q$
18	Mengembalikan nilai true jika $v = r$ dan false jika $v \neq r$
19	Mencetak hasil error exception apabila pada proses try terdapat
20	error dan mengembalikan nilai false

Pseudocode 5.6 Implementasi Verifikasi

Pseudocode 5.6 mengembalikan nilai *true* atau *false* yang dibentuk dari hasilP, hasilQ, hasilG, hasilY, hasilR, hasilS dan M.

5.2 Implementasi Sistem

Implementasi sistem membahas tentang pengimplementasian antarmuka yang sudah dirancang pada bab perancangan. Antarmuka merupakan media yang digunakan sebagai perantara interaksi yang dilakukan oleh sistem dan pengguna.

5.2.1 Implementasi Antarmuka Beranda

Antarmuka beranda merupakan halaman utama pada sistem ini. Halaman ini digunakan oleh Tamu untuk memilih langkah selanjutnya yaitu daftar atau masuk.



Gambar 5.1 Implementasi Antarmuka Beranda

Gambar 5.1 adalah hasil implementasi antarmuka beranda. Aktor Tamu menggunakan halaman ini untuk mendaftar menggunakan tombol daftar dan dapat masuk ke dalam sistem menggunakan tombol masuk.

5.2.2 Implementasi Antarmuka Daftar

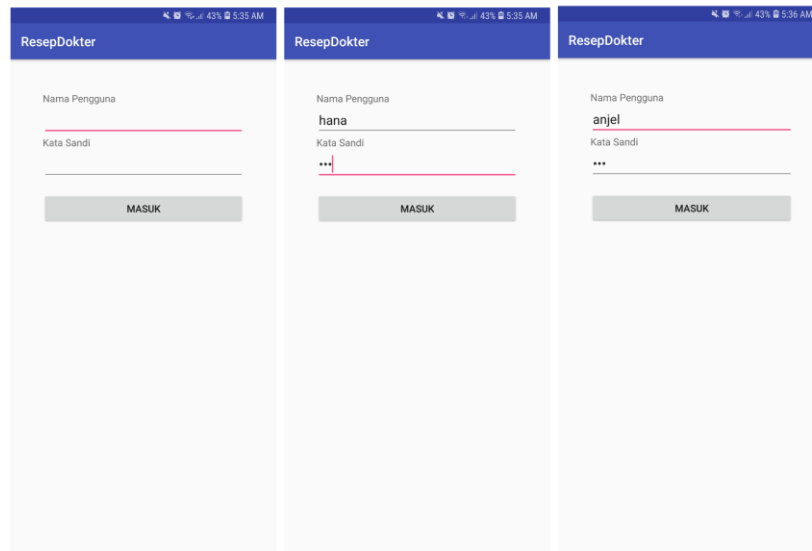
Antarmuka daftar merupakan halaman yang digunakan oleh Tamu untuk melakukan proses pendaftaran. Antarmuka daftar dapat dilihat pada Gambar 5.2.

Gambar 5.2 Implementasi Antarmuka Daftar

Proses ini Tamu memasukkan data pribadi seperti nama, nama pengguna, kata sandi, jenis kelamin, NIP, nomor handphone, alamat, dan profesi. Profesinya terdiri dari dua yaitu dokter dan apoteker.

5.2.3 Implementasi Antarmuka Masuk

Antarmuka Masuk merupakan halaman yang digunakan oleh Tamu untuk masuk ke dalam sistem, sehingga dapat mengakses sistem dengan baik. Antarmuka masuk dapat dilihat pada Gambar 5.3.

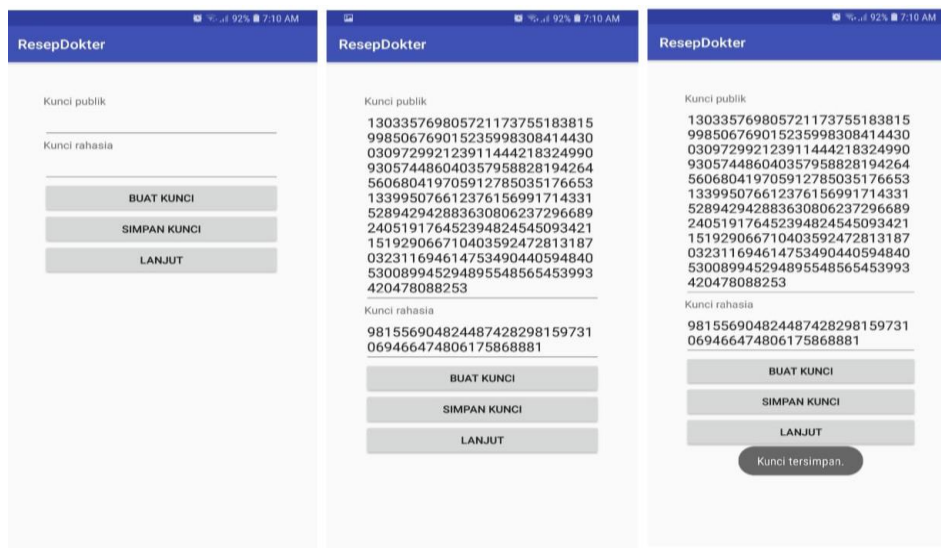


Gambar 5.3 Implementasi Antarmuka Masuk

Tamu memasukkan nama pengguna dan kata sandi sesuai dengan data yang dimasukkan pada saat pendaftaran.

5.2.4 Implementasi Antarmuka Dokter Beranda

Antarmuka dokter beranda merupakan halaman yang digunakan dokter untuk menghasilkan kunci publik dan kunci privat. Antarmuka dokter beranda dapat dilihat pada Gambar 5.4.



Gambar 5.4 Implementasi Antarmuka Dokter Beranda

Antarmuka ini seorang dokter mendapatkan kunci publik dan kunci privat dengan menggunakan tombol buat kunci. Tombol simpan kunci digunakan untuk menyimpan kunci publik dan kunci privat tersebut.

5.2.5 Implementasi Antarmuka Dokter Resep

Antarmuka dokter resep merupakan halaman yang digunakan dokter untuk menuliskan dan menandatangani resep. Antarmuka dokter resep dapat dilihat pada Gambar 5.5.

The image shows three sequential screenshots of the 'ResepDokter' application interface. Each screen has a blue header with the title 'ResepDokter'. The first two screens show a form with fields for 'Nama dokter' (filled with 'dr. Hana'), 'Nama pasien' (filled with 'Tampubolon'), and 'ID pasien' (filled with '001'). Below these is a 'Resep' field containing the text 'Paramex, konidin. 3 x 1 hari'. At the bottom of each screen are four buttons: 'TANDA TANGAN', 'SIMPAN', 'KIRIM', and 'KELUAR'. The third screenshot shows the 'KIRIM' button highlighted with a dark background and a white message 'Resep berhasil ditandatangani.' (Prescription successfully signed).

Gambar 5.5 Implementasi Antarmuka Dokter Resep (a)

The image shows two sequential screenshots of the 'ResepDokter' application interface. Both screens have a blue header with the title 'ResepDokter'. The first screen shows the same form as the previous ones, with the 'Resep' field containing 'Paramex, konidin. 3 x 1 hari'. At the bottom are four buttons: 'TANDA TANGAN', 'SIMPAN', 'KIRIM', and 'KELUAR'. The 'SIMPAN' button is highlighted with a dark background and a white message 'Resep tersimpan!' (Prescription saved!). The second screenshot shows the 'KIRIM' button highlighted with a dark background and a white message 'Resep terkirim!' (Prescription sent!).

Gambar 5.5 Implementasi Antarmuka Dokter Resep (b)

Antarmuka ini dokter memasukkan nama, ID pasien dan resep untuk menandatangani resep menggunakan tombol tanda tangan. Data tersebut disimpan ke dalam *database* menggunakan tombol simpan, dan untuk

mengirimkan resep yang telah ditandatangani kepada apoteker menggunakan tombol kirim. Tombol keluar digunakan untuk keluar dari sistem.

5.2.6 Implementasi Antarmuka Apoteker Beranda

Antarmuka apoteker beranda merupakan halaman yang digunakan oleh apoteker untuk melihat daftar resep yang telah ditandatangani dan dikirim oleh dokter kepada apoteker. Antarmuka apoteker beranda dapat dilihat pada Gambar 5.6.

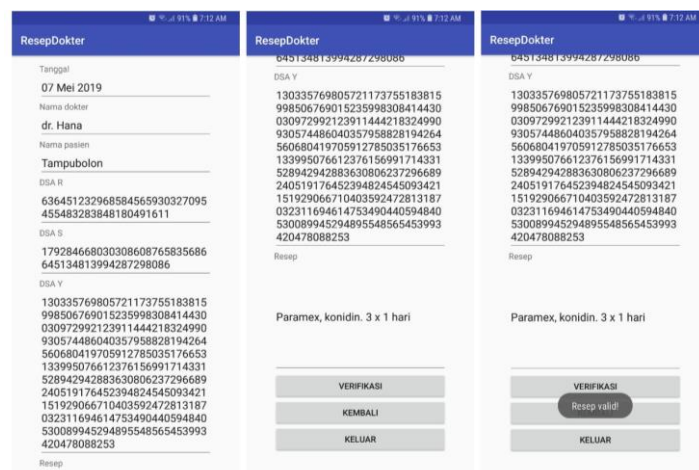


Gambar 5.6 Implementasi Antarmuka Apoteker Beranda

Antarmuka apoteker beranda merupakan resep yang dikirimkan oleh dokter yaitu tanggal, nama dokter, dan nama pasien.

