

# Rancang Bangun Aplikasi Bank Sampah Berbasis Web Dengan Menggunakan Algoritma Kriptografi Advanced Encryption Standard (AES) Untuk Keamanan Data Transaksi Nasabah

**(STUDI KASUS: BANK SAMPAH MALAKA SARI – JAKARTA TIMUR)**



PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGRI SYARIF HIDAYATULLAH  
JAKARTA

## LEMBAR PERSETUJUAN

**Rancang Bangun Aplikasi Bank Sampah Berbasis Web Dengan Menggunakan  
Algoritma Kriptografi Advanced Encryption Standard (AES) Untuk Keamanan  
Data Transaksi Nasabah  
(Studi Kasus: Bank Sampah Malaka Sari – Jakarta Timur)**

Skripsi

Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer

Pada Fakultas Sains dan Teknologi

Universitas Islam Negeri Syarif Hidayatullah Jakarta

Oleh :

**Aditya Puji Nugroho**  
**1113091000103**

Menyetujui,

Pembimbing I



**Arini, S.T., M.T.**

**NIP. 1976013120090120001**

Pembimbing II



**Hendra Bayu Suseno, M.Kom**

**NIP. 1982121120091210003**

Mengetahui,

Ketua Program Studi Teknik Informatika



**Dr. Imam Marzuki Shofi, MT.**

**NIP.197202052008011010**

## LEMBAR PENGESAHAN


Skripsi yang berjudul "*Rancang Bangun Aplikasi Bank Sampah Berbasis Web Dengan Menggunakan Algoritma Kriptografi AES Untuk Keamanan Data Transaksi Nasabah*" telah diuji dan dinyatakan lulus dalam sidang Munaqosyah Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah Jakarta pada ..... Skripsi ini telah diterima sebagai salah satu syarat untuk memperoleh gelar Sarjana Strata Satu (S1) Program Studi Teknik Informatika.

Jakarta, ..... 2019

Tim Penguji,

Penguji I,

Penguji II,

  
Rizal Broer Bahawares, M.Kom

  
Nenny Anggraini, S.Kom, MT.

NIP. 197108062014111001

NIDN. 031 0097604

Tim Pembimbing,

Pembimbing I

Pembimbing II

  
Arini, S.T., M.T.

  
Hendra Bayu Suseno, M.Kom.


NIP. 1976013120090120001

NIP. 1982121120091210003

Mengetah  
ui,

Dekan

Ketua

Fakultas Sains dan Teknologi,  
  
Prof. Dr. Lily Surayya Eka  
Putri, M. Env. Stud

NIP. 19690404 200501 2 005

Program Studi Teknik Informatika,

  
Dr. Imam Marzuki Shofi, MT

NIP. 197202052008011010

## PERNYATAAN ORISINALITAS

Dengan ini saya menyatakan bahwa:

1. Skripsi ini merupakan hasil karya asli saya yang diajukan untuk memenuhi salah satu persyaratan memperoleh gelar Strata 1 di UIN Syarif Hidayatullah Jakarta.
2. Semua sumber yang saya gunakan dalam Penelitian ini telah saya cantumkan sesuai dengan ketentuan yang berlaku di UIN Syarif Hidayatullah Jakarta.
3. Apabila di kemudian hari terbukti karya ini bukan hasil karya asli saya atau merupakan hasil jiplakan karya orang lain, maka saya bersedia menerima sanksi yang berlaku di UIN Syarif Hidayatullah Jakarta.

Jakarta, November 2019



Aditya Puji Nugroho

NIM. 1113091000103

## ABSTRAK

**Aditya Puji Nugroho – 1113091000103.** Rancang Bangun Aplikasi Bank Sampah Berbasis Web Dengan Menggunakan Algoritma Kriptografi AES Untuk Keamanan Data Transaksi Nasabah (Studi Kasus: Bank Sampah Malaka Sari – Jakarta Timur). Dibimbing oleh **Arini, ST, MT dan Hendra Bayu Suseno, M.Kom**

Bank sampah malaka yang berlokasi di Jakarta timur berdiri sejak 2008, bank sampah malaka sari memiliki jumlah nasabah mencapai lebih dari 300 orang dan sampah yang terserap setiap bulan mencapai 2 – 2.5 ton. dalam menjalankan kegiatan nya dari hasil wawancara dan observasi masih terdapat beberapa masalah, antara lain: sulitnya mendata pemasukan sampah, kurangnya integritas antara buku tabungan nasabah dengan pembukuan bank sampah dan nasabah yang sering tidak membawa buku tabungan bahkan ada yang hilang, untuk mengantisipasi hal tersebut serta meningkatkan kinerja dalam pelayanan bank sampah maka diperlukan sebuah aplikasi untuk pengolahan data yang efektif dan efisien, yang didalamnya dapat memiliki system penyimpanan data yang aman dan melengkapi aplikasi tersebut dengan teknik pengamanan yang cukup baik. Dalam hal ini peneliti menggunakan algoritma kriptografi AES (*Advanced Encryption System*). Kriptografi AES hanya mengacu pada data saldo bank sampah, dan hasil enkripsi akan disimpan di dalam database mysql. Berdasarkan implementasi dan pengujian program, peneliti dapat menyimpulkan bahwa algoritma AES dinyatakan aman dalam mengamankan data transaksi nasabah karena sulit untuk ditembus oleh serangan brute force dan juga memerlukan waktu yang sangat lama untuk menemukan kunci yang benar serta memberikan alternatif bagi pihak Bank Sampah Malaka Sari untuk mengelola data bank sampah dengan baik karena adanya integritas data antara bank sampah dengan nasabah dan mampu menjaga dan melindungi kerahasiaan data dan informasi.

**Kata Kunci** : AES (Advanced Encryption System), MySQL, Kriptografi, Php, Bank Sampah



## KATA PENGANTAR

*Assalamualaikum Wr. Wb*

*Alhamdulillah*, segala puji hanya milik Allah SWT. *Shalawat* serta salam tercurah kepada Rasulullah SAW, semoga tersampaikan juga kepada keluarga, para sahabat, dan Kita semua sebagai umat Beliau, *Aamiin*. Berkat limpahan karunia dan rahmat Allah SWT, Penulis mampu menyelesaikan skripsi dengan judul “Rancang Bangun Aplikasi Bank Sampah Berbasis Web Dengan Menggunakan Algoritma Kriptografi AES Untuk Keamanan Data Transaksi Nasabah (Studi Kasus: Bank Sampah Malaka Sari – Jakarta Timur)” sebagai salah satu syarat untuk memperoleh gelar Sarjana Komputer (S.Kom) Strata Satu (S1) Program Studi Teknik Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Syarif Hidayatullah.

Dalam penyusunan skripsi ini tidak sedikit hambatan yang Penulis hadapi. Namun Penulis menyadari bahwa kelancaran dalam penyusunan skripsi ini berkat bantuan, dorongan, bimbingan, dan doa dari berbagai pihak sehingga kendala yang Penulis hadapi *alhamdulillah* dapat teratasi tentunya dengan izin Allah SWT. Oleh karena itu penulis mengucapkan terimakasih kepada:

1. Ibu Prof. Dr. Lily Surayya Eka Putri, M. Env. Stud, selaku Dekan Fakultas Sains dan Teknologi.
2. Bapak Dr. Imam M. Shofi, selaku ketua Program Studi Teknik Informatika, serta Bapak Andrew Fiade, M.Kom, selaku sekretaris Program Studi Teknik Informatika.
3. Ibu Arini ST, M.T, selaku Dosen Pembimbing I dan Bapak Hendra Bayu Suseno M.Kom, selaku Dosen Pembimbing II yang telah memberikan bimbingan, motivasi, dan arahan kepada Peneliti sehingga skripsi ini bisa selesai dengan baik.
4. Bapak Rizal Broer Bahawares, M.Kom. dan Ibu Nenny Anggraini, S.Kom, M.T. selaku dosen–dosen penguji yang telah memberikan

masukan dan saran-saran mulai dari rencana penelitian hingga selesainya skripsi ini.

5. Bapak Rizal Broer Bahawares, M.Kom sebagai dosen pembimbing yang selama ini mendampingi penulis selama menjalankan proses perkuliahan.
6. Bapak dan Ibu dosen di Fakultas Sains dan Teknologi Universitas Islam Negeri (UIN) Syarif Hidayatullah Jakarta, khususnya Program Studi Teknik Informatika atas bimbingan dan bantuannya hingga Peneliti dapat menyelesaikan studi dengan baik.
7. Bapak Prakoso, selaku Pengurus Bank Sampah Malaka Sari yang bersedia meluangkan waktunya untuk berdiskusi dan sebagai sumber kepakaran dalam Penelitian skripsi ini.
8. Ibunda tercinta Hamidah, Ayahanda tercinta Edy Pudjiono dan Kakak tercinta Suci Puji Ananda yang selalu memberikan doa dan dukungan baik materil maupun motivasi untuk menyelesaikan skripsi ini.
9. Teman-teman Program Studi Teknik Informatika program kerjasama FTUI pola 2 angkatan 2013 yang selalu mendukung dan memotivasi Peneliti.
10. Yogi Wiharso dan Hilal Helmi Balfas yang sudah meluangkan waktunya untuk memberikan dukungan, bantuan dan doanya dalam membantu penulisan skripsi ini agar dapat diselesaikan
11. Agustina Aling, yang telah membantu dan memberikan semangat setiap harinya dalam penyelesaian skripsi ini.

Akhir kata penulis berharap semoga hasil pemikiran yang tertuang dalam skripsi ini dapat bermanfaat. Amin.

*Wassalamualaikum Wr.Wb*

Jakarta, November 2019

Aditya Puji Nugroho

1113091000103

## DAFTAR ISI

LEMBAR PERSETUJUAN .....	II
LEMBAR PENGESAHAN.....	III
PERNYATAAN ORISINALITAS .....	IV
ABSTRAK.....	V
KATA PENGANTAR.....	VI
DAFTAR ISI.....	VIII
DAFTAR TABEL .....	X
DAFTAR GAMBAR.....	XII
BAB I .....	1
PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Batasan Masalah .....	4
1.4 Tujuan Penelitian .....	5
1.5 Manfaat Penelitian .....	5
1.6 Metode Penelitian.....	6
1.6.1 Metode Pengumpulan Data .....	6
1.6.2 Metode Pengembangan Sistem .....	7
1.7 Sistematika Penulisan .....	7
BAB II .....	9
LANDASAN TEORI .....	9
2.1 Rancang Bangun.....	9
2.2 Aplikasi.....	9
2.3 Database.....	9
2.4 Bank Sampah .....	10
2.5 Algoritma.....	11
2.6 Kriptografi.....	12
2.7 Advanced Encryption Standard (AES) / Rijindael .....	20
2.8 PHP (Hypertext Preprocessor).....	47
2.9 MySql .....	48
2.10 Rapid Application Diagram (RAD).....	49
2.11 Pengujian Blackbox .....	52
BAB III.....	55



<b>METODOLOGI PENELITIAN</b> .....	55
<b>3.1 Metode Pengumpulan Data</b> .....	55
<b>3.1.1 Studi Lapangan</b> .....	55
<b>3.1.2 Studi Pustaka</b> .....	56
<b>3.1.3 Studi Literature</b> .....	56
<b>3.2 Metode Pengembangan Sistem</b> .....	59
<b>3.3 Kerangka Berfikir Penelitian</b> .....	61
<b>BAB IV</b> .....	62
<b>ANALISA DAN PEMBAHASAN</b> .....	62
<b>4.1 Requirements Planning (Perencanaan Syarat-Syarat)</b> .....	62
<b>4.1.1 Analisis Masalah</b> .....	62
<b>4.1.2 Tinjauan Bank Sampah</b> .....	63
<b>4.1.3 Identifikasi Sistem Berjalan</b> .....	63
<b>4.1.4 Identifikasi Sistem Usulan</b> .....	65
<b>4.2 Workshop Design</b> .....	66
<b>4.3 Fase Implementasi</b> .....	120
<b>BAB V</b> .....	129
<b>HASIL DAN PEMBAHASAN</b> .....	129
<b>5.1 Hasil Output</b> .....	129
<b>BAB VI</b> .....	135
<b>KESIMPULAN DAN SARAN</b> .....	135
<b>6.1 Kesimpulan</b> .....	135
<b>6.2 Saran</b> .....	135
<b>DAFTAR PUSTAKA</b> .....	137
<b>LAMPIRAN</b> .....	139

## DAFTAR TABEL

Tabel 1. 1 Hasil Analisis Perbandingan Kecepatan Algoritma.....	2
Tabel 1. 1 Grafik Presentasi Perbandingan Kecepatan Enkripsi dan Dekripsi .....	3
Tabel 2. 1 Tiga Buah Versi AES.....	22
Tabel 2. 2 Contoh Sebuah State .....	24
Tabel 2. 3 Matriks Affine.....	27
Tabel 2.4 Table S-Box Didalam AES .....	28
Tabel 2.5 Transformasi SubBytes .....	29
Tabel 2.6 Transformasi ShiftRows .....	31
Tabel 2.7 Transformasi ShiftRows terhadap state hasil.....	32
Tabel 2.8 Transformasi MixColumns .....	33
Tabel 2.9 Table E untuk membantu perhitungan MixColumns.....	36
Tabel 2.10 Table L untuk membantu perhitungan MixColumns.....	37
Tabel 2.11 Transformasi AddRoundKey .....	41
Tabel 2.12 InvShiftRows didalam AES .....	45
Tabel 2.13 InvShiftRows di dalam AES .....	46
Tabel 2. 14 Perbandingan Blackbox Testing dan WhiteboxTesting.....	53
Tabel 3. 1 Perbandingan Studi Literatur Sejenis.....	56
Tabel 4.1 Identifikasi Aktor .....	73
Tabel 4.2 Identifikasi Use Case .....	73
Tabel 4.3 Deskripsi Use Case Login.....	76
Tabel 4.4 Deskripsi Use Case Input Transaksi .....	77
Tabel 4.5 Deskripsi Use Case Lihat Data Transaksi.....	78
Tabel 4.6 Deskripsi Use Case Input Debit.....	79
Tabel 4.7 Deskripsi Use Case Tambah Data.....	80
Tabel 4.8 Deskripsi Use Case Lihat Data Transaksi.....	81
Tabel 4.9 Deskripsi Use Case Edit Harga Jual .....	83
Tabel 4.10 Deskripsi Use Case Lihat Riwayat Transaksi .....	84
Tabel 4.11 Deskripsi Use Case Lihat Data Pemasukan Sampah .....	85
Tabel 4.12 Deskripsi Use Case Log Out.....	85
Tabel 4.13 Table Pengurus.....	106
Tabel 4.14 Table Log .....	107
Tabel 4.15 Table Komoditas .....	108
Tabel 4.16 Table Harga Jual Sampah .....	108
Tabel 4.17 Table Id Akses .....	109
Tabel 4.18 Table Transaksi Nasabah .....	109
Tabel 4.19 Table Data Pemasukan Sampah.....	110
Tabel 4.20 Table Saldo Nasabah.....	113
Tabel 4.21 Table Nasabah.....	114
Tabel 4.22 Table Saldo Bank Sampah .....	114

Tabel 4.23 Table Bank Sampah .....	115
Tabel 4.24 Tabel Pengujian Level Admin .....	124
Tabel 4.24 Tabel Pengujian Level Nasabah.....	129



## DAFTAR GAMBAR

Gambar 2.1 Skema Enkripsi Dan Dekripsi Dengan Menggunakan Kunci .....	13
Gambar 2.2 Sistem Kriptografi Simetrik .....	15
Gambar 2.3 Sistem Kriptografi Asimetrik .....	16
Gambar 2.4 Skema Algoritma Simetris .....	17
Gambar 2.5 Skema Algoritma Asimetris .....	19
Gambar 2.6 Matriks state berukuran 4 x 4 untuk blok pesan 128-bit .....	24
Gambar 2.7 Diagram Proses Enkripsi di dalam AES .....	26
Gambar 2.8 Tahap Pengembangan RAD .....	50
Gambar 3.1 Kerangka Berfikir Penelitian .....	61
Gambar 4.1 Proses Sistem Berjalan Alur Pendaftaran .....	64
Gambar 4.2 Proses Sistem Berjalan Bank Sampah .....	64
Gambar 4.3 Proses Sistem Usulan Bank Sampah .....	65
Gambar 4.4 Use Case Diagram .....	75
Gambar 4.5 Activity Diagram Login .....	87
Gambar 4.6 Activity Diagram Input Transaksi .....	88
Gambar 4.7 Activity Diagram Data Transaksi .....	89
Gambar 4.8 Activity Diagram Input Debit .....	90
Gambar 4.9 Activity Diagram Tambah Data .....	91
Gambar 4.10 Activity Diagram Transaksi Saldo .....	92
Gambar 4.11 Activity Diagram Edit Harga Jual .....	93
Gambar 4.12 Activity Diagram Transaksi Saldo .....	94
Gambar 4.14 Activity Diagram Data Pemasukan Sampah .....	96
Gambar 4.15 Activity Diagram Log Out .....	97
Gambar 4.16 Sequence Diagram Log in .....	98
Gambar 4.17 Sequence Diagram Input Transaksi .....	99
Gambar 4.18 Sequence Diagram Data Transaksi .....	100
Gambar 4.19 Sequence Diagram Input Debit .....	100
Gambar 4.20 Sequence Diagram Tambah Data .....	101
Gambar 4.21 Sequence Diagram Transaksi Saldo .....	102
Gambar 4.22 Sequence Diagram Edit Harga Jual .....	103
Gambar 4.23 Sequence Diagram Riwayat Transaksi .....	104
Gambar 4.24 Sequence Diagram Data Pemasukan Sampah .....	104
Gambar 4.25 Class Diagram .....	105
Gambar 4.26 Database Schema .....	106
Gambar 4.27 Halaman Login .....	116
Gambar 4.28 Halaman Admin .....	117
Gambar 4.29 Halaman Saldo Bank Sampah .....	117

Gambar 4.30 Halaman Harga Jual Sampah .....	118
Gambar 4.31 Halaman Riwayat Transaksi.....	118
Gambar 4.32 Halaman Transaksi Nasabah .....	119
Gambar 4.33 Data Transaksi Nasabah .....	119
Gambar 4.34 Halaman Debit Nasabah.....	120
Gambar 4.35 Halaman Tambah Data Nasabah .....	120
Gambar 4.36 Halaman Nasabah.....	121
Gambar 4.37 Halaman Info Saldo Nasabah .....	121
Gambar 4.38 Halaman Data Transaksi Nasabah.....	122
Gambar 4.39 Cmd Brute Force .....	123
Gambar 4.40 Pengujian Sistem Aplikasi .....	123
Gambar 4.41 Serangan Brute Force .....	124
Gambar 5.1 Halaman Login .....	132
Gambar 5.2 Halaman Utama Admin.....	132
Gambar 5.3 Halaman Saldo Bank Sampah .....	133
Gambar 5.4 Halaman Edit Harga Jual.....	133
Gambar 5.5 Riwayat Transaksi Nasabah .....	134
Gambar 5.6 Halaman Transaksi Nasabah .....	134
Gambar 5.7 Halaman Data Transaksi Nasabah.....	135
Gambar 5.8 Halaman Debit Nasabah.....	135
Gambar 5.9 Halaman Tambah Nasabah.....	136
Gambar 5.10 Halaman Utama Nasabah .....	136
Gambar 5.11 Halaman Profile.....	137



## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Salah satu permasalahan besar yang dialami kota-kota besar di Indonesia adalah persampahan. (Anih Sri Suryani, 2014) Sampah dapat diartikan sebagai konsekuensi adanya aktivitas kehidupan manusia. Tidak dapat dipungkiri, sampah akan selalu ada selama aktivitas kehidupan masih terus berjalan. Setiap tahunnya, dapat dipastikan volume sampah akan selalu bertambah seiring dengan pola konsumerisme masyarakat yang semakin meningkat. Kementerian Lingkungan Hidup mencatat rata-rata penduduk Indonesia menghasilkan sekitar 2,5 liter sampah per hari atau 625 juta liter dari jumlah total penduduk. Kondisi ini akan terus bertambah sesuai dengan kondisi lingkungannya.

Sebagai salah satu solusi untuk mengatasi masalah tersebut, Kementerian Lingkungan Hidup melakukan upaya pengembangan Bank Sampah. Kegiatan ini mengajarkan masyarakat untuk memilah sampah, sekaligus menumbuhkan kesadaran masyarakat dalam pengolahan sampah secara bijak. Harapannya akan dapat mengurangi jumlah sampah yang diangkut ke TPA. Pembangunan bank sampah ini merupakan momentum awal dalam membina kesadaran kolektif masyarakat untuk mulai memilah, mendaur-ulang, dan memanfaatkan sampah. Hal ini penting, karena sampah mempunyai nilai jual dan pengelolaan sampah yang berwawasan lingkungan dapat menjadi budaya baru Indonesia.

Peran Bank Sampah menjadi penting dengan terbitnya Peraturan Pemerintah (PP) Nomor 81 Tahun 2012 tentang Pengelolaan Sampah Rumah Tangga dan Sampah Sejenis Sampah Rumah Tangga. PP tersebut mengatur tentang kewajiban produsen untuk melakukan kegiatan 3R dengan cara menghasilkan produk yang menggunakan kemasan yang mudah diurai oleh proses alam, yang menimbulkan sampah sesedikit mungkin, menggunakan bahan baku produksi yang dapat didaur ulang dan diguna ulang; dan/atau menarik kembali sampah dari produk dan kemasan produk untuk didaur ulang dan diguna ulang. Dengan adanya Bank Sampah, maka produsen dapat melakukan kerjasama dengan Bank Sampah yang

ada agar dapat mengolah sampah dari produk yang dihasilkannya sesuai dengan amanat PP tersebut.

Salah satu contoh bank sampah yang masih berjalan dari 2008 hingga sekarang ialah Bank Sampah Malaka Sari RW 03 Jakarta Timur. Sebagai salah satu bank sampah yang mendapatkan penghargaan *Gold* dari pemprov DKI Jakarta melalui program “Jakarta *Green and Clean*” dengan jumlah nasabah mencapai lebih dari 300 orang dan sampah yang terserap setiap bulan mencapai 2 – 2.5 Ton.

Akan tetapi dalam proses pengimplementasian nya, dari hasil wawancara dan obeservasi masih terdapat beberapa masalah, antara lain: nasabah sering tidak membawa buku tabungan, bahkan ada yang hilang, sulitnya men data pemasukan sampah, kurangnya integritas data antara buku tabungan nasabah dengan pembukuan bank sampah, maka diperlukan sebuah aplikasi untuk pengolahan data yang efektif dan efisien, yang didalamnya dapat memiliki sistem penyimpanan data yang aman dan tentunya melengkapi aplikasi dengan teknik pengaman yang cukup baik untuk mengantisipasi adanya kemungkinan buruk yang akan terjadi seperti manipulasi data. Dalam penelitian ini peneliti akan mengimplementasikan sistem pengamanan dengan menggunakan teknik kriptografi.

Berdasarkan jurnal yang peneliti dapatkan, terdapat 4 metode algoritma kriptografi. DES (*Data Encryption Standard*), IDEA (*International Data Encryption Algorithm*), Blowfish (*OpenPGP.Cipher.4*) dan AES (*Advanced Encryption Standard*). Keempat mode tersebut mempunyai kelebihan dan kekurangan.

**Table 1.1** Hasil Analisis Perbandingan Kecepatan Algoritma

Algoritma	Enkripsi (kbyte/detik)	Dekripsi (kbyte/detik)
DES	402	608

AES	1.508	1.433
IDEA	173	57
Blowfish	1.063	1.075

Berdasarkan hasil analisis (Donzilo Antonio Meko, 2018) Perbandingan Kecepatan dekripsi algoritma keempat algoritma di atas dapat dilihat bahwa persentasi tertinggi dari algoritma AES 45%, Blowfish 34%, DES 19% dan terakhir adalah algoritma IDEA yang hanya memiliki kecepatan 2%. Sehingga dapat disimpulkan bahwa algoritma AES jauh lebih cepat dari algoritma dari ketiga algoritma lainnya dalam hal proses enkripsi dan dekripsi data diikuti oleh Blowfish, DES, IDEA dan yang terakhir adalah AES.

**Table.1.2** Grafik Presentasi Perbandingan Kecepatan Enkripsi dan Dekripsi



Berdasarkan penelitian yang telah dilakukan maka dapat disimpulkan bahwa :

- Kecepatan enkripsi dan dekripsi data dengan menggunakan algoritma AES lebih baik dibanding algoritma Blowfish dimana untuk persentasi kecepatan algoritma AES adalah 48% untuk proses enkripsi data dan 45% untuk dekripsi data. Sedangkan algoritma Blowfish memiliki kecepatan enkripsi dan dekripsi data sama yaitu 34%

- Kecepatan enkripsi dan dekripsi data dengan menggunakan algoritma AES lebih baik dibanding algoritma IDEA dimana untuk persentasi kecepatan algoritma AES adalah 48% untuk proses enkripsi data dan 45% untuk dekripsi data. Sedangkan algoritma IDEA memiliki kecepatan enkripsi dan dekripsi data sama yaitu 34%
- Kecepatan enkripsi dan dekripsi data dengan menggunakan algoritma AES lebih baik dibanding algoritma ketiga algoritma lainnya yaitu DES, Blowfish dan IDEA dimana kecepatan tertinggi untuk proses enkripsinya adalah 48% dan kecepatan dekripsinya adalah 45% dan kecepatan terendah dimiliki oleh algoritma IDEA yaitu kecepatan enkripsi sebesar 5% dan dekripsi sebesar 2%

Merujuk kelebihan yang dimiliki oleh Algoritma AES (*Advanced Encryption Standard*) pada penjelasan yang sudah disebutkan diatas maka penulis mengangkat topik dengan judul “*Rancang Bangun Aplikasi Bank Sampah Dengan Menggunakan Algoritma AES Untuk Keamanan Data Transaksi Nasabah (Studi Kasus: Bank Sampah Malaka Sari – Jakarta Timur)*”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang diatas, Maka rumusan masalah yang didapat yaitu bagaimana merancang bangun Aplikasi Bank Sampah dengan menggunakan Algoritma *Advanced Encryption Standard (AES)* untuk Keamanan Data Transaksi Nasabah?