5.2.7 Implementasi Antarmuka Apoteker Resep

Antarmuka apoteker resep merupakan halaman yang digunakan oleh apoteker untuk memverifikasi resep yang telah ditandatangani dokter. Antarmuka apoteker resep dapat dilihat pada Gambar 5.7.



Gambar 5.7 Implementasi Antarmuka Apoteker Resep

Antarmuka apoteker resep merupakan antarmuka yang digunakan apoteker untuk melakukan verifikasi resep yang telah ditandatangani oleh dokter.

BAB 6 PENGUJIAN DAN PEMBAHASAN

6.1 Parameter Pengujian

Parameter pengujian yang digunakan untuk algoritme DSA dengan menggunakan SHA-1 yaitu:

1. Pengujian *test vector* untuk mengetahui kebenaran DSA yang telah diimplementasikan.
2. Pengukuran waktu proses pembentukan dan verifikasi tanda tangan pada DSA.
3. Pengujian *Brute Force* yang dilakukan untuk mengetahui kunci privat yang dihasilkan DSA.
4. Pengujian *Collision attack* dan *Birthday attack* yang dilakukan untuk mendapatkan nilai *hash* yang sama dengan pesan asli
5. Pengujian autentikasi dan *non-repudiation*.
6. Pengujian *Black Box* untuk mengetahui apakah kebutuhan fungsional dan non-fungsional pada sistem sudah dengan benar diterapkan dengan baik.

6.2 Test Vector

6.2.1 Tujuan Pengujian

Test vector dilakukan untuk melihat kebenaran DSA yang diimplementasikan sudah sesuai dengan ketentuan, di mana akan merujuk pada *test vector* oleh NIST. Tanda tangan yang dihasilkan pada DSA yang akan digunakan pada penelitian ini akan dibandingkan dengan *test vector* tersebut.

6.2.2 Prosedur pengujian

Pengujian dilakukan dengan memasukkan nilai p, q, g, k, x, y dan pesan dengan format *decimal* lalu menjalankan program DSA. Kemudian mencocokkan nilai tanda tangan yang dihasilkan program dengan *test vector*. Jika hasilnya sama maka DSA yang diimplementasikan sudah sesuai dengan ketentuan dan jika berbeda maka DSA yang diimplementasikan tidak sesuai dengan ketentuan. Berikut data uji yang digunakan pada pengujian *test vector*.

Test vector untuk algoritme DSA menggunakan SHA-1.

- a. Pesan=
"241223019074170885828848249312147770083144668951633313125426
3441555184625728349100842752981595320256857031843348130170125
8473792659406180743877440206430548032768475400874609240480590
1623745428543973995391719121639368447059323814480326757639928
6337041380547069018929584628744531375841463669069864042640514
5588"
 $r = 749091408038016034262059782508984372934578116834$
 $s = 1138623541673479483228690836490483185984900562085$
- b. Pesan=
"236083129811428595082569191905046400356633099707771984855236

2317896449729156998112190710854775411296675225505340047905334
0699692139171633452951714226164109781546324578914293987249530
1055487420915395696336646575257597604518488973967029633728036
6137413751829557465738152886140878688598589765813980639227031
7416"

r = 500058060705103327052358081243797725273464289607

s = 712970052610601716224194634523957745130746386495

6.2.3 Hasil Pengujian

Hasil tanda tangan dari DSA yang digunakan pada penelitian ini.

a. Pesan=

"241223019074170885828848249312147770083144668951633313125426
3441555184625728349100842752981595320256857031843348130170125
8473792659406180743877440206430548032768475400874609240480590
1623745428543973995391719121639368447059323814480326757639928
6337041380547069018929584628744531375841463669069864042640514
5588"

r = 749091408038016034262059782508984372934578116834

s = 1138623541673479483228690836490483185984900562085

b. Pesan=

"236083129811428595082569191905046400356633099707771984855236
2317896449729156998112190710854775411296675225505340047905334
0699692139171633452951714226164109781546324578914293987249530
1055487420915395696336646575257597604518488973967029633728036
6137413751829557465738152886140878688598589765813980639227031
7416"

r = 500058060705103327052358081243797725273464289607

s = 712970052610601716224194634523957745130746386495

Tanda tangan yang digunakan pada penelitian ini sama dengan hasil *test vector*. Kesimpulannya bahwa algoritme yang digunakan sudah benar dan sesuai dengan ketentuan untuk DSA.

6.3 Kinerja *Digital Signature Algorithm*

6.3.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui waktu yang dibutuhkan algoritme DSA dalam pembentukan dan verifikasi tanda tangan. Kemudian membandingkan kedua hasilnya untuk mengetahui apakah terjadi perbedaan yang signifikan antara kedua proses tersebut.

6.3.2 Prosedur Pengujian

Prosedur pengujian ini dilakukan dengan cara menjalankan 30 kali program DSA dengan kombinasi kunci privat dan kunci publik dan pesan yang sama. Setelah

mendapatkan waktu proses tanda tangan dan verifikasi maka akan dilakukan uji statistik yaitu normalitas dan mengetahui apakah terdapat perbedaan yang signifikan pada 30 hasil tersebut. Data uji yang digunakan untuk pengujian kinerja DSA dapat dilihat pada Tabel 6.1.

Tabel 6.1 Data Uji Pengukuran Waktu

Nama Field	Isi Field
Nama Dokter	Dokter1
Nama Pasien	Hanaria Rotua
ID Pasien	1234567
Resep	Vosea 2 x sehari 1 tablet sebelum makan, Asam Mefenamat 3 x sehari 1 tablet sesudah makan

6.3.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, hasil kinerja pada saat pembentukan tanda tangan dan verifikasi pada DSA dapat dilihat pada Tabel 6.2.

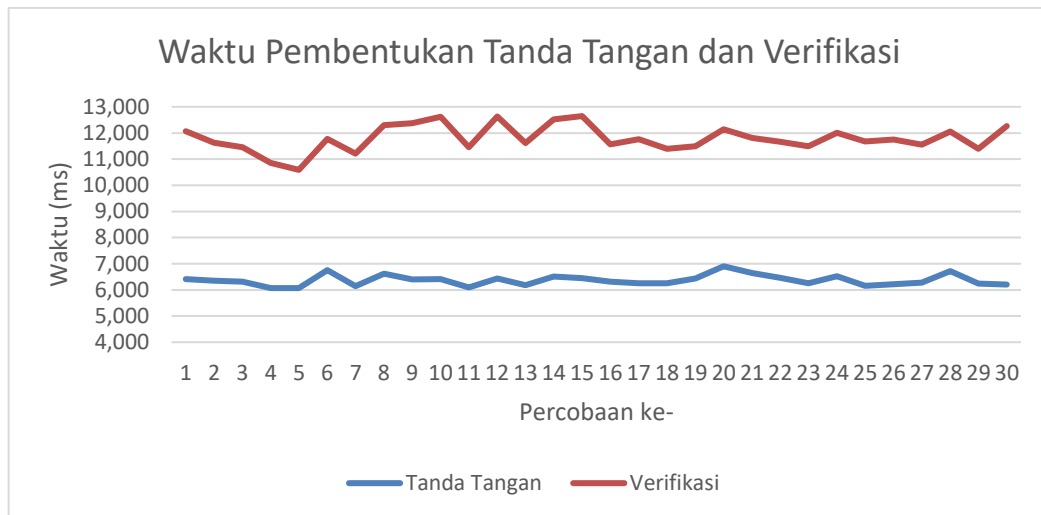
Tabel 6.2 Waktu Proses Pembentukan Tanda Tangan dan Verifikasi

No	Kecepatan (ms)	
	Tanda Tangan	Verifikasi
1	6,4158	12,0682
2	6,3558	11,6213
3	6,311	11,4556
4	6,0719	10,8609
5	6,0693	10,5896
6	6,7584	11,7783
7	6,138	11,2055
8	6,6187	12,2972
9	6,3965	12,3726
10	6,4186	12,6173
11	6,094	11,4554
12	6,436	12,6273
13	6,1825	11,6203
14	6,512	12,5217

Tabel 6.2 Waktu Proses Pembentukan Tanda Tangan dan Verifikasi (Lanjutan)

No	Kecepatan (ms)	
	Tanda Tangan	Verifikasi
15	6,4538	12,6463
16	6,3189	11,5707
17	6,2484	11,7675
18	6,2541	11,3914
19	6,4336	11,4916
20	6,9036	12,1433
21	6,6463	11,8059
22	6,466	11,6614
23	6,2571	11,4965
24	6,5184	12,0065
25	6,1528	11,6815
26	6,2209	11,7555
27	6,2774	11,5548
28	6,7227	12,0561
29	6,2359	11,3953
30	6,2005	12,26739
Rata- Rata	6,36963	11,79276

Tabel 6.2 dapat digambarkan grafik hubungan antara proses pembentukan dan verifikasi tanda tangan pada Gambar 6.1.



Gambar 6.1 Hubungan Proses Pembentukan Tanda Tangan dan Verifikasi

Berdasarkan Tabel 6.2 terdapat 30 sampel dengan rata-rata pada proses pembentukan dan verifikasi tanda tangan menggunakan DSA adalah 6,36963 ms dan 11,79276 ms. Kecepatan waktu proses pembentukan dan verifikasi tidak menentu karena saat menjalankan program tergantung kondisi komputer saat mengeksekusi program. Untuk mengetahui pada kedua proses ini terdapat perbedaan yang signifikan, maka akan dilakukan uji statistik. Pengujian yang akan dilakukan adalah normalitas dan *Independent Sample t-test*.