## 1.3 Batasan Masalah

Dengan terbatasnya kemampuan, maka penulis menyadari perlu adanya pembatasan masalah yaitu:

1. Bahasa pemrograman dan sistem *Database* yang digunakan adalah PHP 7.1.30 sebagai bahasa pemrograman, MySQL 10.3.16 sebagai *Database server*, XAMPP 3.2.4 sebagai *webserver*.
2. Metode pengumpulan data dalam penelitian ini dengan melakukan studi lapangan, studi pustaka dan studi literature.

3. Metode pengembangan sistem yang digunakan berorientasi objek dengan model *Rapid Application Development* (RAD) yang terdiri atas tahap *Requirement Planning*, *RAD Design Workshop* dan *Implementation*.
4. Pada penelitian ini implementasi algoritma *advanced encryption system* (AES) untuk proses kriptografi (enkripsi dan dekripsi) hanya pada transaksi antar nasabah.

#### 1.4 Tujuan Penelitian

Tujuan dari penelitian dalam penulisan skripsi ini adalah merancang bangun aplikasi Bank Sampah berbasis Web dengan menggunakan Algoritma Kriptografi AES untuk Keamanan Data Transaksi Nasabah guna meningkatkan pelayanan Bank Sampah Malaka Sari.

#### 1.5 Manfaat Penelitian

Sesuai dengan permasalahan dan tujuan penelitian, maka manfaat penelitian dapat dirumuskan sebagai berikut:.

##### 1. Bagi Mahasiswa

1. Mengembangkan dan menerapkan ilmu – ilmu yang diperoleh selama perkuliahan
2. Menambahkan pengetahuan dan cara kerja bank sampah secara real time
3. Memotivasi mahasiswa untuk ikut berperan dalam kegiatan bank sampah

##### 2. Bagi Tempat Penelitian

1. Proses akumulasi keuangan yang terprogram dan valid.
2. Monitoring data yang terstruktur dengan baik diharapkan mampu memaksimalkan kegiatan bank sampah.
3. Keamanan data yang tinggi guna meningkatkan nasabah dalam berperan pada kegiatan bank sampah.
4. Membangun relasi antara bank sampah dengan dunia pendidikan



5. Pengurus dapat melakukan transaksi nasabah secara efektif dan efisien karena tidak perlu mencatat di buku besar sebagai integritas data dan dapat memonitoring pemasukan sampah serta mengecek pengeluaran dan pemasukan saldo.
  6. Mempermudah nasabah untuk mengecek data transaksi.
3. Bagi Universitas
1. Mengetahui kemampuan mahasiswa dalam menguasai materi ilmu yang telah diperoleh selama kuliah
  2. Sebagai referensi dan bahan evaluasi pada penelitian berikutnya

## 1.6 Metode Penelitian

Pada penelitian ini, peneliti menggunakan metode pengumpulan data dan metode pengembangan sistem :

### 1.6.1 Metode Pengumpulan Data

#### 1. Studi Pustaka

Mengumpulkan data dengan cara membaca, serta mempelajari buku-buku referensi, jurnal, dan lain-lain yang dapat dijadikan acuan pembahasan penelitian sebagai data informasi utama yang digunakan dalam masalah ini.

#### 2. Studi Lapangan

##### a) Observasi

Mengumpulkan data dengan cara mengamati secara langsung terhadap proses kegiatan yang terjadi di lapangan.

##### b) Wawancara

Mengumpulkan data dengan cara wawancara secara langsung kepada pihak-pihak yang berkaitan dengan penelitian ini

##### c) Studi Literatur

Adalah metode yang digunakan penulis dalam mencari perbandingan dari penelitian yang sudah ada dan membahas tentang masalah yang sejenis

### 1.6.2 Metode Pengembangan Sistem

Dalam rancang bangun aplikasi bank sampah pusat ini penulis menggunakan metode (RAD) Rapid Application Development, meliputi:

1. Fase perencanaan syarat-syarat, melakukan identifikasi terhadap kebutuhan informasi untuk memecahkan permasalahan dan menganalisa metode yang tepat guna memberikan solusi untuk mencapai tujuan dan syarat-syarat informasi
2. Fase perancangan (design workshop)
  - a. Fase Perancangan Aplikasi : merancang proses-proses yang terjadi pada sistem program, *database*, maupun *interface* aplikasi yang hendak dibangun,
  - b. Fase Konstruksi  
Membangun aplikasi yang dibuat dengan cara pengkodean program, Melakukan validasi pada sistem dalam pembuatan aplikasi/konstruksi bank sampah pusat dan konstruksi *interface*
3. Fase implementasi  
fase di mana aplikasi sistem yang dibangun pada fase sebelumnya diimplementasikan. Dalam fase ini dilakukan implementasi dalam *secara real time* karena data akan di olah secara *online* dan melakukan pengujian terhadap aplikasi

### 1.7 Sistematika Penulisan

Penyusunan laporan skripsi ini penulis bagi dalam beberapa bab yang secara singkat dapat dijelaskan sebagai berikut :

## BAB I PENDAHULUAN

Bab ini berisikan latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metode penelitian dan sistematika penulisan.

## **BAB II LANDASAN TEORI**

Bab ini menguraikan teori yang terkait dengan permasalahan yang diambil penulis. Teori-teori diambil dari beberapa literatur sejenis, jurnal, dokumentasi dan informasi dari berbagai sumber.

## **BAB III METODOLOGI PENELITIAN**

Bab ini menguraikan teori yang terkait dengan permasalahan yang diambil penulis. Teori-teori diambil dari beberapa literatur sejenis, jurnal, dokumentasi dan informasi dari berbagai sumber.

## **BAB IV IMPLEMENTASI**

Bab ini berisi tentang hasil penelitian berdasarkan metode yang sudah dipilih dan melakukan pembahasan berdasarkan hasil penelitian dan perumusan masalah.

## **BAB V HASIL DAN PEMBAHASAN**

Bab ini berisi tentang hasil penelitian berdasarkan metode yang sudah dipilih dan melakukan pembahasan berdasarkan hasil penelitian dan perumusan masalah

## **BAB VI PENUTUPAN**

Bab ini berisi kesimpulan yang di dapat dari hasil penelitian yang dilakukan dan saran-saran yang dapat digunakan sebagai bahan masukan untuk pengembangan lebih lanjut

## **BAB II**

### **LANDASAN TEORI**

#### **2.1 Rancang Bangun**

Rancang bangun adalah proses pembangunan sistem untuk menciptakan sistem baru maupun mengganti atau memperbaiki sistem yang telah ada baik secara keseluruhan maupun hanya sebagian. (Yuntari Purba Sari, 2017)

#### **2.2 Aplikasi**

Aplikasi merupakan suatu subteks perangkat lunak komputer yang memanfaatkan kemampuan komputer langsung untuk melakukan satu tugas yang diinginkan pengguna. Terdapat beberapa teori yang mendefinisikan aplikasi seperti yang di kemukakan oleh beberapa ahli, di antaranya adalah (Theresa Ayu, 2016):

- a) Aplikasi adalah satu unit perangkat lunak yang dibuat untuk melayani kebutuhan akan beberapa aktivitas seperti system perniagaan, game, pelayanan masyarakat, periklanan, atau semua proses yang hamper dilakukan manusia.
- b) Aplikasi merupakan program yang dikembangkan untuk memenuhi kebutuhan pengguna dalam menjalankan pekerjaan tertentu. Jadi aplikasi merupakan sebuah program yang dibuat dalam sebuah perangkat lunak dengan computer untuk memudahkan pekerjaan atau tugas-tugas seperti penerapan, penggunaan dan penambahan data yang dibutuhkan.

#### **2.3 Database**

Database adalah kumpulan data terstruktur. Agar dapat menambahkan, mengakses, dan memproses data yang tersimpan dalam database komputer, dibutuhkan sistem manajemen basis data (database management system). Dalam pengembangan perangkat lunak tradisional yang memanfaatkan pemrosesan file, setiap kelompok pengguna menyimpan file-file-nya sendiri untuk menangani aplikasi pengolahan

datanya masing-masing. Hal ini mengakibatkan adanya kerangkapan data atau disebut dengan redundancy.

Redundansi dalam proses penyimpanan data yang terjadi berkali-kali dapat mengakibatkan beberapa masalah. Pertama, ada kebutuhan untuk melakukan pembaruan logis tunggal, misalnya seperti memasukkan data pada siswa baru beberapa kali: satu kali untuk setiap file tempat data siswa direkam. Hal ini menyebabkan duplikasi data. Kedua, ruang penyimpanan terbuang ketika data yang sama disimpan berulang kali, dan masalah ini mungkin serius untuk database yang besar. Ketiga, file yang mewakili data yang sama mungkin menjadi tidak konsisten. Hal ini bisa terjadi karena update diaplikasikan pada beberapa file tapi tidak untuk file yang lain. (Cosmas Eko Suharyanto, Dkk, 2017).

## 2.4 Bank Sampah

Bank Sampah adalah suatu sistem pengelolaan sampah kering secara kolektif yang mendorong masyarakat untuk berperan serta aktif di dalamnya. Sistem ini akan menampung, memilah, dan menyalurkan sampah bernilai ekonomi pada pasar sehingga masyarakat mendapat keuntungan ekonomi dari menabung sampah.

Semua kegiatan dalam sistem bank sampah dilakukan dari, oleh dan untuk masyarakat. Seperti halnya bank konvensional, bank sampah juga memiliki sistem manajerial yang operasionalnya dilakukan oleh masyarakat. Bank sampah bisa juga memberikan manfaat ekonomi untuk masyarakat.

Sampah yang disetorkan oleh nasabah sudah harus dipilah. Persyaratan ini mendorong masyarakat untuk memisahkan dan mengelompokkan sampah. Misalnya, berdasarkan jenis material: plastik, kertas, kaca dan metal. Jadi, bank sampah akan menciptakan budaya baru agar masyarakat mau memilah sampah. (Yayasan Unilever Indonesia, 2013).



## 2.5 Algoritma

Algoritma adalah sistim kerja komputer memiliki brainware, hardware, dan software. Tanpa salah satu dari ketiga sistim tersebut, komputer tidak akan berguna. Kita akan lebih fokus pada software komputer. Software terbangun atas susunan program) dan syntax (cara penulisan/pembuatan program). Untuk menyusun program atau syntax, diperlukannya langkahlangkah yang sistematis dan logis untuk dapat menyelesaikan masalah atau tujuan dalam proses pembuatan suatu software. Maka, algoritma berperan penting dalam penyusunan program atau syntax tersebut

Pengertian algoritma adalah susunan yang logis dan sistematis untuk memecahkan suatu masalah atau untuk mencapai tujuan tertentu. Dalam dunia komputer, algoritma sangat berperan penting dalam pembangunan suatu software(Gun Gun Maulana, 2017).

### 2.5.1 Struktur Dasar Algoritma

#### a. Sekuensial (runtunan)

Pada struktur sekuensial ini langkah-langkah yang dilakukan dalam algoritma diproses secara berurutan. Dimulai dari langkah pertama, kedua, dan seterusnya. Pada dasarnya suatu program memang menjalankan suatu proses dari yang dasar seperti struktur ini.

#### b. Struktur seleksi

Struktur seleksi menyatakan pemilihan langkah yang didasarkan oleh suatu kondisi atau pengambilan suatu keputusan. Struktur ini ditandai selalu dengan bentuk flowchart decision (flowchart yang berbentuk belah ketupat). Banyak contoh yang dapat diterapkan pada struktur jenis ini jika itu menyangkut keputusan, diantaranya: diskon yang berbeda berdasarkan jumlah barang yang ingin dibeli.

c. Struktur perulangan

Struktur ini memberikan suatu perintah atau tindakan yang dilakukan beberapa kali. Misalnya jika teman mau menuliskan kata “belajar c” sebanyak sepuluh kali. Akan lebih efisien jika teman menggunakan struktur ini dari pada sekedar menuliskannya berturut-turut sebanyak sepuluh kali

## 2.6 Kriptografi

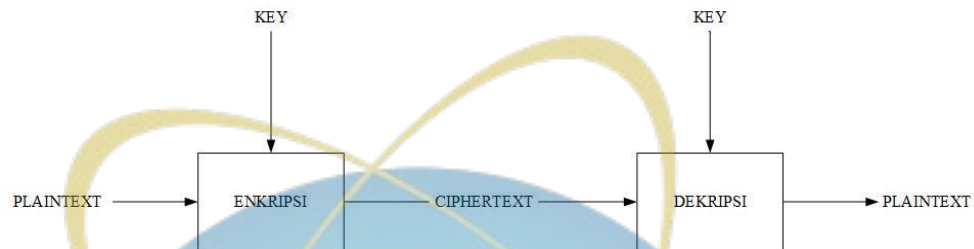
### 2.6.1 Definisi Kriptografi

Kriptografi (cryptography) merupakan ilmu dan seni untuk menjaga pesan agar aman. (Cryptography is the art and science of keeping messages secure) “Crypto” berarti “secret” (rahasia) dan “graphy” berarti “writing” (tulisan). Jadi, kriptologi adalah ilmu dan seni untuk menjaga keamanan pesan yang akan dikirim ke penerima sehingga data atau pesan tersebut aman dan tidak diketahui oleh pihak ketiga. Data atau pesan yang akan di kirim di ubah menjadi kode-kode yang tidak dipahami oleh pihak ketiga.