Tabel 6.3 Hasil Uji Normalitas Waktu Proses Tanda Tangan dan Verifikasi

	Shapiro-Wilk		
	Statistic	df	Sig.
Tanda Tangan	.953	30	.199
Verifikasi	.961	30	.332

Tabel 6.3 menjelaskan bahwa hasil uji normalitas dengan nilai signifikansi untuk tanda tangan dan verifikasi adalah 0,199 dan 0,332. Dasar pengambilan keputusan dalam uji normalitas nilai signifikan $> 0,05$ sehingga dapat disimpulkan bahwa data penelitian dalam hal ini adalah tanda tangan dan verifikasi terdistribusi normal. Karena hasilnya terdistribusi normal maka akan dilanjutkan dengan uji *independent sample t-test*.

Tabel 6.4 Hasil *Independent Sample t-test*

		Levene's Test for Equality of variances	
		F	Sig.
Waktu	Equal variances assumed	14.263	.000
	Equal variances not assumed		

		t-test for Equality of Means		
		t	df	Sig. (2-tailed)
Waktu	Equal variances assumed	-54.434	58	.000
	Equal variances not assumed	-54.434	38.852	.000

Tabel 6.4 menjelaskan bahwa hasil uji *independent t-test* dengan nilai signifikansi (2-tailed) adalah 0,000. Dasar pengambilan keputusan dalam uji *independent t-test*, jika nilai signifikansi (2-tailed) < 0,05 maka terdapat perbedaan yang signifikan. Sehingga kesimpulannya adalah terdapat perbedaan antara tanda tangan dan verifikasi.

6.4 Pengujian *Brute Force Attack*

6.4.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui tingkat ketahanan dan kekuatan DSA dari serangan *Brute Force*. Mekanisme yang akan dilakukan adalah untuk menemukan kunci privat yang dihasilkan *Brute Force*. Pengujian ini menggunakan tintinwe dengan judul DSArengK.

6.4.2 Prosedur Pengujian

Pengujian ini dilakukan dengan cara menyiapkan pesan yang akan dicari kunci privatnya. Kemudian menjalankan program *brute force* sebanyak 100.000 kali percobaan pada tiap pesan. Jika kunci privat yang dihasilkan oleh *Brute Force* sama dengan kunci privat sebenarnya maka serangan yang dilakukan berhasil, dan jika nilai kedua kunci berbeda maka akan dinilai gagal. Data uji yang digunakan pada pengujian ini dapat dilihat pada Tabel 6.5.

Tabel 6.5 Data Uji *Brute Force Attack*

No	Pesan	Private Key Sebenarnya
1	Vosea 2 x sehari 1 tablet sebelum makan	25160723710418924894621309167 5530044377868706201
2	Asam Mefenamat 3 x sehari 1 tablet sesudah makan	61037371600849733893999877417 6468066951474287561

Tabel 6.5 Data Uji Brute Force Attack (Lanjutan)

No	Pesan	Private Key Sebenarnya
3	Paracetamol 100 mg sacc lactisg 3 x sehari 1 tablet sesudah makan	18196990728140173425463932472 1036720800828933341
4	Sanmol 3 x sehari 1 tablet	92508933129094147480795554604 3552049771098670671
5	SL ad 3 x sehari 1 tablet sesudah makan	46186401957711986789486063895 7848847177957458206
6	CTM 2 mg 3 x sehari 1 tablet sesudah makan	12878483651303139355639587069 35862096649778290041
7	Lactas Calcium 300mg 3 x sehari 1 tablet sesudah makan	13318100856525124927815790691 33241593178518021172
8	Konidin 3 x sehari 1 tablet sesudah makan	16953095141123194759771758156 0855900617948459392
9	Dulcolax 5 mg 3 x sehari 1 tablet sesudah makan	66103650585803090986846589004 0368369434437527230
10	Lodia 2mg 3 x sehari 1 tablet sebelum makan	42717594638475956810405628778 24231999206792472
11	Democolin 3 x sehari 1 tablet sesudah makan	77160555334939488380465968667 8428319909943130741
12	Paratusin 3 x sehari 1 tablet sesudah makan	15403581807779059169411200167 0394122485767316018
13	Becom-C 3 x sehari 1 tablet sesudah makan	14947449810163631818272214594 3745542157811457626
14	Ibuprofen 3 x sehari 1 tablet sesudah makan	53630806128294091704895950711 6219075601733444370
15	Dumin 3 x sehari 1 sendok sesudah makan	53478434141207009463673904326 213791130233235686
16	Capl Kalmoxicil 3 x sehari 1 tablet sesudah makan	33014738687206724866831615640 9893755364033593616
17	Phenobarbital 3 x sehari 1 tablet sesudah makan	14091452706917278192763660474 08446123380513296
18	Ephedrine 5mg 3 x sehari 1 tablet sesudah makan	74895672098643265563925391142 3017819090673534615
19	Aminophylin 150mg 3 x sehari 1 tablet sesudah makan	12768208462823983915520192111 61487318755408221615

Tabel 6.5 Data Uji Brute Force Attack (Lanjutan)

No	Pesan	Private Key Sebenarnya
20	Glyceril Guaicolate 3 x sehari 1 tablet sesudah makan	420746608662512518579730216204574591308755249035
21	Codein HCL 1 x sehari 1 sendok sesudah makan	967415521082192371799449919796316178156963881912
22	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	759933873067049288368222642339171532364463993755
23	Luminal 1 x sehari 1 sendok sesudah makan	136087142697797041384974550055295967107023546513
24	Erythromycin 250mg 3 x sehari 1 tablet sesudah makan	92256278166257325862773584607726544837852990677
25	Vitamin B-compl 3 x sehari 1 tablet sesudah makan	1063045796121454963393546185323395623829333450518
26	Paramex 2 x sehari 1 tablet sesudah makan	409069281480710792729093873639487151540402465953
27	lanzoprazole 30mg 3 x sehari sesudah makan	906519658703550867104812851119835656714233369118
28	New diabetes 4 3 x sehari 1 tablet sesudah makan	969233871187509631506881696865951877945275445746
29	Sanmag sirup 120ml 2 x sehari 1 sendok sesudah makan	1086193437603521545435889756882681536664100244249
30	Vometa sirup 60 ml 3 x sehari 1 sendok sesudah makan	96163273141566654783049096883549787149202637773

6.4.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, Tabel 6.6 adalah hasil pengujian *brute force* untuk mengetahui kunci privat dari tiap pesan yang dihasilkan oleh DSA.

Tabel 6.6 Hasil Pengujian Brute Force

No	Private Key yang dihasilkan Brute Force	Status
1	936005010310112936448013239370608550362335148088	Gagal
2	751319032456342014719485765692003856927924108389	Gagal
3	832802900921477448283420709416827730295222552661	Gagal
4	883936464344470818928752504832256581153287005187	Gagal
5	558270036309731189308006014303813990704521791116	Gagal

Tabel 6.6 Hasil Pengujian *Brute Force* (Lanjutan)

No	Private Key yang dihasilkan Brute Force	Status
6	714558768493382170954559453537311648557410031499	Gagal
7	777999344430845911261188513498817391136412725869	Gagal
8	99535985684848878318542116652074891411424016956	Gagal
9	106353804978341237149206624736898303836946208547	Gagal
10	226208272062933640652386307647335017963601970938	Gagal
11	770111100556564438535198098987306688133128432491	Gagal
12	759964162486060161321084328343456639605260410910	Gagal
13	168467667480199253722825807737765511086399858374	Gagal
14	521810499310176936931046012387399005488327486807	Gagal
15	960347045818504599706489013201853402935998308326	Gagal
16	662426699425903745044278826534986493984334955488	Gagal
17	739366150681571124587073625134238966988065984134	Gagal
18	1257829535685480018837420448619903993019876259174	Gagal
19	735588674685440971614476499821640981797113420979	Gagal
20	612577389438255763090650006785341835506186413570	Gagal
21	525360647466209180402311791233971940178127448845	Gagal
22	1196430957755853565609944686475482327399308462059	Gagal
23	515844872116565712887927626745079166645938776451	Gagal
24	334660506538725599413332900845577601886499405615	Gagal
25	487833757734846887509822873807071779809401895814	Gagal
26	1068908725348054507988166505473891652078487361314	Gagal
27	1049998675256066458329500707443624699911598113759	Gagal
28	727966856523316917736773477726149719980988051480	Gagal
29	181205267650965100743522999996759936669661063862	Gagal
30	949482693265979716445707319120918838936654211053	Gagal

Tabel 6.6 merupakan hasil pengujian dari tiap pesan yang telah dilakukan sebanyak 100.000 kali percobaan pada tiap pesan. Pengujian yang telah dilakukan pada 30 pesan untuk menemukan kunci privat dan semua percobaan tersebut gagal karena kunci privat yang dihasilkan tidak sama dengan kunci privat yang sebenarnya. Untuk mengetahui jumlah kemungkinan kunci yang digunakan dapat menggunakan persamaan 6.3.

$$\text{Kemungkinan kunci} = 2^{\text{bit kunci}} \quad (6.3)$$

Mencari banyaknya percobaan dapat dihitung dengan menggunakan persamaan 6.4.

$$\text{Banyak percobaan} = 0,5 \times \text{jumlah kemungkinan kunci} \quad (6.4)$$

Mencari lama waktu percobaan dapat dihitung dengan menggunakan persamaan 6.5.

$$\text{Waktu percobaan} = \text{banyak percobaan} \times 10^6 \text{ per detik} \quad (6.5)$$

Maka jika menggunakan kunci privat dari pesan dengan ukuran 1024-bit dapat dihitung seperti pada Tabel 6.7.