Kriptografi membuat data atau pesan menjadi kode-kode terlebih dahulu oleh pengirim. Proses ini dikenal dengan enkripsi. Enkripsi diartikan sebagai proses diubahnya data atau pesan yang hendak dikirim menjadi bentuk yang hampir tidak dikenali oleh pihak ketiga. Setelah data atau pesan itu sampai kepada penerima, maka penerima melakukan dekripsi yang merupakan kebalikan dari enkripsi. Dekripsi diartikan sebagai proses mengubah data atau pesan kembali ke bentuk semula sehingga data atau pesan dapat tersampaikan dan dimengerti oleh penerima. Data atau pesan asli dinamakan plaintext sedangkan sesudah dikodekan dinamakan ciphertext. Proses enkripsi dan dekripsi memerlukan kunci dalam mekanismenya dan biasanya berupa string atau deretan bilangan. Berikut ini contoh proses enkripsi dan dekripsi yang digunakan dalam pengiriman pesan. (Sumandri, 2017)

### 2.9.2 Tujuan Kriptografi

Ada empat tujuan mendasar dari ilmu kriptografi ini yang juga merupakan aspek keamanan informasi yaitu (Ary Hidayatullah dan Entik Insanudin, 2016) :



**Gambar 2.1** Skema Enkripsi Dan Dekripsi Dengan Menggunakan Kunci

- a. Kerahasiaan, adalah layanan yang digunakan untuk menjaga isi dari informasi dari siapapun kecuali yang memiliki otoritas atau kunci rahasia untuk membuka atau mengupas informasi yang telah disandi.
- b. Integritas data, adalah berhubungan dengan penjagaan dari perubahan data secara tidak sah. Untuk menjaga integritas data, sistem harus memiliki kemampuan untuk mendeteksi manipulasi data oleh pihak-pihak yang tidak berhak, antara lain penyisipan, penghapusan, dan pensubsitusian data lain kedalam data yang sebenarnya.
- c. Autentikasi, adalah berhubungan dengan identifikasi atau pengenalan, baik secara kesatuan sistem maupun informasi itu sendiri. Dua pihak yang saling berkomunikasi harus saling memperkenalkan diri. Informasi yang dikirimkan melalui kanal harus diautentikasi keaslian, isi datanya, waktu pengiriman, dan lain-lain
- d. Non-repudiasi, atau penyangkalan adalah usaha untuk mencegah terjadinya penyangkalan terhadap pengiriman

atau terciptanya suatu informasi oleh yang mengirimkan atau membuat.

### 2.6.3 Sistem Kriptografi Klasik

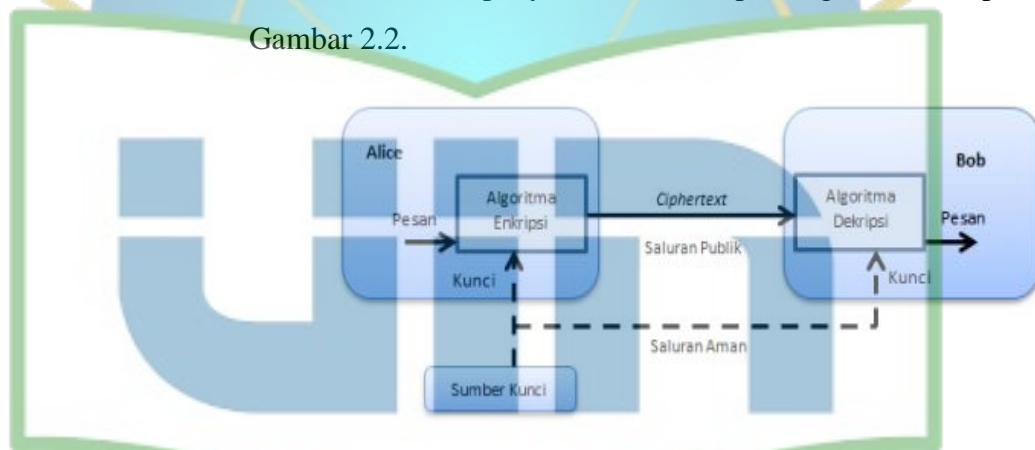
Sistem kriptografi klasik umumnya telah digunakan jauh sebelum era komputer. Kriptografi klasik juga dibagi menjadi dua jenis *cipher* yaitu *cipher* transposisi yang mengubah susunan huruf - huruf di dalam pesan dan *cipher* substitusi yang mengganti setiap huruf atau kelompok huruf dengan sebuah huruf atau kelompok huruf lain. Kriptografi klasik, teknik enkripsi yang digunakan adalah enkripsi simetris dimana kunci dekripsi sama dengan kunci enkripsi. Penyandian ini berorientasi pada karakter. Terdapat 5 bagian dalam sistem kriptografi klasik (Theresa Ayu, 2016) yaitu:

1. *Plaintext* Pesan atau data dalam bentuk aslinya yang dapat dibaca dan masukan bagi algoritma enkripsi.
2. *Secret Key* Masukan bagi algoritma enkripsi merupakan nilai yang bebas terhadap teks asli dan menentukan hasil keluaran algoritma enkripsi.
3. *Ciphertext* Hasil dari proses algoritma enkripsi dan teks asli dianggap telah tersembunyi.
4. Algoritma Enkripsi, Algoritma enkripsi memiliki 2 masukan yaitu teks asli dan kunci rahasia, kedua masukan tersebut akan diproses sehingga menghasilkan teks sandi.
5. Algoritma Dekripsi, Algoritma dekripsi memiliki 2 masukan yaitu teks sandi dan kunci rahasia, keduanya akan diproses sehingga menghasilkan teks asli.

#### 2.6.4 Sistem Kriptografi Modern

Sistem kriptografi modern umumnya berorientasi pada bit. Untuk *public key cryptography*, diperlukan teknik enkripsi asimetris dimana kunci dekripsi tidak sama dengan kunci enkripsi. Enkripsi, dekripsi dan pembuatan kunci untuk teknik enkripsi asimetris memerlukan komputasi yang lebih intensif dibandingkan enkripsi simetris, karena enkripsi asimetris menggunakan bilangan - bilangan yang sangat besar. Beberapa mekanisme yang berkembang pada kriptografi modern (Theresa Ayu, 2016):

1. Penyandian dengan kunci simetrik (*symmetric key encipherment*). Penyandian dengan kunci simetrik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai sama. Penyandian ini masih digunakan pada kriptografi modern. Skema penyandian ini dapat digambarkan pada Gambar 2.2.

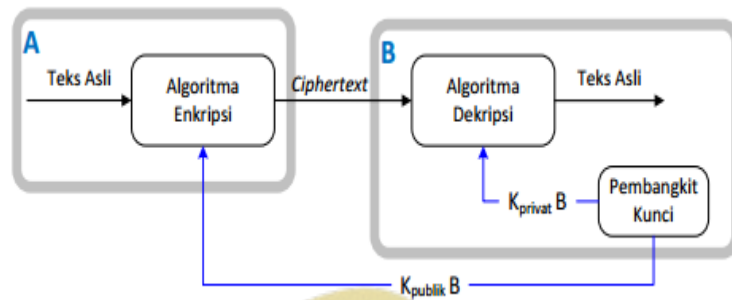


Gambar 2.2 Sistem Kriptografi Simetrik

(Sumber : Theresa Ayu, 2016)

2. Penyandian dengan kunci asimetrik (*asymmetric key encipherment*) Penyandian dengan kunci asimetrik yang disebut juga dengan kunci publik adalah penyandian yang kunci enkripsi dan kunci dekripsi bernilai berbeda. Penyandian ini yang banyak dikembangkan. Skema penyandian ini dapat digambarkan pada Gambar 2.3.





**Gambar 2.3** Sistem Kriptografi Asimetrik

(Sumber: Theresa Ayu, 2016)

Tidak seperti sistem kriptografi klasik di mana setiap entitas harus saling mengetahui kunci rahasia, sistem kriptografi modern yang juga disebut kriptografi kunci asimetrik, memiliki dua jenis kunci, yaitu kunci enkripsi dan kunci dekripsi yang berbeda. Dalam kriptografi kunci asimetris, hampir semua algoritma kriptografinya menggunakan konsep kunci publik, kecuali algoritma Pohlig - Hellman karena kunci enkripsi maupun kunci dekripsinya bersifat privat.

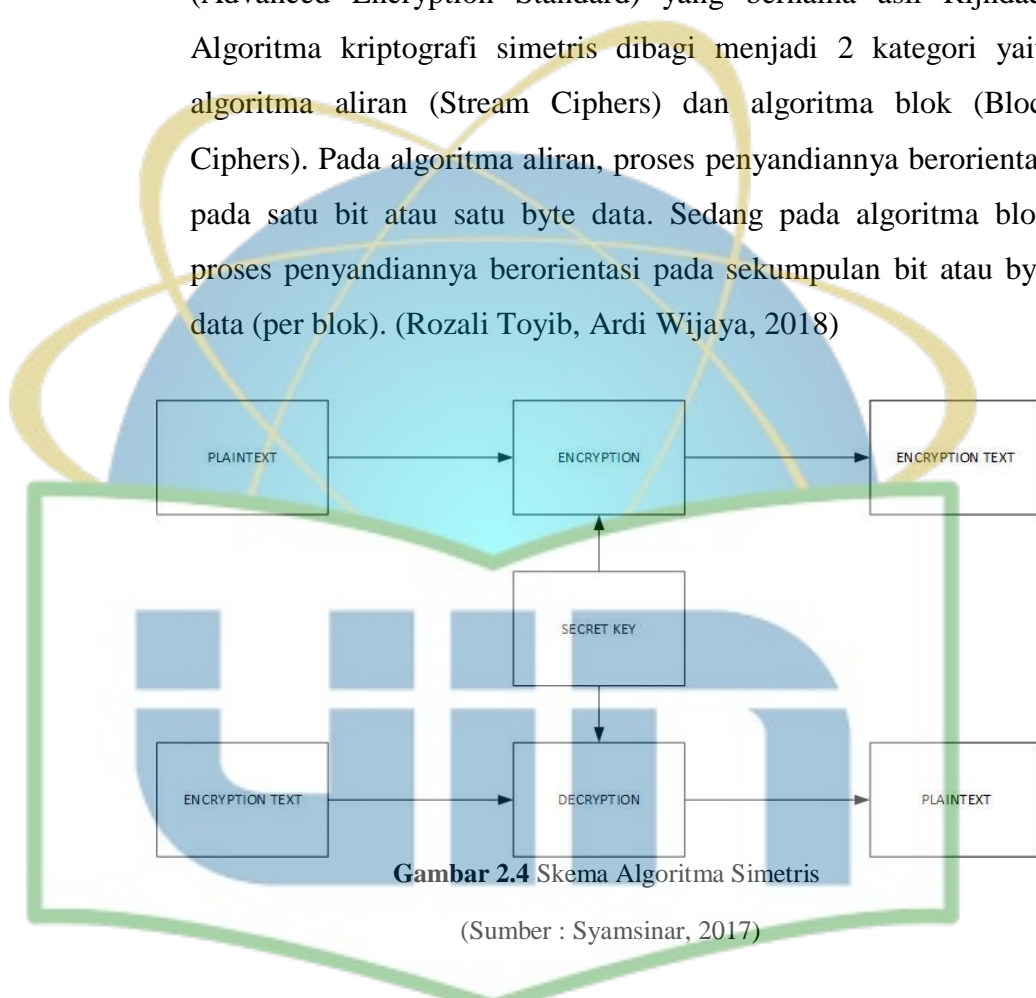
### 2.6.5 Jenis Algoritma Kriptografi

Berdasarkan jenis kunci yang digunakannya, algoritma kriptografi dikelompokkan menjadi dua bagian, yaitu algoritma simetris dan algoritma asimetris.

#### 2.6.5.1 Algoritma Simetris

Algoritma simetris dapat juga disebut sebagai algoritma konvensional, dimana kunci dekripsi dapat ditentukan dari kunci enkripsinya, begitu pula sebaliknya. Pada algoritma simetrik, kunci enkripsi dan kunci dekripsinya sama. Keamanan dari algoritma ini terletak pada kuncinya, jika kunci diberitahukan atau dibocorkan maka siapa saja dapat mengenkrip dan mendekrip data, jadi kunci harus benar-benar rahasia dan aman.