Tabel 6.7 Tabel *Brute Force*

Ukuran Bit	Jumlah Kemungkinan	Banyak Percobaan	Lama Waktu
1024	2^{1024}	$0,5 \times 2^{1024}$	$(0,5 \times 2^{1024}) \times 10^6$ detik

Tabel 6.7 dapat dilihat bahwa untuk mengetahui kunci privat pesan dengan ukuran sebesar 1024-bit akan menggunakan waktu yang sangat lama. Semakin besar ukuran kunci maka akan semakin sulit dibobol oleh *brute force attack*. Kesimpulan dari hasil pengujian *brute force* telah dilakukan adalah bahwa tingkat ketahanan dan kekuatan pada DSA yang telah diterapkan dapat dinilai aman dari *brute force attack* karena semakin sulit pemecahan algoritme kuncinya maka tingkat keamanannya semakin baik.

6.5 Pengujian *Collision Attack*

6.5.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui tingkat ketahanan dan keamanan *hash function* yang digunakan dari *collision attack*. Mekanisme yang akan dilakukan adalah untuk menemukan pada dua pesan yang berbeda namun memiliki nilai *hash* yang sama. Pengujian ini menggunakan cr-marcstevens dengan judul *sha1collisiondetection*.

6.5.2 Prosedur Pengujian

Pengujian ini dilakukan dengan cara menyiapkan dua pesan (pesan asli dan pesan palsu) dalam bentuk pdf. Kemudian menjalankan program *collision attack* pada kedua pesan tersebut. Jika mendapatkan nilai *hash* yang sama pada kedua pesan tersebut maka serangan berhasil dan jika tidak dapat menemukan nilai *hash* yang sama maka serangan gagal. Data uji yang digunakan pada pengujian ini dapat dilihat pada Tabel 6.8.

Tabel 6.8 Data uji *Collision Attack*

No	Pesan Asli	Pesan Palsu
1	Vosea 2 x sehari 1 tablet sebelum makan	Alprazolam 3 x sehari 1 tablet sebelum makan
2	Asam Mefenamat 3 x sehari 1 tablet sesudah makan	Amfetamin 3 x sehari 1 tablet sebelum makan
3	Paracetamol 100 mg sacc lactisg 3 x sehari 1 tablet sesudah makan	Amitriptyline 2 x sehari 1 tablet sesudah makan
4	Sanmol 3 x sehari 1 tablet	Haloperidol 3 x sehari 1 tablet
5	SL ad 3 x sehari 1 tablet sesudah makan	Hydroquinone 3 x sehari 1 tablet sesudah makan
6	CTM 2 mg 3 x sehari 1 tablet sesudah makan	Memantine 3 x sehari 1 tablet sebelum makan
7	Lactas Calcium 300mg 3 x sehari 1 tablet sesudah makan	Metoprolol 3 x sehari 1 tablet sesudah makan
8	Konidin 3 x sehari 1 tablet sesudah makan	Mycophenolate Sodium 3 x sehari 1 tablet sesudah makan
9	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	Glibenclamide 3 x sehari 1 tablet sesudah makan
10	Lodia 2mg 3 x sehari 1 tablet sebelum makan	Guaifenesin 2mg 3 x sehari 1 tablet sebelum makan
11	Democolin 3 x sehari 1 tablet sesudah makan	Neurobion 3 x sehari 1 tablet sebelum makan
12	Paratusin 3 x sehari 1 tablet sesudah makan	Nifedipine 3 x sehari 1 tablet sebelum makan
13	Becom-C 3 x sehari 1 tablet sesudah makan	Nystatin 2 x sehari 1 tablet sesudah makan
14	Ibuprofen 3 x sehari 1 tablet sesudah makan	Noscapine 3 x sehari 1 tablet
15	Dumin 3 x sehari 1 sendok sesudah makan	Nitrogen Oksida 3 x sehari 1 tablet sesudah makan
16	Capl Kalmoxicil 3 x sehari 1 tablet sesudah makan	Nevirapine 3 x sehari 1 tablet sebelum makan
17	Phenobarbital 3 x sehari 1 tablet sesudah makan	Nicotinamide 3 x sehari 1 tablet sesudah makan

Tabel 6.8 Data uji *Collision Attack* (Lanjutan)

18	Ephedrine 5mg 3 x sehari 1 tablet sesudah makan	Dekongestan 3 x sehari 1 tablet sesudah makan
19	Aminophylin 150mg 3 x sehari 1 tablet sesudah makan	Dextrose 3 x sehari 1 tablet sesudah makan
20	Glyceril Guaicolate 3 x sehari 1 tablet sesudah makan	Diazepam 2mg 3 x sehari 1 tablet sebelum makan
21	Codein HCL 1 x sehari 1 sendok sesudah makan	Diltiazem 3 x sehari 1 tablet sebelum makan
22	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	Diuretik 3 x sehari 1 tablet sebelum makan
23	Luminal 1 x sehari 1 sendok sesudah makan	Dumolid 2 x sehari 1 tablet sesudah makan
24	Erythromycin 250mg 3 x sehari 1 tablet sesudah makan	Donepezil 3 x sehari 1 tablet
25	Vitamin B-compl 3 x sehari 1 tablet sesudah makan	Disulfiram 3 x sehari 1 tablet sesudah makan
26	Paramex 2 x sehari 1 tablet sesudah makan	Digoxin 3 x sehari 1 tablet sebelum makan
27	lansoprazole 30mg 3 x sehari sesudah makan	Metoprolol 3 x sehari 1 tablet sesudah makan
28	New diabetes 4 3 x sehari 1 tablet sesudah makan	Quimidine 3 x sehari 1 tablet sesudah makan
29	Sanmag sirup 120ml 2 x sehari 1 sendok sesudah makan	Quinolone 3 x sehari 1 tablet sebelum makan
30	Vometa sirup 60 ml 3 x sehari 1 sendok sesudah makan	Ketorolac 3 x sehari 1 tablet sesudah makan

6.5.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, pada Tabel 6.9 merupakan hasil pengujian *Collision Attack* yang dilakukan untuk mengetahui tingkat ketahanan dan keamanan *hash function* yang digunakan.

Tabel 6.9 Hasil Pengujian *Collision Attack*

No	Hasil	Status
1	1318851794806383075210380683479861367934140406031	Berhasil

Tabel 6.10 Hasil Pengujian *Collision Attack* (Lanjut)

No	Hasil	Status
2	7969584811395566443676782834016820249644871 1420	Berhasil
3	1035140168816235963372235982818115660080633 156719	Berhasil
4	1025881074100821786641329368424644887874712 130777	Berhasil
5	1350487010696669011912780839223416827400091 208049	Berhasil
6	6042608753361910376849526580616966440801479 37871	Berhasil
7	7262574315648407582761334983060285180469453 52308	Berhasil
8	3913291048923669594261423938833510144281344 33405	Berhasil
9	4395095780128732983079064170761279864209810 07541	Berhasil
10	2208740192659387242006467969986498616212831 19135	Berhasil
11	1354381122867637627549614111397702368156380 511151	Berhasil
12	1117937686876798281585862777064481285541458 958293	Berhasil
13	4882033135536023049741934986920249046789242 23977	Berhasil
14	9634975001691579428810690391045728899580565 3854	Berhasil
15	4769007680391174957462835105154014523995944 28805	Berhasil
16	1346924800193987465251818356603668119911884 857284	Berhasil
17	1200035296016186324754885701250556989065082 699640	Berhasil
18	1053887974348337813073280903811023977030370 502469	Berhasil

Tabel 6.10 Hasil Pengujian *Collision Attack* (Lanjut)

No	Hasil	Status
19	9160391825912115563916093915039036425510224 64430	Berhasil
20	1344376817677386018718900529942240130770880 621878	Berhasil
21	3380648001173837662673296427094115108697263 45302	Berhasil
22	4122306856463608074102091734234556608267301 97902	Berhasil
23	9363702505159706463318721846844399082524177 27865	Berhasil
24	1364684535964477676094121755030525599345244 934951	Berhasil
25	8005747546851490219974033656719037430654015 95893	Berhasil
26	5990352286939665064567794139375200510239088 4404	Berhasil
27	9662524316802279764943232841499108660549052 84938	Berhasil
28	1273697118811797348334222413422979108396793 779968	Berhasil
29	7253726518030649266407490082511874629958164 07050	Berhasil
30	4548399416100383265114082593626297494575086 66859	Berhasil

Tabel 6.9 merupakan hasil pengujian dari *collision attack* yang telah dilakukan pada 30 pesan untuk menemukan nilai *hash* yang sama antara kedua pesan. Pengujian yang telah dilakukan semuanya berstatus berhasil. Kesimpulan dari hasil pengujian *collision attack* bahwa tingkat ketahanan dan keamanan pada *hash function* yang diterapkan dapat dinilai tidak aman atau dengan kata lain aspek integriti pada pesan tersebut belum terpenuhi.

6.6 Pengujian *Birthday Attack*

6.6.1 Tujuan Pengujian

Pengujian ini dilakukan untuk mengetahui tingkat ketahanan dan keamanan *hash function* yang menggunakan *birthday attack*. Tujuan pengujian ini dilakukan

adalah untuk menemukan kolisi atau beberapa pesan berbeda yang memiliki nilai *hash* yang sama. Pengujian ini menggunakan kode yang dibuat oleh Vagradez dengan judul *Birthday Attack: Cryptographic hash function Collisions*.

6.6.2 Prosedur Pengujian

Pengujian ini dilakukan dengan cara menyiapkan pesan yang akan dicari nilai *hash* yang sama. Kemudian menjalankan program *birthday attack* pada pesan tersebut. Pada pengujian ini dilakukan pada 6 karakter pertama pada pesan, karena semakin banyak karakter yang akan dicari kolisinya maka semakin lama waktu yang dibutuhkan. Jika mendapatkan kolisi maka serangan berhasil dan jika tidak mendapatkan kolisi maka serangan gagal. Data uji yang digunakan pada pengujian ini dapat dilihat pada Tabel 6.10.

Tabel 6.10 Data Uji *Birthday Attack*

No	Pesan
1	Vosea 2 x sehari 1 tablet sebelum makan
2	Asam Mefenamat 3 x sehari 1 tablet sesudah makan
3	Paracetamol 100 mg sacc lactisg 3 x sehari 1 tablet sesudah makan
4	Sanmol 3 x sehari 1 tablet
5	SL ad 3 x sehari 1 tablet sesudah makan
6	CTM 2 mg 3 x sehari 1 tablet sesudah makan
7	Lactas Calcium 300mg 3 x sehari 1 tablet sesudah makan
8	Konidin 3 x sehari 1 tablet sesudah makan
9	Ephidrib HCL 1 x sehari 1 sendok sesudah makan
10	Lodia 2mg 3 x sehari 1 tablet sebelum makan
11	Democolin 3 x sehari 1 tablet sesudah makan
12	Paratusin 3 x sehari 1 tablet sesudah makan
13	Becom-C 3 x sehari 1 tablet sesudah makan
14	Ibuprofen 3 x sehari 1 tablet sesudah makan
15	Dumin 3 x sehari 1 sendok sesudah makan
16	Capl Kalmoxicil 3 x sehari 1 tablet sesudah makan
17	Phenobarbital 3 x sehari 1 tablet sesudah makan
18	Ephedrine 5mg 3 x sehari 1 tablet sesudah makan
19	Aminophylin 150mg 3 x sehari 1 tablet sesudah makan
20	Glyceril Guaicolate 3 x sehari 1 tablet sesudah makan
21	Codein HCL 1 x sehari 1 sendok sesudah makan

Tabel 6.11 Data Uji *Birthday Attack* (Lanjutan)

No	Pesan
22	Ephidrib HCL 1 x sehari 1 sendok sesudah makan
23	Luminal 1 x sehari 1 sendok sesudah makan
24	Erythromycin 250mg 3 x sehari 1 tablet sesudah makan
25	Vitamin B-compl 3 x sehari 1 tablet sesudah makan
26	Paramex 2 x sehari 1 tablet sesudah makan
27	lanzoprazole 30mg 3 x sehari sesudah makan
28	New diabetes 4 3 x sehari 1 tablet sesudah makan
29	Sanmag sirup 120ml 2 x sehari 1 sendok sesudah makan
30	Vometa sirup 60 ml 3 x sehari 1 sendok sesudah makan

6.6.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, Tabel 6.11 adalah hasil pengujian *birthday attack* yang dilakukan untuk mengetahui tingkat ketahanan dan keamanan *hash funtion* yang digunakan.

Tabel 6.11 Hasil Pengujian *Birthday Attack*

No	Hasil	Status
1	11761228	Berhasil
2	10934888	Berhasil
3	2092024	Berhasil
4	5217629	Berhasil
5	15528290	Berhasil
6	14827724	Berhasil
7	1525216	Berhasil
8	15442752	Berhasil
9	13996717	Berhasil
10	2283562	Berhasil
11	709633	Berhasil
12	7851568	Berhasil
13	095404	Berhasil
14	10128907	Berhasil

Tabel 6.11 Hasil Pengujian *Birthday Attack* (Lanjutan)

No	Hasil	Status
15	12410077	Berhasil
16	881221113	Berhasil
17	13514257	Berhasil
18	3071024	Berhasil
19	141419106	Berhasil
20	1415141010	Berhasil
21	613127515	Berhasil
22	135921013	Berhasil
23	121061412	Berhasil
24	43615514	Berhasil
25	10107079	Berhasil
26	8cd125	Berhasil
27	7111314	Berhasil
28	7141250	Berhasil
29	115810511	Berhasil
30	111510338	Berhasil

Tabel 6.11 merupakan hasil pengujian yang telah dilakukan pada 30 pesan untuk menemukan kolisi pada masing-masing pesan. Pengujian yang telah dilakukan berstatus berhasil. Kesimpulan dari hasil pengujian *birthday attack* bahwa tingkat ketahanan dan keamanan pada *hash function* yang diterapkan dapat dinilai tidak aman atau dengan kata lain aspek integrity belum terpenuhi.

6.7 Pengujian Autentikasi dan *Non-repudiation*

6.7.1 Tujuan Pengujian

Pengujian autentikasi dan *non-repudiation* dilakukan untuk memeriksa keabsahan pengirim pesan.

6.7.2 Prosedur Pengujian

Pengujian ini dilakukan dengan menjalankan program DSA dan membangkitkan dua pasangan kunci yaitu pasangan kunci pertama (A) dan pasangan kunci kedua (B). Pengujian dilakukan pertama kali dengan menandatangani pesan dengan menggunakan kunci privat A dan diverifikasi menggunakan kunci publik A. Pengujian selanjutnya dilakukan dengan

menandatangani pesan yang sama pada pengujian pertama dengan menggunakan kunci privat A dan diverifikasi menggunakan kunci publik B. Data uji yang digunakan pada pengujian ini dapat dilihat pada Tabel 6.12.

Tabel 6.12 Data Uji Pengujian Autentikasi

	Kunci Publik	Kunci Privat
A	13041485363067170036145688912893154338220237685 77944972545621259639820721637638765628540158425 45292140869360720276066427965708784714229173834 51367362788435171430671937066304065678211195099 29271585947641149361313813184534321389340348012 07837911580529855250249248808462654833603782258 933517505919782404782018855	25160723710 41892489462 13091675530 04437786870 6201
B	1258799350473226224844432313308264039020722669 8860149389273366169620555208185169487464258649 7099270672654830702638466073679852212373669711 1169021085284734446119850718213777420846957189 3417184326136883013984639419614607346337811971 0000812903868756734977543299341165089261907713 037046143751695442504420605103505	77160555334 93948838046 59686678428 31990994313 0741

6.7.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, berikut hasil pengujian autentikasi dapat dilihat pada Tabel 6.13.

Tabel 6.12 Hasil Pengujian Autentikasi

Nama Kasus Uji	Percobaan ke-	Status
Autentikasi dan Non-repudation	1	Valid
Autentikasi dan Non-repudation	2	Tidak Valid

Tabel 6.13 merupakan hasil pengujian autentikasi dimana pengujian pertama valid yang memiliki makna bahwa tanda tangan tersebut sah dibuat oleh pemilik aslinya yaitu pemilik kunci privat A maka orang yang menuliskan pesan tersebut tidak dapat menyangkal bahwa dialah yang menulis pesan tersebut. Pengujian kedua tidak valid yang memiliki makna bahwa tanda tangan tersebut tidak sah atau dengan kata lain bahwa orang yang menandatangani dengan orang yang memverifikasi berbeda. Maka hasil pengujian ini menunjukkan bahwa aspek autentikasinya dan *non-repudiation* sudah terpenuhi.

6.8 Pengujian *Black Box*

6.8.1 Tujuan Pengujian

Pengujian *Black Box* dilakukan untuk mengetahui apakah sistem yang sudah dibangun sudah sesuai dengan kebutuhan fungsional dan non-fungsional yang sudah dirancang sebelumnya.

6.8.2 Prosedur Pengujian

Pengujian ini dilakukan dengan menggunakan *Use Case Scenario* yang sudah dirancang sebelumnya. Jika pengujian sudah sesuai dengan *Use Case scenario* maka pengujian valid dan jika tidak sesuai maka pengujiannya tidak valid.

6.8.3 Hasil Pengujian

Berdasarkan pengujian yang telah dilakukan, berikut hasil pengujian *black box* pada sistem ini dapat dilihat pada Tabel 6.14.

Tabel 6.13 Hasil Pengujian *Black Box* Kebutuhan Fungsional

Nama Kasus Uji	Tindakan	Hasil yang Diharapkan	Hasil	Status
Daftar	<ol style="list-style-type: none">1. Tamu mengakses halaman daftar.2. Tamu memasukkan data seperti nama, nama pengguna, kata sandi, jenis kelamin, NIP, nomor telepon, alamat dan profesi pada halaman Daftar.3. Tamu menekan tombol daftar yang berada dibawah form pendaftaran.4. Sistem melakukan penyimpanan data pada database.	Penyimpanan data Tamu berhasil, menampilkan pesan “Berhasil Mendaftar” dan diarahkan ke halaman utama.	Penyimpanan data Tamu berhasil, menampilkan pesan “Berhasil Mendaftar” dan diarahkan ke halaman utama.	Valid

Tabel 6.14 Hasil Pengujian *Black Box* Kebutuhan Fungsional (Lanjutan)

Nama Kasus Uji	Tindakan	Hasil yang Diharapkan	Hasil	Status
Masuk	<ol style="list-style-type: none"> 1. Tamu berada pada halaman masuk. 2. Tamu memasukkan data diri yaitu nama pengguna dan kata sandi. 3. Tamu menekan tombol masuk. 4. Sistem melakukan otentikasi pengguna dengan data yang telah disimpan pada database. 	Otentikasi berhasil menampilkan pesan “Berhasil Masuk” dan akan diarahkan kehalaman dokter beranda jika dokter yang melakukan proses masuk dan diarahkan ke halaman beranda jika apoteker yang masuk.	Otentikasi berhasil menampilkan pesan “Berhasil Masuk” dan akan diarahkan kehalaman dokter beranda jika dokter yang melakukan proses masuk dan diarahkan ke halaman beranda jika apoteker yang masuk.	Valid
Buat Kunci	<ol style="list-style-type: none"> 1. Dokter berada pada halaman dokter beranda. 2. Dokter menekan tombol buat kunci. 	Kunci publik dan kunci privat berhasil dibuat dan kedua kunci ditampilkan pada halaman dokter beranda.	Kunci publik dan kunci privat berhasil dibuat dan kedua kunci ditampilkan pada halaman dokter beranda.	Valid
Simpan Kunci	<ol style="list-style-type: none"> 1. Dokter telah mendapatkan kunci publik dan kunci privat. 2. Dokter menekan tombol simpan kunci. 	Kunci publik dan kunci privat berhasil disimpan pada database.	Kunci publik dan kunci privat berhasil disimpan pada database.	Valid
Lanjut	<ol style="list-style-type: none"> 1. Dokter telah mendapatkan kunci publik dan kunci privat. 2. Dokter menekan tombol lanjut 	Sistem telah menampilkan halaman dokter resep.	Sistem telah menampilkan halaman dokter resep.	Valid