Algoritma simetris atau disebut juga algoritma kriptografi konvensional adalah algoritma yang menggunakan kunci yang sama untuk proses enkripsi dan proses dekripsi. Contoh algoritma kunci simetris adalah DES (Data Encryption Standard), blowfish, twofish, MARS, IDEA, 3DES (DES diaplikasikan 3 kali), AES (Advanced Encryption Standard) yang bernama asli Rijndael. Algoritma kriptografi simetris dibagi menjadi 2 kategori yaitu algoritma aliran (Stream Ciphers) dan algoritma blok (Block Ciphers). Pada algoritma aliran, proses penyandiannya berorientasi pada satu bit atau satu byte data. Sedangkan pada algoritma blok, proses penyandiannya berorientasi pada sekumpulan bit atau byte data (per blok). (Rozali Toyib, Ardi Wijaya, 2018)



**Gambar 2.4** Skema Algoritma Simetris

(Sumber : Syamsinar, 2017)

Berikut adalah Kelebihan dan Kekurangan dari Algoritma Simetris. (Syamsinar, 2017):

Kelebihan Algoritma Simetris:

- a. Algoritma kriptografi simetri dirancang dengan proses enkripsi/ deskripsi membutuhkan waktu yang singkat.

- b. Algoritma kriptografi simetri dapat disusun untuk menghasilkan cipher yang lebih kuat.
- c. Autentifikasi pengirim pesan langsung diketahui dari ciphertext yang diterima, karena kunci hanya diketahui oleh pengirim dan penerima pesan saja.

#### Kekurangan Algoritma Simetris:

- a. Kunci simetri harus dikirim melalui saluran yang aman. Kedua entitas yang berkomunikasi harus menjaga kerahasiaan kunci ini.
- b. Kunci harus sering diubah, mungkin pada setiap sesi komunikasi.

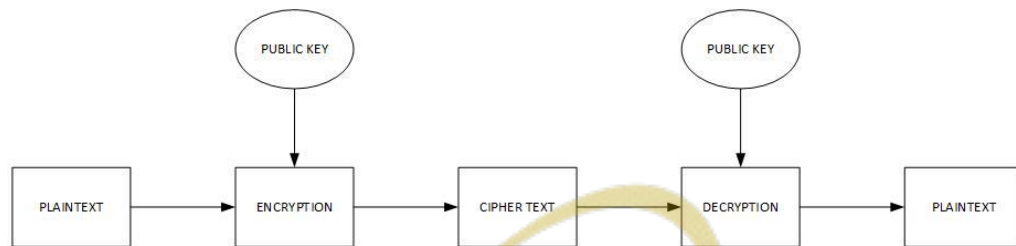
#### 2.7.2.2 Algoritma Asimetris

Algoritma Asimetris adalah pasangan kunci kriptografi yang salah satunya digunakan untuk proses enkripsi dan satu lagi lagi deskripsi. Semua orang yang mendapatkan kunci publik dapat menggunakannya untuk mengenkripsi suatu pesan, sedangkan hanya satu orang saja yang memiliki rahasia itu, yang dalam hal ini kunci rahasia, untuk melakukan pembongkaran terhadap kode yang dikirim untuknya. Contoh algoritma terkenal yang menggunakan kunci asimetris adalah RSA (merupakan singkatan dari nama penemunya, yakni Rivest, Shamir dan Adleman). (Sumandri, 2017)

Algoritma tak simetri sering juga disebut dengan algoritma kunci publik, dengan arti kata kunci yang digunakan untuk melakukan enkripsi dan dekripsinya berbeda. Pada algoritma tak simetri kunci terbagi menjadi 2 (dua) bagian (Geby Geta Putri, Wiwin Styorini, Rizki Dian Rahayani, 2018):

- a. Kunci umum (public key) adalah kunci yang dapat dan boleh diketahui oleh semua orang.

- b. Kunci pribadi (private key) adalah kunci yang hanya dapat diketahui penerima dan bersifat rahasia.



**Gambar 2.1** Skema Algoritma Asimetris

(Sumber: Syamsinar, 2017)

Berikut adalah Kelebihan dan Kekurangan dari Algoritma Asimetris. (Syamsinar, 2017):

Kelebihan Algoritma Asimetris:

- a. Masalah keamanan pada distribusi kunci dapat lebih baik, karena hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi.
- b. Pasangan kunci publik/kunci privat tidak perlu diubah bahkan dalam periode waktu yang panjang.
- c. Dapat digunakan untuk mengamankan pengiriman kunci simetris.
- d. Beberapa algoritma kunci-publik dapat digunakan untuk memberi tanda tangan digital pada pesan.

Kekurangan Algoritma Asimetris:

- a. Masalah keamanan pada distribusi kunci dapat lebih baik, karena hanya kunci privat yang perlu dijaga kerahasiaannya oleh setiap entitas yang berkomunikasi.

- b. Pasangan kunci publik/kunci privat tidak perlu diubah bahkan dalam periode waktu yang panjang.
- c. Dapat digunakan untuk mengamankan pengiriman kunci simetris.
- d. Beberapa algoritma kunci-publik dapat digunakan untuk memberi tanda tangan digital pada pesan

## 2.7 Advanced Encryption Standard (AES) / Rijindael

DES (*Data Encryption Standard*) sudah berakhir masa penggunaannya sebagai standard enkripsi kriptografi simetri. DES dianggap sudah tidak aman lagi karena perangkat keras khusus kunci enkripsi sudah bisa ditemukan dalam waktu yang singkat. *National Institute of Standards and Technology (NIST)* sebagai agensi Departemen Perdagangan AS mengusulkan kepada Pemerintah Federal AS untuk sebuah standard kriptografi kriptografi yang baru.

Untuk menghindari kontroversi mengenai standard yang baru tersebut, sebagaimana pada pembuatan DES (NSA sering dicurigai mempunyai “pintu belakang” atau *trapdoor* untuk mengungkap cipherteks yang dihasilkan oleh DES tanpa mengetahui kunci), maka NIST mengadakan sayembara terbuka untuk membuat standard algoritma kriptografi yang baru sebagai pengganti DES. Standard tersebut kelak diberi nama *Advanced Encryption Standard (AES)*.

Persyaratan yang diajukan oleh NIST tentang algoritma yang baru tersebut adalah:

1. Algoritma yang ditawarkan termasuk ke dalam kelompok algoritma kriptografi simetri berbasis *cipher* blok.
2. Seluruh rancangan algoritma harus publik (tidak dirahasiakan).
3. Ukuran blok yang dienkripsi adalah 128 bit.



4. Algoritma dapat diimplementasikan baik sebagai software maupun hardware.

NIST menerima 15 proposal algoritma yang masuk. Konferensi umum pun diselenggarakan untuk menilai keamanan algoritma yang diusulkan. Pada bulan Agustus 1998, NIST memilih 5 finalis yang didasarkan pada aspek keamanan algoritma, kemangkusan (*efficiency*), fleksibilitas, dan kebutuhan memori (penting untuk embedded system). Finalis tersebut adalah:

1. *Rijindael* (dari Vincent Rijmen dan Joan Daemen – Belgia, 86 Suara).
2. *Serpent* (dari Ross Anderson, Eli Biham, dan Lard Knudsen – Inggris, Israel, dan Norwegia, 59 Suara).
3. *Twofish* (dari tim yang diketuai oleh Bruce Schneier – USA, 31 Suara).
4. *RC6* (dari Laboratorium RSA – USA, 23 Suara).
5. *MARS* (dari IBM, 13 Suara)

Pada bulan Oktober 2000, NIST mengumumkan untuk memilih Rijindael, dan pada bulan November 2001, Rijindael ditetapkan sebagai AES, dan diharapkan Rijindael menjadi standard kriptografi yang dominan paling sedikit selama 10 tahun.

### 2.7.1 Penjelasan Algoritma AES (Advanced Encryption Standard)

Algoritma Rijindael (AES) beroperasi pada medan Galois  $GF(2^8)$ , ini artinya semua operasi aritmetika dilakukan pada *byte* berukuran 8 bit di dalam  $GF(2^8)$ . Rijindael (AES) mendukung panjang kunci 128 bit sampai 256 bit dengan step 32 bit. Panjang kunci dan ukuran blok dapat dipilih secara independen. Setiap blok dienkripsi dalam sejumlah putaran tertentu sebagaimana halnya pada DES.

Karena AES menetapkan panjang kunci adalah 128, 192, dan 256, maka dikenal AES-128, AES-192, dan AES-256. Tabel meresmikan perbedaan ketiga versi AES tersebut.

	Panjang Kunci (Nk words)	Ukuran Blok (Nb words)	Jumlah Putaran (Nr)
AES-128	4	4	10
AES-192	6	4	12
AES-256	8	4	14

Tabel 2.1 Tiga Buah Versi AES

**1 word = 32 bit**

Secara de-fakto, hanya ada 2 varian AES, yaitu AES-128 dan AES-256, karena akan sangat jarang pengguna menggunakan kunci yang panjang nya 192 bit. Karena AES mempunyai panjang kunci paling sedikit 128 bit, maka AES tahan terhadap serangan *brute-force (exhaustive key search)* dengan teknologi saat ini. Dengan panjang kunci 128-bit, maka terdapat sebanyak

$$2^{128} = 3,4 \times 10^{38}$$

kemungkinan kunci. Jika digunakan komputer tercepat yang dapat mencoba 1 juta kunci setiap detik, maka akan dibutuhkan waktu  $5,4 \times 10^{24}$  tahun untuk mencoba seluruh kemungkinan kunci. Jika digunakan komputer tercepat yang dapat mencoba 1 juta kunci setiap milidetik, maka akan dibutuhkan waktu  $5,4 \times 10^{18}$  tahun untuk mencoba seluruh kemungkinan kunci.

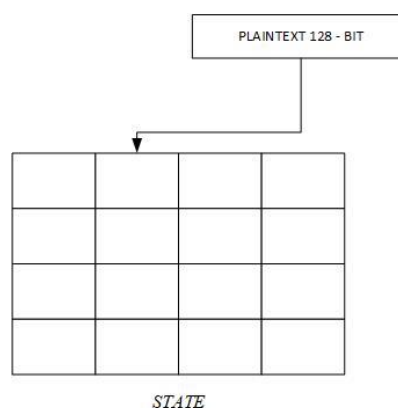
Seperti pada DES, *Rijndael* (AES) menggunakan substitusi dan permutasi, dan sejumlah putaran (*cipher* berulang) – setiap

putaran menggunakan kunci internal yang berbeda (kunci setiap putaran disebut *round key*). Tetapi tidak seperti DES yang beroperasi dalam bit, *Rijindael* beroperasi dalam *byte* untuk memangkuskan implementasi algoritma kedalam *software* dan *hardware*. Etipa byte dinyatakan dalam notasi hexsadesimal. Misalnya byte 11010100 dalam notasi hexadesimal dalah D<sub>4</sub> (1101 = D, 0100 = 4). Perbedaan lain dengan DES adalah *Rijindael* tidak menggunakan jaringan Feistel.

Algoritma *Rijindael* (AES) mempunyai 3 parameter:

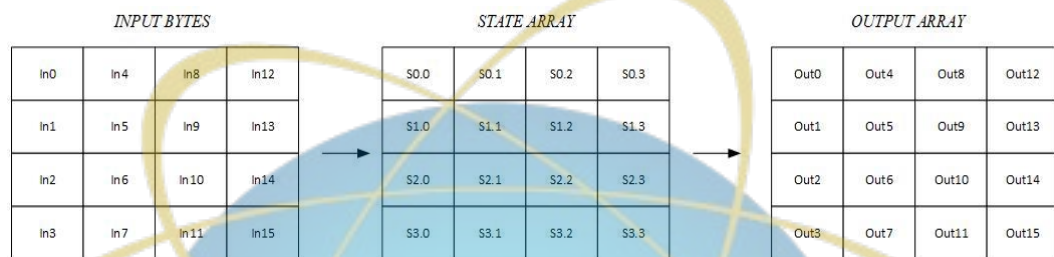
1. *In* : larik yang berukuran 16-byte, yang berisi data masukan
2. *Out* : larik yang berukuran 16-byte, yang berisi hasil enkripsi
3. *Key* : larik yang berukuran 16 – byte, yang berisi kunci *ciphering* (disebut juga cipher key)

Dengan 16 byte, maka baik blok data dan kunci yang berukuran 128-bit dapat disimpan didalam ketiga larik tersebut ( $128 = 16 \times 8$ ). Selama kalkulasi plainteks menjadi cipherteks, status data sekarang disimpan didalam matriks dua dimensi, *state*, bertipte byte dan berukuran *Nrows* x *Ncols*. Untuk blok data 128-bit, ukuran *state* adalah 4 x 4. Elemen matriks *state* diacu sebagai  $S[r.c]$ , dengan  $0 \leq r < 4$  dan  $0 \leq c < Nb$  ( $Nb$  adalah panjang blok dibagi 32. Pada AES -128,  $Nb = 128/32 = 4$ ).