Tabel 6.14 Hasil Pengujian *Black Box* Kebutuhan Fungsional (Lanjutan)

Nama Kasus Uji	Tindakan	Hasil yang Diharapkan	Hasil	Status
Kirim	1. Dokter telah mendapatkan resep yang telah ditanda tangani. 2. Dokter menekan tombol Kirim.	Resep berhasil dikirimkan.	Resep berhasil dikirimkan.	Valid
Keluar	Dokter menekan tombol keluar.	Telah berhasil keluar dari sistem dan diarahkan ke halaman beranda.	Telah berhasil keluar dari sistem dan diarahkan ke halaman beranda.	Valid
Verifikasi	1. Apoteker masuk dalam sistem dan telah menerima resep Yang ditandatangani dan kunci publik dari dokter. 2. Menekan tombol Verifikasi.	Mendapatkan hasil verifikasi resep.	Mendapatkan hasil verifikasi resep.	Valid
Keluar	Apoteker menekan tombol keluar.	Apoteker berhasil keluar dari sistem dan diarahkan ke halaman beranda.	Apoteker berhasil keluar dari sistem dan diarahkan ke halaman beranda.	Valid

Tabel 6.14 pengujian kebutuhan fungsional dengan pengujian *black box* terdapat dua belas pengujian dan semuanya berstatus valid dan dengan demikian sistem yang dibangun sudah memenuhi kebutuhan fungsional yang sudah dirancang. Pengujian black box pada kebutuhan non-fungsional, dapat dilihat pada Tabel 6.15.

Tabel 6.14 Hasil Pengujian *Black Box* Kebutuhan Non-fungsional

Nama Kasus Uji	Tindakan	Hasil yang Diharapkan	Hasil	Status
<i>Usability</i>	Pengguna menggunakan sistem dan melakukan proses yang ada pada sistem.	Pengguna berhasil dan mudah mengerti saat pengoperasian Sistem.	Pengguna berhasil dan mudah mengerti saat pengoperasian sistem.	Valid
<i>Avaibility</i>	Pengguna melakukan pengoperasian sistem kapan saja.	Pengguna dapat menggunakan sistem kapan saja.	Pengguna dapt menggunakan sistem kapan saja.	Valid

Tabel 6.15 pengujian kebutuhan non-fungsional dengan black box testing terdapat dua pengujian dan semuanya berstatus valid dan dengan demikian sistem yang dirancang sudah memenuhi kebutuhan non-fungsional yang sudah dirancang.

BAB 7 PENUTUP

Bab Penutup menjelaskan tentang kesimpulan yang telah didapatkan dari semua proses yang telah dilakukan. Selain dari kesimpulan juga akan diberikan saran yang akan digunakan jika penelitian ini dikembangkan lagi.

7.1 Kesimpulan

Berdasarkan semua proses yang telah dilakukan, adapun kesimpulan untuk penelitian yang dilakukan adalah:

1. Dalam mengimplementasikan DSA pada *secure electronic prescription* berbasis Android terdiri dari beberapa cara yaitu pertama melakukan implementasi hasil perancangan analisis kebutuhan sistem seperti kebutuhan fungsional dan non-fungsional. Kemudian dilakukan implementasi dari perancangan pemodelan kebutuhan sistem seperti *use case diagram*, *use case scenario*, *class diagram*, *sequence diagram*. Setelah melakukan perancangan sistem, kemudian dilakukan implementasi dari perancangan algoritme yaitu pembangkit parameter DSA, pembangkit sepasang kunci, pembangkit tanda tangan, dan verifikasi tanda tangan. Dan yang terakhir adalah implementasi dari perancangan antarmuka yaitu antarmuka beranda, daftar, masuk, dokter beranda, dokter resep, apoteker beranda dan apoteker resep. Setelah melakukan beberapa cara ini maka dapat disimpulkan bahwa implementasi DSA pada *secure electronic prescription* telah berhasil.
2. Kinerja implementasi *digital signature* menggunakan DSA pada *secure electronic prescription* terdiri dari beberapa pengujian yaitu:
 - a. Pengujian *test vector* telah dilakukan dan hasil dari implementasi tanda tangan sama dengan hasil *test vector*.
 - b. Pengujian kinerja dilakukan pada satu pesan menggunakan kunci privat dan kunci publik yang sama dengan 30 kali percobaan menghasilkan rata-rata waktu untuk membentuk tanda tangan dan verifikasi adalah 6,36963 ms dan 11,79276 ms.
 - c. Pengujian *black box* telah dilakukan dan semua percobaan yang dilakukan berhasil, sehingga sistem yang dibangun dinilai baik dan sudah sesuai dengan rancangan kebutuhan fungsional dan non-fungsional.
3. Keamanan implementasi *digital signature* menggunakan DSA pada *secure electronic prescription* terdiri dari beberapa pengujian yaitu:
 - a. Pengujian *brute force* telah dilakukan dan semua percobaan yang dilakukan gagal, sehingga kunci privat yang dihasilkan oleh DSA masih aman.
 - b. Pengujian *collision attack* dan *birthday attack* telah dilakukan dan semua percobaan yang dilakukan berhasil, sehingga *hash function* yang digunakan yaitu SHA-1 tidak aman digunakan atau aspek integrity belum terpenuhi.
 - c. Pengujian autentikasi dan *non-repudiation* telah dilakukan dan kedua aspek ini telah terpenuhi.

7.2 Saran

Berdasarkan kesimpulan yang telah dibuat, adapun saran untuk penelitian yang akan dilakukan selanjutnya adalah:

1. Sisi kewananan dalam aspek autentikasi dan integritas perlu ditingkatkan dengan melakukan proses penandatanganan dan verifikasi menggunakan DSA dengan *hash function* SHA-224 atau SHA-256, karena *hash function* yang digunakan SHA-1 masih sangat rentan terhadap serangan *collision attack* dan *birthday attack*.

Perlu dilakukannya dalam pengembangan dan penambahan kebutuhan fungsional dalam penerapan sistem *e-prescription* berbasis Android, karena sistem ini hanya memiliki kebutuhan fungsional yang hanya fokus pada proses penandatanganan dan verifikasi.

DAFTAR REFERENSI

- Andorid Developers, 2018. *Android Studio*. [online] Tersedia di: <<https://developer.android.com/studio>> [Diakses 20 April 2019]
- Android Developers, 2018. *Mengenal Android Studio*. [online] Tersedia di: <<https://developer.android.com/studio/intro/?hl=id>> [Diakses 13 Januari 2019]
- Android, 2014. *Android 5.0, Lollipop*. [online] Tersedia di: <<https://www.android.com/versions/lollipop-5-0/>> [Diakses 20 April 2019]
- Ariyus. D., 2006. *Kriptografi: Keamanan Data dan Komunikasi*. 1st ed. Graha Ilmu.
- Coustasse, A., Engelbert, K., Portefield, A., 2014. *Electronic Prescribing: Improving the Efficiency and Accuracy of Prescribing in the Ambulatory Care Setting*, [e-journal] Tersedia melalui: US National Library of medicine National Institutes of Health <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3995494/>> [Diakses pada 23 Desember 2018]
- Firebase, 2012. *Firebase Database*. [online] Tersedia di: <<https://firebase.google.com/?hl=id>> [Diakses 20 April 2019]
- Hellman M., Diffie W., 1976. *New Directions in Cryptography*. IEEE Transactions on Information Theory. [Online] Tersedia di: <<https://ieeexplore.ieee.org/document/1055638>> [Diakses pada 30 Mei 2019]
- Institute for Safe Medication Practices Canada (ISMP Canada), 2007. *Definition of Terms*. [online] Tersedia di: <<https://www.ismp-canada.org/definitions.htm>> [Diakses 28 Maret 2019]
- Institute of Medicine (IOM), 2006. *Preventing Medication Errors*, [e-journal] 31(12):8. Tersedia melalui: US National Library of Medicine National Institutes of Health <<https://www.ncbi.nlm.nih.gov/pubmed/17149128>> [Diakses pada 7 Januari 2019]
- Jakimoski K., Lazareska L., 2017. *Analysis of the Adbantages and Disadvantages of Android and iOS Systems and Converting Applications from Android to iOS Platform and Vice Versa* [e-journal] 6(5). Tersedia melalui: <<http://www.sciencepublishinggroup.com/j/ajsea>> [Diakses 22 April 2019]
- Kohno T., Bellare M., 2004. *Hash Function Balance and Its Impact on Birthday Attacks*. [Online] Tersedia di: <https://link.springer.com/chapter/10.1007/978-3-540-24676-3_24> [Diakses pada 30 Mei 2019]
- Kumar, N., & Sowmya, G., A., 2013. *Brute Force Attack – Blocking Technique*. International Journal of Engineering and Computer Science Science, [online] Tesedia di: <<http://www.ijecs.in/index.php/ijecs/article/view/1810>> . [Diakses 01 Mei 2019]

- Laerd Statistics, 2018. *Independent t-test for two samples*, [online] Tersedia di: <<https://statistics.laerd.com/statistical-guides/independent-t-test-statistical-guide.php>> [Diakses 20 Juni 2019]
- Laerd Statistics, 2018. *One-way ANOVA in SPSS Statistics*, [online] Tersedia di: <<https://statistics.laerd.com/spss-tutorials/one-way-anova-using-spss-statistics.php>> [Diakses 20 Juni 2019]
- Lesyk V., Dragoni N., Conti M., 2016. *A Survey of Man In The Middle Attacks*. IEEE Communication Survey & Tutorials [e-journal] 18(3). Tersedia melalui: <<https://ieeexplore.ieee.org/document/7442758>> [Diakses 18 Juni 2019]
- Markov Y., dkk, 2017. *The first collision for full SHA-1*. [online] Tersedia di: <https://link.springer.com/chapter/10.1007/978-3-319-63688-7_19> [Diakses pada 30 Mei 2019]
- Minuz, P., Velo, G., P., 2009. *Medication errors: prescribing faults and prescription errors*. [e-journal] 67(6): 624-628. Tersedia melalui: US National Library of Medicine National Institutes of Health <<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2723200/>> [Diakses pada 7 Januari 2019]
- Mukhopadhyay, D., Lohani, K., 2017. *Reduction of Medication Errors While Prescribing Using Evidence Based Treatment*. ieeexplore digital library. [online] Tersedia di: <<https://ieeexplore.ieee.org/document/8318747>> [Diakses pada 5 januari 2019]
- National Institute of Standard and Technology (NIST). 2013. *Digital Signature Standard*. [Online] Tersedia di: <<https://csrc.nist.gov/publications/detail/fips/186/4/final>> [Diakses pada 3 April 2019].
- Noviyanto, F., Nugroho D. Y., 2013. *Membangun Sistem Pembuatan Resep Obat untuk Mencegah Pemalsuan Dengan Teknik Code Generator Berbasis Web*. [online]. Tersedia di: <<http://journal.uad.ac.id/index.php/JSTIF/article/view/2518>> [Diakses 28 Maret 2019]
- Ohio Board Pharmacy, 2018. *Meet Ohio Prescription Verification and Indication Requirements in PCC eRx*. [online] PCC Learn. Tersedia di: <<http://learn.pcc.com/help/meet-ohio-prescription-requirements-pcc-erx/>> [Diakses 28 Maret 2019]
- Schneier, B., 1996. *Applied Cryptography Second Edition*. [e-book] Tersedia di: <<https://archive.org/details/AppliedCryptographyBruceSchneier>> [Diakses 20 Februari 2019]
- Sentosa Y., 2011. *Tanda Tangan Digital pada E-Resep untuk Mencegah Pemalsuan Resep Dokter dan sebagai Media Anti Penyangkalan Dokter*. [online] Tersedia di: <<http://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi/2010->

- 2011/Makalah2/Makalah2-IF3058-Sem2-2010-2011-010.pdf> [Diakses 28 Maret 2019].
- Sivaraman, K., 2006. *A Comparision study of RSA and DSA Algorithm in Mobile Cloude Computing*. International Journal of Pure and Applied Mathematics [online] Tersedia di: <<https://acadpubl.eu/jsi/2017-116-8/articles/8/42.pdf>> [Diakses pada 24 Januari 2019].
- Steven Marc, 2017. *SHA1 Collision Detection*. [online] Tesedia di: <<https://github.com/cr-marcstevens/sha1collisiondetection>> [Diakses 26 Mei 2019]
- Suganya K., Ramya K., 2013. *Design and Implementation of Digital Signatures*. International Journal of Engineering Research & Technology [Online] Tesedia di: <<https://www.ijert.org/research/design-and-implementation-of-digital-signatures-IJERTV2IS120633.pdf>> [Diakses pada 4 April 2019]
- Sunaringtyas, S.U., Sadikin, M.A., 2016. *Implementing Digital Signature for the Secure Electronic Prescription Using QR-code Base on Android smartphone*. ieeexplore digital library [online] Tersedia di: <<https://ieeexplore.ieee.org/abstract/document/7873856>> [Diakses 13 Desember 2018]
- Susanti, K., 2013. *Identifikasi Medication Error pada Fase Prescribing, Transcribing, dan Dispensing di Depo Farmasi Rawat Inap Penyakit Dalam Gedung Teratai, Instalasi Farmasi RSUP Fatmawati Periode 2013*. S1. UIN SYARIF HIDAYATULLAH JAKARTA. Tersedia di: <<http://repository.uinjkt.ac.id/dspace/bitstream/123456789/26463/1/IKA%20SUSANTI-FKIK.pdf>> [Diakses 16 Januari 2019]
- Tintinwe. 2013. *DSAregeK*. [online]. Tersedia di: <<https://github.com/tintinweb/DSAregeK>> . [Diakses 13 Mei 2019]
- Tintinweb, 2012. *DSA Regen K*. [online] Tersedia di: <<https://github.com/tintinweb/DSAregeK>> [Diakses 26 Mei 2019]
- Wahyuni A., 2011. *Aplikasi Kriptografi untuk Pengamanan E-Dokumen dengan Metode Hybrid : Biometrik Tandatangan dan DSA (Digital Signature Algorithm)*. [Online] Tersedia di: <<https://core.ac.uk/download/pdf/11728847.pdf>> [Diakses pada 29 Maret 2019]
- Vagradez, 2014. *Birthday Attack: Cryptographic Hash Functions Collisions*. [online] <<https://github.com/Vagradez/multi-algocollisions/blob/master/README.md>> [Diakses 26 Mei 2019]
- Widiyanto, A., 2007. *Meningkatkan Keamanan Komputer Anda*. Semarang: Neomedia Press.

LAMPIRAN A HASIL PENGUJIAN

A.1 Hasil Pengujian *Brute Force*

No	Pesan	Private Key Sebenarnya	Private Key yang dihasilkan Brute Force
1	Vosea 2 x sehari 1 tablet sebelum makan	25160723710418924894 62130916755300443778 68706201	93600501031011293644 80132393706085503623 35148088
2	Asam Mefenamat 3 x sehari 1 tablet sesudah makan	61037371600849733893 99987741764680669514 74287561	75131903245634201471 94857656920038569279 24108389
3	Paracetamol 100 mg sacc lactisg 3 x sehari 1 tablet sesudah makan	18196990728140173425 46393247210367208008 28933341	83280290092147744828 34207094168277302952 22552661
4	Sanmol 3 x sehari 1 tablet	92508933129094147480 79555460435520497710 98670671	88393646434447081892 87525048322565811532 87005187
5	SL ad 3 x sehari 1 tablet sesudah makan	46186401957711986789 48606389578488471779 57458206	55827003630973118930 80060143038139907045 21791116
6	CTM 2 mg 3 x sehari 1 tablet sesudah makan	12878483651303139355 63958706935862096649 778290041	71455876849338217095 45594535373116485574 10031499
7	Lactas Calcium 300mg 3 x sehari 1 tablet sesudah makan	13318100856525124927 81579069133241593178 518021172	77799934443084591126 11885134988173911364 12725869
8	Konidin 3 x sehari 1 tablet sesudah makan	16953095141123194759 77175815608559006179 48459392	99535985684848878318 54211665207489141142 4016956
9	Dulcolax 5 mg 3 x sehari 1 tablet sesudah makan	66103650585803090986 84658900403683694344 37527230	14086935721123657662 45888436803107251796 1283789
10	Lodia 2mg 3 x sehari 1 tablet sebelum makan	42717594638475956810 40562877824231999206 792472	22620827206293364065 23863076473350179636 01970938

11	Democolin 3 x sehari 1 tablet sesudah makan	77160555334939488380 46596866784283199099 43130741	77011110055656443853 51980989873066881331 28432491
12	Paratusin 3 x sehari 1 tablet sesudah makan	15403581807779059169 41120016703941224857 67316018	75996416248606016132 10843283434566396052 60410910
13	Becom-C 3 x sehari 1 tablet sesudah makan	14947449810163631818 27221459437455421578 11457626	16846766748019925372 28258077377655110863 99858374
14	Ibuprofen 3 x sehari 1 tablet sesudah makan	53630806128294091704 89595071162190756017 33444370	52181049931017693693 10460123873990054883 27486807
15	Dumin 3 x sehari 1 sendok sesudah makan	53478434141207009463 67390432621379113023 3235686	96034704581850459970 64890132018534029359 98308326
16	Capl Kalmoxicil 3 x sehari 1 tablet sesudah makan	33014738687206724866 83161564098937553640 33593616	66242669942590374504 42788265349864939843 34955488
17	Phenobarbital 3 x sehari 1 tablet sesudah makan	14091452706917278192 76366047408446123380 513296	73936615068157112458 70736251342389669880 65984134
18	Ephedrine 5mg 3 x sehari 1 tablet sesudah makan	74895672098643265563 92539114230178190906 73534615	12578295356854800188 37420448619903993019 876259174
19	Aminophylin 150mg 3 x sehari 1 tablet sesudah makan	12768208462823983915 52019211161487318755 408221615	73558867468544097161 44764998216409817971 13420979
20	Glyceril Guaicolate 3 x sehari 1 tablet sesudah makan	42074660866251251857 97302162045745913087 55249035	61257738943825576309 06500067853418355061 86413570
21	Codein HCL 1 x sehari 1 sendok sesudah makan	96741552108219237179 94499197963161781569 63881912	52536064746620918040 23117912339719401781 27448845
22	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	75993387306704928836 82226423391715323644 63993755	11964309577558535656 09944686475482327399 308462059