**Gambar 2.6** Matriks state berukuran 4 x 4 untuk blok pesan 128-bit

Pada awal enkripsi, 16-byte data masukan (plaintexts),  $in_0, in_1, \dots, in_{15}$  disalin ke dalam matriks *state* (direalisasikan oleh fungsi *CopyInToState* (*state*, *in*)).



Contoh sebuah *state*:

**Table 2.2** Contoh Sebuah State

23	A2	BC	4A
D4	03	97	F3
16	48	CD	50
FF	DA	10	64

**Gambar 2.7** Penyalinan byte – byte pesan ke dalam matriks state

Operasi enkripsi/dekripsi dilakukan terhadap matriks *S*, dan luarannya ditampung didalam larik *out*. Skema penyalinan larik masukan *in* ke matriks *S* adalah sebagai berikut:

$$S[r.c] \leftarrow in[r + 4c] \text{ untuk } 0 \leq r < 4 \text{ dan } 0 \leq c < Nb$$

Skema penyalinan matriks *S* ke larik luaran *out* adalah sebagai berikut:

$$out[r + 4c] \leftarrow S[r, c] \text{ untuk } 0 \leq r < 4 \text{ dan } 0 \leq c < Nb$$

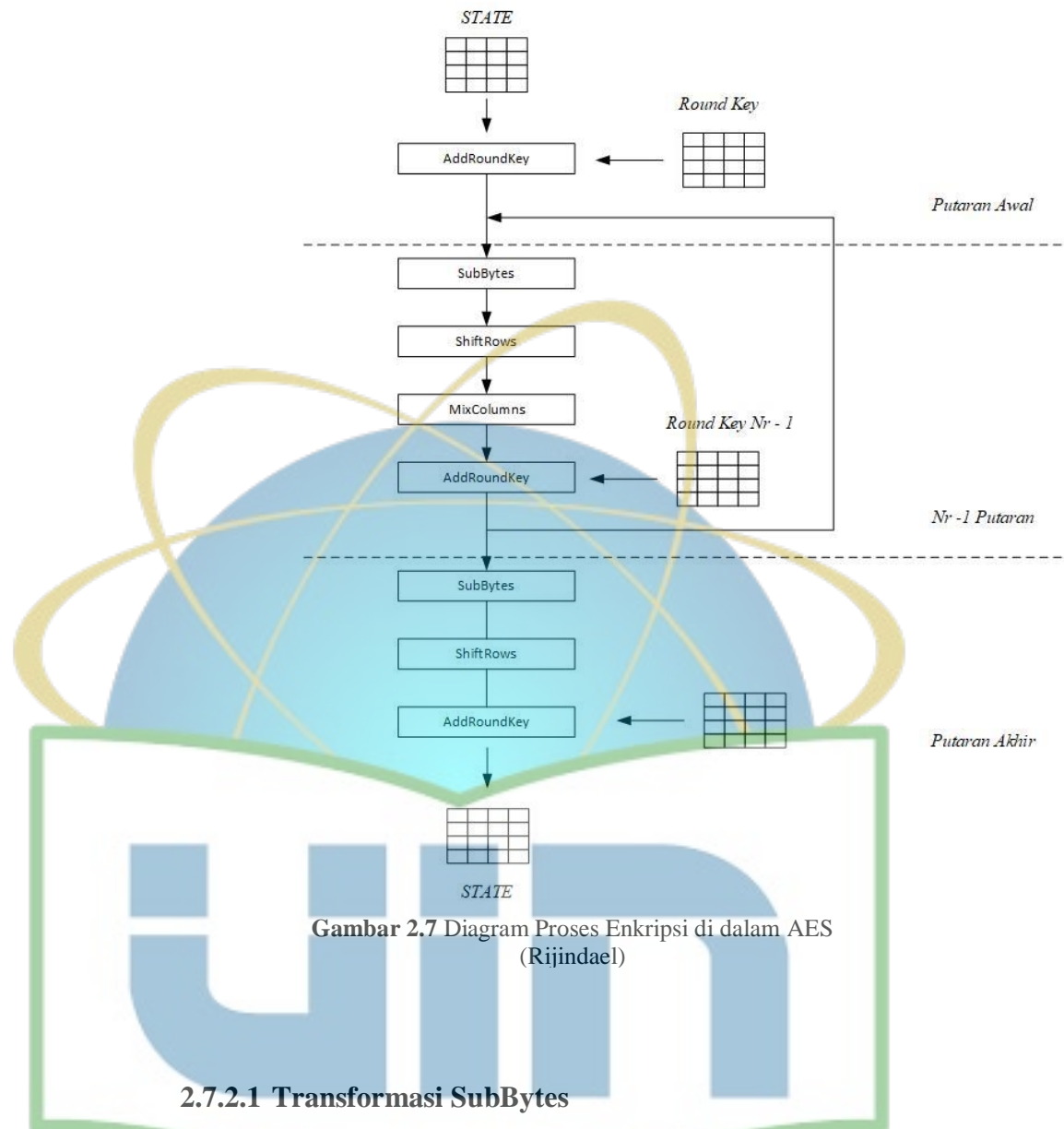
### 2.7.2 Proses Algoritma Advanced Encryption Standard (AES)

Garis besar Algoritma *Rijindael* (AES) yang beroperasi pada blok 128-bit dengan kunci 128-bit adalah sebagai berikut (di luar proses pembangkitan round key):

1. AddRoundKey: melakukan XOR antara state awal (plainteks) dengan cipher key. Tahap ini disebut juga putaran awal.
2. Putaran sebanyak  $N_r - 1$  kali. Proses yang dilakukan pada setiap putaran adalah:
  - a. SubBytes: substitusi byte dengan menggunakan tabel substitusi (S-Box).
  - b. ShiftRows: pergeseran baris – baris array state secara wrapping.
  - c. MixColumns: mengacak data di masing – masing kolom array state.
  - d. AddRoundKey: melakukan XOR antara state sekarang roundkey
3. Proses untuk putaran terakhir:
  - a. SubBytes
  - b. ShiftRows
  - c. AddRoundKey

Hasil putaran akhir adalah cipherteks yang dinyatakan di dalam state.





Gambar 2.7 Diagram Proses Enkripsi di dalam AES (Rijindael)

### 2.7.2.1 Transformasi SubBytes

Transformasi SubBytes merupakan operasi substitusi di dalam *Rijindael* (AES) berdasarkan tabel *lookup*. SubBytes memetakan setiap byte dari larik state dengan menggunakan tabel substitusi S-box. Tidak seperti DES yang mempunyai S-box berbeda pada setiap putaran. AES hanya mempunyai satu buah S-box. Tabel S-box berupa matriks berukuran 16 x 16 yang entrinya merupakan permutasi dari 256 kemungkinan susunan karakter heksadesimal. Karakter heksadesimal ada 16 kemungkinan

(0, 1, 2,, ..., D, E, F). karena satu byte terdiri dari 8 bit dan satu karakter heksadesimal panjangnya 4 bit, maka satu byte dinyatakan dalam dua karakter heksadesimal.

Oleh karena itu, jumlah kemungkinan representasi satu byte sebagai dua karakter heksadesimal  $16 \times 16 = 256$  kemungkinan, sehingga ada 256 elemen matriks S-box.

Angka-angka di dalam tabel S-box tersebut terlihat seperti di isi secara acak, namun sebenarnya itu dihasilkan dari proses perhitungan sebagai berikut:

1. Insialisasi S-box dengan nilai yang menarik dari baris ke baris. Baris ke-0 diisi dengan nilai 00, 01, 02, .....,0F. Baris ke-1 diisi dengan nilai 10, 11, 12, ...,1F.
2. Untuk setiap nilai pada baris y kolom x, tentukan balikan nya dalam  $GF(2^8)$ . Nilai 00 dipetakan ke dirinya sendiri.
3. Hasil dari langkah 2 dikonversi ke vektor kolom bit  $(b_0, b_1, \dots, b_7)^T$ .
4. Kalikan vektor kolom bit  $(b_0, b_1, \dots, b_7)^T$  dengan sebuah matriks *affine* sebagai berikut:

Table 2.3 Matriks Affine

$$\begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

5. Selanjutnya, konversi hasil perhitungan  $(b'_0, b'_1, \dots, b'_8)^T$  ke dalam heksadesimal, menjadi elemen S-box(x,y).

**Table 2.4** Table S-Box Didalam AES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	8B	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Operasi substitusi menggunakan S-box sangat sederhana, caranya adalah sebagai berikut: untuk setiap byte pada matriks *state* misalkan  $S[r, c] = xy$ , yang dalam hal ini  $xy$  adalah digit heksadesimal dari nilai  $S[r, c]$ , maka nilai substitusinya adalah  $S'[r, c]$ , yang merupakan perpotongan baris  $x$  dengan kolom  $y$  di dalam S-box.

**Table 2.5** Transformasi SubBytes

$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$	$b_{00}$	$b_{01}$	$b_{02}$	$b_{03}$
$a_{10}$	$a_{11}$	$a_{12}$	$a_{13}$	$b_{10}$	$b_{11}$	$b_{12}$	$b_{13}$
$a_{20}$	$a_{21}$	$a_{22}$	$a_{23}$	$b_{20}$	$b_{21}$	$b_{22}$	$b_{23}$
$a_{30}$	$a_{31}$	$a_{32}$	$a_{33}$	$b_{30}$	$b_{31}$	$b_{32}$	$b_{33}$

S-Box

Sebagai contoh, *state* adalah sebagai berikut:

23	A2	BC	4A
D4	03	97	F3
16	48	CD	50
FF	DA	10	64

Elemen pertama, 23, disubstitusi dengan elemen pada perpotongan baris 2 dengan kolom 3 pada S-box, yaitu 26, diperlihatkan sebagai berikut :

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	8B	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Jadi operasi substitusi pada  $S[o, o] = 23$  menghasilkan  $S'[o, o] = 26$ . Jika operasi substitusi diteruskan untuk semua elemen *state* lain nya, maka hasilnya adalah sebagai berikut:



23	A2	BC	4A
D4	03	97	F3
16	48	CD	50
FF	DA	10	64

→

26	3A	65	D6
48	7B	88	0D
47	52	BD	State
16	57	CA	43

### 2.7.2.2 Transformasi ShiftRows

Transformasi *ShiftRows* melakukan pergeseran secara wrapping (siklik) pada 3 baris terakhir dari matriks *state*. Jumlah pergeseran bergantung pada nilai baris (*r*). Elemen – elemen pada baris  $r = 1$  digeser sejauh 1 byte ke kiri, elemen – elemen pada baris  $r = 2$  digeser sejauh 2 byte ke kiri dan elemen – elemen pada baris  $r = 3$  digeser sejauh 3 byte, baris  $r = 0$  tidak digeser.

Table 2.6 Transformasi ShiftRows()

	$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$
Geser 1	$a_{10}$	$a_{11}$	$a_{12}$	$a_{13}$
Geser 2	$a_{20}$	$a_{21}$	$a_{22}$	$a_{23}$
Geser 3	$a_{30}$	$a_{31}$	$a_{32}$	$a_{33}$

ShiftRows() →

$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$
$a_{11}$	$a_{12}$	$a_{12}$	$a_{10}$
$a_{22}$	$a_{23}$	$a_{20}$	$a_{21}$
$a_{33}$	$a_{30}$	$a_{31}$	$a_{32}$

Sebagai contoh, hasil SubBytes sebelumnya dilakukan transformasi ShiftRows, hasilnya adalah seperti pada gambar.