23	Luminal 1 x sehari 1 sendok sesudah makan	13608714269779704138 49745500552959671070 23546513	51584487211656571288 79276267450791666459 38776451
24	Erythromycin 250mg 3 x sehari 1 tablet sesudah makan	92256278166257325862 77358460772654483785 2990677	33466050653872559941 33329008455776018864 99405615
25	Vitamin B-compl 3 x sehari 1 tablet sesudah makan	10630457961214549633 93546185323395623829 333450518	48783375773484688750 98228738070717798094 01895814
26	Paramex 2 x sehari 1 tablet sesudah makan	40906928148071079272 90938736394871515404 02465953	10689087253480545079 88166505473891652078 487361314
27	lanzaprazole 30mg 3 x sehari sesudah makan	90651965870355086710 48128511198356567142 33369118	10499986752560664583 29500707443624699911 598113759
28	New diabetes 4 3 x sehari 1 tablet sesudah makan	96923387118750963150 68816968659518779452 75445746	72796685652331691773 67734777261497199809 88051480
29	Sanmag sirup 120ml 2 x sehari 1 sendok sesudah makan	10861934376035215454 35889756882681536664 100244249	18120526765096510074 35229999967599366696 61063862
30	Vometa sirup 60 ml 3 x sehari 1 sendok sesudah makan	96163273141566654783 04909688354978714920 2637773	94948269326597971644 57073191209188389366 54211053

A.2 Hasil Pengujian *Collision Attack*

No	Pesan	Pesan	Hashing
1	Vosea 2 x sehari 1 tablet sebelum makan	Alprazolam 3 x sehari 1 tablet sebelum makan	13188517948063830752 10380683479861367934 140406031
2	Asam Mefenamat 3 x sehari 1 tablet sesudah makan	Amfetamin 3 x sehari 1 tablet sebelum makan	79695848113955664436 76782834016820249644 8711420

3	Paracetamol 100 mg sacc lactisg 3 x sehari 1 tablet sesudah makan	Amitriptyline 2 x sehari 1 tablet sesudah makan	10351401688162359633 72235982818115660080 633156719
4	Sanmol 3 x sehari 1 tablet	Haloperidol 3 x sehari 1 tablet	10258810741008217866 41329368424644887874 712130777
5	SL ad 3 x sehari 1 tablet sesudah makan	Hydroquinone 3 x sehari 1 tablet sesudah makan	13504870106966690119 12780839223416827400 091208049
6	CTM 2 mg 3 x sehari 1 tablet sesudah makan	Memantine 3 x sehari 1 tablet sebelum makan	60426087533619103768 49526580616966440801 47937871
7	Lactas Calcium 300mg 3 x sehari 1 tablet sesudah makan	Metoprolol 3 x sehari 1 tablet sesudah makan	72625743156484075827 61334983060285180469 45352308
8	Konidin 3 x sehari 1 tablet sesudah makan	Mycophenolate Sodium 3 x sehari 1 tablet sesudah makan	39132910489236695942 61423938833510144281 34433405
9	Dulcolax 5 mg 3 x sehaei 1 tablet sesudah makan	Glibenclamide 3 x sehaei 1 tablet sesudah makan	43950957801287329830 79064170761279864209 81007541
10	Lodia 2mg 3 x sehari 1 tablet sebelum makan	Guaifenesin 2mg 3 x sehari 1 tablet sebelum makan	22087401926593872420 06467969986498616212 83119135
11	Democolin 3 x sehari 1 tablet sesudah makan	Neurobion 3 x sehari 1 tablet sebelum makan	13543811228676376275 49614111397702368156 380511151
12	Paratusin 3 x sehari 1 tablet sesudah makan	Nifedipine 3 x sehari 1 tablet sebelum makan	11179376868767982815 85862777064481285541 458958293
13	Becom-C 3 x sehari 1 tablet sesudah makan	Nystatin 2 x sehari 1 tablet sesudah makan	48820331355360230497 41934986920249046789 24223977
14	Ibuprofen 3 x sehari 1 tablet sesudah makan	Noscapine 3 x sehari 1 tablet	96349750016915794288 10690391045728899580 5653854
15	Dumin 3 x sehari 1 sendok sesudah makan	Nitrogen Oksida 3 x sehari 1 tablet sesudah makan	47690076803911749574 62835105154014523995 94428805

16	Capl Kalmoxicil 3 x sehari 1 tablet sesudah makan	Nevirapine 3 x sehari 1 tablet sebelum makan	13469248001939874652 51818356603668119911 884857284
17	Phenobarbital 3 x sehari 1 tablet sesudah makan	Nicotinamide 3 x sehari 1 tablet sesudah makan	12000352960161863247 54885701250556989065 082699640
18	Ephedrine 5mg 3 x sehari 1 tablet sesudah makan	Dekongestan 3 x sehari 1 tablet sesudah makan	10538879743483378130 73280903811023977030 370502469
19	Aminophyllin 150mg 3 x sehari 1 tablet sesudah makan	Dextrose 3 x sehari 1 tablet sesudah makan	91603918259121155639 16093915039036425510 22464430
20	Glyceril Guaicolate 3 x sehari 1 tablet sesudah makan	Diazepam 2mg 3 x sehari 1 tablet sebelum makan	13443768176773860187 18900529942240130770 880621878
21	Codein HCL 1 x sehari 1 sendok sesudah makan	Diltiazem 3 x sehari 1 tablet sebelum makan	33806480011738376626 73296427094115108697 26345302
22	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	Diuretik 3 x sehari 1 tablet sebelum makan	41223068564636080741 02091734234556608267 30197902
23	Luminal 1 x sehari 1 sendok sesudah makan	Dumolid 2 x sehari 1 tablet sesudah makan	93637025051597064633 18721846844399082524 17727865
24	Erythromycin 250mg 3 x sehari 1 tablet sesudah makan	Donepezil 3 x sehari 1 tablet	13646845359644776760 94121755030525599345 244934951
25	Vitamin B-compl 3 x sehari 1 tablet sesudah makan	Disulfiram 3 x sehari 1 tablet sesudah makan	80057475468514902199 74033656719037430654 01595893
26	Paramex 2 x sehari 1 tablet sesudah makan	Digoxin 3 x sehari 1 tablet sebelum makan	59903522869396650645 67794139375200510239 0884404
27	lanzoprazole 30mg 3 x sehari sesudah makan	Metoprolol 3 x sehari 1 tablet sesudah makan	96625243168022797649 43232841499108660549 05284938
28	New diabetes 4 3 x sehari 1 tablet sesudah makan	Quimidine 3 x sehari 1 tablet sesudah makan	12736971188117973483 34222413422979108396 793779968

29	Sanmag sirup 120ml 2 x sehari 1 sendok sesudah makan	Quinolone 3 x sehari 1 tablet sebelum makan	72537265180306492664 07490082511874629958 16407050
30	Vometa sirup 60 ml 3 x sehari 1 sendok sesudah makan	Ketorolac 3 x sehari 1 tablet sesudah makan	45483994161003832651 14082593626297494575 08666859

A.3 Hasil Pengujian *Birthday Attack*

No	Pesan	Hasil
1	Vosea 2 x sehari 1 tablet sebelum makan	11761228
2	Asam Mefenamat 3 x sehari 1 tablet sesudah makan	10934888
3	Paracetamol 100 mg sacc lactisg 3 x sehari 1 tablet sesudah makan	2092024
4	Sanmol 3 x sehari 1 tablet	5217629
5	SL ad 3 x sehari 1 tablet sesudah makan	15528290
6	CTM 2 mg 3 x sehari 1 tablet sesudah makan	14827724
7	Lactas Calcium 300mg 3 x sehari 1 tablet sesudah makan	1525216
8	Konidin 3 x sehari 1 tablet sesudah makan	15442752
9	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	13996717
10	Lodia 2mg 3 x sehari 1 tablet sebelum makan	2283562
11	Democolin 3 x sehari 1 tablet sesudah makan	709633
12	Paratusin 3 x sehari 1 tablet sesudah makan	7851568
13	Becom-C 3 x sehari 1 tablet sesudah makan	095404
14	Ibuprofen 3 x sehari 1 tablet sesudah makan	10128907
15	Dumin 3 x sehari 1 sendok sesudah makan	12410077
16	Capl Kalmoxicil 3 x sehari 1 tablet sesudah makan	881221113
17	Phenobarbital 3 x sehari 1 tablet sesudah makan	13514257
18	Ephedrine 5mg 3 x sehari 1 tablet sesudah makan	3071024
19	Aminophylin 150mg 3 x sehari 1 tablet sesudah makan	141419106
20	Glyceril Guaicolate 3 x sehari 1 tablet sesudah makan	1415141010
21	Codein HCL 1 x sehari 1 sendok sesudah makan	613127515
22	Ephidrib HCL 1 x sehari 1 sendok sesudah makan	135921013
23	Luminal 1 x sehari 1 sendok sesudah makan	121061412
24	Erythromycin 250mg 3 x sehari 1 tablet sesudah makan	43615514
25	Vitamin B-compl 3 x sehari 1 tablet sesudah makan	10107079
26	Paramex 2 x sehari 1 tablet sesudah makan	8cd125
27	lanzoprazole 30mg 3 x sehari sesudah makan	7111314
28	New diabetes 4 3 x sehari 1 tablet sesudah makan	7141250

29	Sanmag sirup 120ml 2 x sehari 1 sendok sesudah makan	115810511
30	Vometa sirup 60 ml 3 x sehari 1 sendok sesudah makan	111510338