**Table 2.7** Transformasi ShiftRows terhadap state hasil

26	3A	65	D6
48	7B	88	0D
47	52	BD	53
16	57	CA	43

 $\xrightarrow{\text{ShiftRows()}}$ 

26	3A	65	D6
7B	88	0D	48
BD	53	47	52
43	16	57	CA

### 2.7.2.3 Transformasi MixColumns

Transformasi MixColumns mengalikan matriks state dengan sebuah matriks tertentu sebaga berikut:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$

MixColumns memberikan efek difusi pada cipher. Setiap kolom diperlakukan sebagai polinum 4-suku pada  $GF(2^8)$ . Transformasi MixColumn pada sebuah kolom matriks state dinyatakan sebagai.

$$\begin{bmatrix} S'_{0,j} \\ S'_{1,j} \\ S'_{2,j} \\ S'_{3,j} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} S_{0,j} \\ S_{1,j} \\ S_{2,j} \\ S_{3,j} \end{bmatrix}$$

$$S'_{0,j} = (\{02\} \bullet S_{0,j}) \oplus (\{03\} \bullet S_{1,j}) \oplus (\{03\} \bullet S_{2,j}) \oplus S_{3,j}$$

$$S'_{1,j} = S_{3,j} \oplus (\{02\} \bullet S_{1,j}) \oplus (\{03\} \bullet S_{2,j}) \oplus S_{0,j}$$

$$S'_{2,j} = S_{0,j} \oplus S_{1,j} \oplus (\{02\} \bullet S_{1,j}) \oplus (\{03\} \bullet S_{3,j})$$

$$S'_{3,j} = (\{03\} \bullet S_{0,j}) \oplus S_{0,j} \oplus S_{1,j} \oplus (\{02\} \bullet S_{3,j})$$

Simbol  $\bullet$  menyatakan perkalian dalam  $GF(2^8)$ , sedangkan simbol  $\otimes$  menyatakan operator bitwise XOR. Gambar

**Table 2.8** Transformasi MixColumns

$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$	$b_{00}$	$b_{01}$	$b_{02}$	$b_{03}$
$a_{10}$	$a_{11}$	$a_{12}$	$a_{13}$	$b_{10}$	$b_{11}$	$b_{12}$	$b_{13}$
$a_{20}$	$a_{21}$	$a_{22}$	$a_{23}$	$b_{20}$	$b_{21}$	$b_{22}$	$b_{23}$
$a_{30}$	$a_{31}$	$a_{32}$	$a_{33}$	$b_{30}$	$b_{31}$	$b_{32}$	$b_{33}$

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} \\ \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \\ \\ \end{bmatrix}$$

Sebagai contoh, lihat kembali hasil transformasi *ShiftRows* sebelumnya:

26	3A	65	D6
7B	88	0D	48
BD	53	47	52
CA	16	57	CA

Operasi MixColumns terhadap kolom pertama:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 26 \\ 7B \\ BD \\ 43 \end{bmatrix} = \begin{bmatrix} 3F \\ 4F \\ F9 \\ 2A \end{bmatrix}$$

$$(02 \bullet 26) \oplus (03 \bullet 7B) \oplus (01 \bullet BD) \oplus (01 \bullet 43) = 3F$$

$$(01 \bullet 26) \oplus (02 \bullet 7B) \oplus (03 \bullet BD) \oplus (01 \bullet 43) = 4F$$

$$(01 \bullet 26) \oplus (01 \bullet 7B) \oplus (02 \bullet BD) \oplus (03 \bullet 43) = F9$$

$$(03 \bullet 26) \oplus (01 \bullet 7B) \oplus (01 \bullet BD) \oplus (01 \bullet 43) = 2A$$

Rijindael (AES) menggunakan **polinom irreducible**  $x^8 + x^4 + x^3 + x + 1$  sebagai modulus dalam operasi perkalian ( $\cdot$ ). Pembagian dengan modulus tersebut diperlukan jika hasil perkalian polinum berderajat  $\geq 8$ .

Hasil perhitungan *MixColumn* diatas diperoleh sebagai berikut:

$$(02 \bullet 26) = (0000\ 0010) \times (0010\ 0110) = x \times (x^5 + x^2 + x) \bmod (x^8 + x^4 + x^3 + x + 1)$$

$$= (x^6 + x^3 + x^2) \bmod (x^8 + x^4 + x^3 + x + 1)$$

$$= x^6 + x^3 + x^2$$

$$= (01001100)$$

$$= 4C$$

$$(03 \bullet 7B) = (0000\ 0011) \times (0111\ 0011) = (x + 1) (x^6 + x^5 + x^4 + x^3 + x + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$$

$$= ((x^7 + x^6 + x^5 + x^4 + x^3 + x^2 + x) \bmod (x^8 + x^4 + x^3 + x + 1))$$

$$(x^6 + x^5 + x^4 + x^3 + x + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$$

$$= (x^7 + (1+1)x^6 + (1+1)x^5 + (1+1)x^4 + x^3 + x^2 + (1+1)x + 1) \bmod (x^8 + x^4 + x^3 + x + 1)$$

$$= (x^7 + x^3 + x^2 + 1)$$

$$= (1000\ 1101)$$

$$= 8D$$

$$(01 \bullet BD) = BD = 10111101$$

$$(01 \bullet 43) = 43 = 01000011$$

Selanjutnya, XOR-kan semua hasil antara tersebut:

$$(02 \bullet 26) = 0100\ 1100$$

$$(03 \bullet 7B) = 1000\ 1101$$

$$(01 \bullet BD) = 1011\ 1101$$

$$(01 \bullet 43) = 0100\ 0011 \oplus$$

$$0011\ 1111 = 3F$$

$$\text{Jadi, } (02 \bullet 26) \oplus (03 \bullet 7B) \oplus (01 \bullet BD) \oplus (01 \bullet 43) = 3F$$

Persamaan lainnya diselesaikan dengan cara yang sama.

Menghitung perkalian seperti di atas tentu saja sangat kompleks.

Ada dua cara yang lebih praktis yang dapat digunakan:

*Cara pertama:* perkalian sebuah nilai dengan  $x$  (yaitu 02) dapat diimplementasikan sebagai pergeseran 1 bit pada nilai tersebut di ikuti *bitwise* XOR dengan 0001 1011 (atau 1B) apabila bit paling kiri dari nilai tersebut adalah satu. Pada contoh di atas,



$(03 \bullet 7B) = (00000010) \times (00100110) = 00100110 \ll 1$  (geser ke kiri satu bit)

= 01001100 (tidak perlu *bitwise XOR*)

Perkalian dengan pangkat  $x$  lebih tinggi diperoleh dengan mengaplikasikan berulang – ulang aturan di atas dan menjumlahkan hasil antaranya.

*Cara kedua:* cara ini lebih mudah karena menggunakan dua tabel. Tabel E dan Tabel L sebagai berikut:

**Table 2.9** Table E untuk membantu perhitungan MixColumns()

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	01	03	05	0F	11	33	55	FF	1A	DE	72	96	A1	F8	13	35
1	5F	E1	38	48	D8	73	95	A4	F7	02	06	0A	1E	22	66	AA
2	E5	34	5C	F4	37	59	EB	26	6A	BE	D9	70	90	AB	E6	31
3	53	F5	04	0C	14	3C	44	CC	4F	D1	68	B8	D3	6E	B2	CD
4	4C	D4	67	A9	E0	3B	4D	D7	62	A6	F1	08	18	28	78	88
5	83	9E	B9	D0	6B	BD	DC	7F	81	98	B3	CE	49	DB	76	9A
6	B5	C4	57	F9	10	30	50	F0	0B	1D	27	69	BB	D6	61	A3
7	FE	19	2B	7D	87	92	AD	EC	2F	71	93	AE	E9	20	60	A0
8	FB	16	3A	4E	D2	6D	B7	C2	5D	E7	32	56	FA	15	3F	41
9	C3	5E	E2	3D	47	C9	40	C0	5B	ED	2C	74	9C	BF	DA	75
A	9F	BA	D5	64	AC	EF	2A	7E	82	9D	BC	DF	7A	8E	89	80
B	9B	B6	C1	58	E8	23	65	AF	EA	25	6F	B1	C8	43	C5	54
C	FC	1F	21	63	A5	F4	07	09	1B	2D	77	99	B0	CB	46	CA
D	45	CF	4A	DE	79	8B	86	91	A8	E3	3E	42	C6	51	F3	0E
E	12	36	5A	EE	29	7B	8D	8C	8F	8A	85	94	A7	F2	0D	17
F	39	4B	DD	7C	84	97	A2	FD	1C	24	6C	B4	C7	52	F6	01

**Table 2.10** Table L untuk membantu perhitungan MixColumns()

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	0	19	1	32	2	1A	C6	4B	C7	1B	68	33	EE	DF	03	
1	64	4	E0	0E	34	8D	81	EF	4C	71	8	C8	F8	69	1C	C1
2	7D	C2	1D	B5	F9	B9	27	6A	4D	E4	A6	72	9A	C9	9	78
3	65	2F	8A	5	21	0F	E1	24	12	F0	82	45	35	93	DA	8E
4	96	8F	DB	BD	36	D0	CE	94	13	5C	D2	F1	40	46	83	38
5	66	DD	FD	30	BF	6	8B	62	B3	25	E2	98	22	88	91	10
6	7E	6E	48	C3	A3	B6	1E	42	3A	6B	28	54	FA	85	3D	BA
7	2B	79	0A	15	9B	9F	5E	CA	4E	D4	AC	E5	F3	73	A7	57
8	AF	58	A8	50	F4	EA	D6	74	4F	AE	E9	D5	E7	E6	AD	E8
9	2C	D7	75	7A	EB	16	0B	F5	59	CB	5F	B0	9C	A9	51	A0
A	7F	0C	F6	6F	17	C4	49	EC	D8	43	1F	2D	A4	76	7B	B7
B	CC	BB	3E	5A	FB	60	B1	86	3B	52	A1	6C	AA	55	29	9D
C	97	B2	87	90	61	BE	DC	FC	BC	95	CF	CD	37	3F	5B	D1
D	53	39	84	3C	41	A2	6D	47	14	2A	9E	5D	56	F2	D3	AB
E	44	11	92	D9	23	20	2E	89	B4	7C	B8	26	77	99	E3	A5
F	67	4A	ED	DE	C5	31	FE	18	0D	63	8C	80	C0	F7	70	07

Cara menggunakan tabel L dan tabel E adalah sebagai berikut:

1. Misalkan peneliti akan menghitung  $(02 \bullet 26)$ . Cari L(02) yaitu perpotongan baris 0 dan kolom 2 pada L, kemudian cari L(26) yaitu perpotongan baris 2 dan kolom 6 pada Tabel 1. Diperoleh  $L(02) = 19$  dan  $L(26) = 27$ . Selanjutnya jumlahkan dan 27 dalam basis 16, yaitu  $19 + 27 = 40$ .
2. Selanjutnya cari E(40), yaitu perpotongan baris 4 dan kolom 0 pada tabel E. Diperoleh  $E(40) = 4C$ . Jadi,  $(02 \bullet 26) = 4C$ .

3. Apabila penjumlahan menghasilkan nilai  $> FF$ , maka kurang nilai tersebut dengan  $FF$ , Misalnya  $A8 + 7B = 123$ . Karena  $123 > FF$ , maka lakukan  $123 - FF = 24$ .
4. Perkalian dengan 1 dan 0 tidak perlu melihat tabel di atas.  
Contohnya,  $3D \bullet 1 = 3D$ , dan  $BC \bullet 0 = 0$

Dengan menggunakan cara kedua ini, maka

$$(03 \bullet 7B) = E(L(03) + L(7B)) = E(01 + E5) = E(01 + E5) = E(E6) = 8D = 1000\ 1101$$

$$(01 \bullet BD) = 1011\ 1101$$

$$(03 \bullet 7B) = 0100\ 0011$$

Selanjutnya,

$$(02 \bullet 26) \oplus (03 \bullet 7B) \oplus (01 \bullet BD) \oplus (01 \bullet 43) = 4C \oplus 8D \oplus 43 = 3F$$

Lakukan perhitungan untuk yang lain nya:

$$\begin{aligned} &(01 \bullet 26) \oplus (03 \bullet 7B) \oplus (01 \bullet BD) \oplus (01 \bullet 43) \\ &= 26 \oplus E(L(02) + L(7B)) \oplus E(L(03) + L(BD)) \oplus 43 \\ &= 26 \oplus E(19) + E5 \oplus E(01 + 55) \oplus 43 \\ &= 26 \oplus E(FE) \oplus E(56) \oplus 43 \\ &= 26 \oplus F6 \oplus DC \oplus 43 \\ &= 4F \end{aligned}$$

$$\begin{aligned} &(01 \bullet 26) \oplus (01 \bullet 7B) \oplus (02 \bullet BD) \oplus (03 \bullet 43) \\ &= 26 \oplus 7B \oplus E(L(02) + L(BD)) \oplus E(L(03) + L(43)) \\ &= 26 \oplus 7B \oplus E(19 + 55) \oplus E(01 + BD)) \\ &= 26 \oplus 7B \oplus E(6E) \oplus E(BE) \end{aligned}$$

$$= 26 \oplus 7B \oplus 61 \oplus C5$$

$$= F9$$

$$(03 \bullet 26) \oplus (01 \bullet 7B) \oplus (01 \bullet BD) \oplus (02 \bullet 43)$$

$$= E(L(03) + L(26)) \oplus 7B \oplus BD \oplus E(L(02) + L(43))$$

$$= E(01 + 27) \oplus 7B \oplus BD \oplus E(19 + BD)$$

$$= E(28) \oplus 7B \oplus BD \oplus E(D6)$$

$$= 6A \oplus 7B \oplus BD \oplus 86$$

$$= 2A$$

a. Operasi MixColumns terhadap kolom kedua:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 3A \\ 88 \\ 53 \\ 16 \end{bmatrix} = \begin{bmatrix} B2 \\ D2 \\ 2E \\ B9 \end{bmatrix}$$

$$(02 \bullet 3A) \oplus (03 \bullet 88) \oplus (01 \bullet 53) \oplus (01 \bullet 16) = B2$$

$$(01 \bullet 3A) \oplus (02 \bullet 88) \oplus (03 \bullet 53) \oplus (01 \bullet 16) = D2$$

$$(01 \bullet 3A) \oplus (01 \bullet 88) \oplus (02 \bullet 53) \oplus (03 \bullet 16) = 2E$$

$$(03 \bullet 3A) \oplus (01 \bullet 88) \oplus (01 \bullet 53) \oplus (02 \bullet 16) = B9$$

b. Operasi MixColumn terhadap kolom ketiga:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 65 \\ 0D \\ 1F \\ 57 \end{bmatrix} = \begin{bmatrix} CD \\ E1 \\ 1F \\ 4B \end{bmatrix}$$

$$(02 \bullet 65) \oplus (03 \bullet 0D) \oplus (01 \bullet 47) \oplus (01 \bullet 57) = CD$$

$$(01 \bullet 65) \oplus (02 \bullet 0D) \oplus (03 \bullet 47) \oplus (01 \bullet 57) = E1$$

$$(01 \bullet 65) \oplus (01 \bullet 0D) \oplus (02 \bullet 47) \oplus (03 \bullet 57) = 1F$$

$$(03 \bullet 65) \oplus (01 \bullet 0D) \oplus (01 \bullet 47) \oplus (02 \bullet 57) = 4B$$

c. Operasi MixColumns terhadap kolom ke empat:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} D6 \\ 48 \\ 52 \\ CA \end{bmatrix} = \begin{bmatrix} F7 \\ 9E \\ 7F \\ F4 \end{bmatrix}$$

$$(02 \bullet D6) \oplus (03 \bullet 48) \oplus (01 \bullet 52) \oplus (01 \bullet CA) = F7$$

$$(01 \bullet D6) \oplus (02 \bullet 48) \oplus (03 \bullet 52) \oplus (01 \bullet CA) = 9E$$

$$(01 \bullet D6) \oplus (01 \bullet 48) \oplus (02 \bullet 52) \oplus (03 \bullet CA) = 7F$$

$$(03 \bullet D6) \oplus (01 \bullet 48) \oplus (01 \bullet 52) \oplus (02 \bullet CA) = F4$$

Hasil transformasi MixColumns seluruhnya:

26	3A	65	D6
48	7B	88	0D
47	52	BD	53
16	57	CA	43

*MixColumns()*

26	3A	65	D6
7B	88	0D	48
BD	53	47	52
43	16	57	CA

#### 2.7.2.4 Transformasi AddRoundKey()

Transformasi ini melakukan operasi bitwise XOR antara sebuah *roundkey* dengan matriks *state*, dan hasilnya disimpan pada matriks *state* yang sama (proses pembangkitan round key akan dijelaskan kemudian). Gambar ..... memperlihatkan transformasi *AddRoundKey*. Elemen – elemen pada matriks *state* di-XOR-kan dengan elemen yang bersesuaian pada matriks round key, yaitu  $a_{ij} \oplus k_{ij} = b_{ij}$ .



**Table 2.11** Transformasi AddRoundKey

<i>State</i>				<i>Round Key</i>				<i>State</i>			
a <sub>00</sub>	a <sub>01</sub>	a <sub>02</sub>	a <sub>03</sub>	k <sub>00</sub>	k <sub>01</sub>	k <sub>02</sub>	k <sub>03</sub>	b <sub>00</sub>	b <sub>01</sub>	b <sub>02</sub>	b <sub>03</sub>
a <sub>11</sub>	a <sub>12</sub>	a <sub>12</sub>	a <sub>10</sub>	k <sub>11</sub>	k <sub>12</sub>	k <sub>12</sub>	$\oplus$	b <sub>10</sub>	b <sub>11</sub>	b <sub>11</sub>	b <sub>11</sub>
a <sub>22</sub>	a <sub>23</sub>	a <sub>20</sub>	a <sub>21</sub>	k <sub>22</sub>	k <sub>23</sub>	k <sub>20</sub>	k <sub>21</sub>	b <sub>20</sub>	b <sub>21</sub>	b <sub>22</sub>	b <sub>23</sub>
a <sub>33</sub>	a <sub>30</sub>	a <sub>31</sub>	a <sub>32</sub>	k <sub>33</sub>	k <sub>30</sub>	k <sub>31</sub>	k <sub>32</sub>	b <sub>30</sub>	b <sub>31</sub>	b <sub>32</sub>	b <sub>33</sub>

Contoh sebuah round key adalah

4F	5A	7B	10
8C	CD	D1	23
67	2A	FF	45
28	0D	93	2C

Dari hasil transformasi *MixColumn* sebelumnya adalah

3F	B2	CD	F7
4F	D2	E1	9E
F9	2E	1F	7F
2A	B9	48	F4

Maka transformasi *AddRoundKey* menghasilkan:

3F	B2	CD	F7
4F	D2	E1	9E
F9	2E	1F	7F
2A	B9	48	F4

 $\oplus$ 

4F	5A	7B	10
8C	CD	D1	23
67	2A	FF	45
28	0D	93	2C

 $=$ 

70	E8	B6	E7
C3	1F	30	BD
9E	04	E0	3A
02	B4	D8	D8

### 2.7.2.5 Ekspansi Kunci

Setiap putaran di dalam algoritma *Rijindael* (AES) menggunakan kunci putaran atau round key. Kunci putaran dibangkitkan dari kunci eksternal dari pengguna yang dinamakan cipher key dan disimbolkan dengan *peubah key*. Pembangkitan semua kunci putaran dilakukan oleh fungsi *KeyExpansion()*.

Pembangkitan kunci putaran di dalam algoritma *Rijindael* (AES) tergolong rumit dan agak sukar diterangkan. Tinjau sebuah larik key yang panjangnya 16 byte (16 elemen) dan  $Nr = 10$  putaran. Sepuluh kunci putaran akan disimpan di dalam matriks *rk*. Elemen awal key langsung menjadi *rk[0]*. Elemen – elemen kunci lainnya akan disimpan di dalam *rk[1]*, *rk[2]*, ..., *rk[10]*.

Algoritma ekspansi kunci adalah sebagai berikut:

1. Salin elemen – elemen key ke dalam larik *w[0]*, *w[1]*, *w[2]*, *w[3]*.

Larik *w[0]* berisi empat elemen pertama key, *w[1]* berisi empat elemen berikutnya, dan seterusnya.

2. Mulai dari  $i = 4$  sampai 43, lakukan:
  - a. Simpan *w[i - 1]* ke dalam perubahan temp
  - b. Jika  $i$  kelipatan 4, lakukan fungsi *g* berikut:
    - Geser *w[i - 1]* satu byte ke kiri secara sirkuler

- Lakukan substitusi dengan S-box terhadap hasil pergeseran tersebut
- XOR-kan hasil di atas dengan round constant (Rcon) ke  $i/4$  (atau  $Rcon[i/4]$ ). Nilai Rcon berbeda – beda untuk  $RC[1] = 1$ ,  $RC[j] = 2 \bullet RC[j-1]$ , simbol  $\bullet$  menyatakan perkalian yang didefinisikan di dalam  $GF(2^8)$ . Nilai  $RC[j]$  di dalam heksadesimal adalah  $[STA_{11}]$ :  $RC[1] = 01$ ,  $RC[1] = 02$ ,  $RC[1] = 04$ ,  $RC[1] = 08$ ,  $RC[1] = 10$ ,  $RC[1] = 20$ ,  $RC[1] = 40$ ,  $RC[1] = 80$ ,  $RC[1] = 1B$ ,  $RC[1] = 36$ .
- Simpan hasil fungsi  $g$  ke dalam peubah temp

c. XOR-kan  $w[i-4]$  dengan temp

Sebagai contoh, misalkan key dalam heksadesimal adalah  $key = (54, 77, 6F, 20, 4F, 6E, 65, 20, 4E, 69, 6E, 65, 20, 54, 77, 6F)$

Dari key langsung diperoleh  $rk[0]$  sebagai berikut:

54	4F	4E	20
77	6E	69	54
6F	65	6E	77
20	20	65	6F

Proses dibawah ini hanya menunjukkan pembangkitan  $rk[1]$ :

1.  $w[0] = (54, 77, 6F, 20)$ ;  $w[1] = (4F, 6E, 65, 20)$ ;  $w[2] = (4E, 69, 6E, 65)$ ;  $w[3] = (20, 54, 77, 6F)$

2. mulai dari  $I = 4$ :

a)  $\text{temp} = w[3]$

b)  $i = 4$  adalah kelipatan 4, maka jalankan fungsi  $g$  terhadap  $w[3]$ :

- Geser  $w[3]$  satu byte ke kiri secara sirkuler:  
 $w[3] = (54, 77, 6E, 20)$ .
- Substitusi hasil pergeserann tersebut dengan S-box:  $(20, F5, 9F, B7)$ .
- XOR-kan hasil di atas dengan  $Rcon[1] = (01, 00, 00, 00)$ , menghasilkan  $(21, F5, 9F, B7)$
- Jadi,  $g(w[3]) = (21, F5, 9F, B7)$

c)  $w[4] = w[0] \oplus g(w[3]) = w[0] \oplus g(w[3]) = (75, 82, F0, 97)$  untuk  $i = 5, 6, 7$ :

$$w[5] = w[4] \oplus w[1] = (3A, EC, 96, B7),$$

$$w[6] = w[5] \oplus w[2] = (74, 85, F8, D2)$$

$$w[7] = w[6] \oplus w[3] = (54, D1, 8F, BD)$$

Dari proses di atas diperoleh  $rk[1]$  sebagai berikut:

75	3A	74	54
82	EC	85	D1
F0	96	F8	8F
97	B7	D2	BD

Selanjutnya dengan meneruskan algoritma ekspansi kunci untuk  $i = 8, 9, 10, 11$ , maka diperoleh  $rk[2] = (w[8], w[9], w[10], w[11])^T$ , untuk  $i = 12, 13, 14, 15$  diperoleh  $rk[3] = (w[12], w[13], w[14], w[15])^T$ , demikian seterusnya sampai diperoleh  $rk[10] = (w[40], w[41], w[42], w[43])^T$ .

### 2.7.2.6 Dekripsi

Algoritma dekripsi mirip dengan algoritma enkripsi, namun dengan beberapa perubahan pada beberapa konstanta dan tabel. Untuk membedakannya dengan algoritma enkripsi, maka untuk fungsi dengan cara yang berkebalikan maka nama-nama fungsinya diberi awalan *inv*, artinya inversi.

#### 2.7.2.7 InvSubBytes()

Operasi substitusi byte di dalam *InvSubBytes()*, sama seperti didalam *SubBytes()*, hanya saja S-box yang digunakan adalah inversi dari S-box terdahulu. Tabel inversi S-box yang digunakan selama dekripsi adalah seperti ditunjukkan pada gambar.

Table 2.12 InvShiftRows didalam AES

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	9	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	8	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	0	8C	BC	D3	0A	F7	E4	58	5	B8	B3	45	6
7	D0	2C	1E	8F	CA	3F	0F	2	C1	AF	BD	3	1	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73

9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	7	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	4	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

### 2.7.2.9 InvShiftRows()

Transformasi *InvShiftRows* sama seperti *ShiftRows* namun melakukan pergeseran dalam arah berlawanan (ke kanan) untuk tiap-tiap baris pada tiga baris terakhir di dalam *state* (Gambar)

Table 2.13 *InvShiftRows* di dalam AES

$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$		$a_{00}$	$a_{01}$	$a_{02}$	$a_{03}$
$a_{10}$	$a_{11}$	$a_{12}$	$a_{13}$	Geser 1	$a_{11}$	$a_{10}$	$a_{11}$	$a_{12}$
$a_{20}$	$a_{21}$	$a_{22}$	$a_{23}$	Geser 1	$a_{22}$	$a_{23}$	$a_{20}$	$a_{21}$
$a_{30}$	$a_{31}$	$a_{32}$	$a_{33}$	Geser 1	$a_{31}$	$a_{32}$	$a_{33}$	$a_{30}$

### 2.7.2.9 InvMixColumns()

Transformasi *InvMixColumn* didefinisikan sebagai perkalian matriks berikut:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S_{0,0} & S_{0,1} & S_{0,2} & S_{0,3} \\ S_{1,0} & S_{1,1} & S_{1,2} & S_{1,3} \\ S_{2,0} & S_{2,1} & S_{2,2} & S_{2,3} \\ S_{3,0} & S_{3,1} & S_{3,2} & S_{3,3} \end{bmatrix} = \begin{bmatrix} S'_{0,0} & S'_{0,1} & S'_{0,2} & S'_{0,3} \\ S'_{1,0} & S'_{1,1} & S'_{1,2} & S'_{1,3} \\ S'_{2,0} & S'_{2,1} & S'_{2,2} & S'_{2,3} \\ S'_{3,0} & S'_{3,1} & S'_{3,2} & S'_{3,3} \end{bmatrix}$$



## 2.8 PHP (*Hypertext Preprocessor*)

Adalah salah satu bahasa pemrograman skrip yang dirancang untuk membangun aplikasi web. Ketika dipanggil dari web browser, program yang ditulis dengan php akan di-parsing di dalam web server oleh interpreter PHP dan diterjemahkan ke dalam dokumen HTML, yang selanjutnya akan ditampilkan ke web browser.

Karena pemrosesan program PHP dilakukan di lingkungan web server, PHP dikatakan sebagai bahasa sisi server(server-side). Oleh sebab itu, seperti yang telah dikemukakan sebelumnya, kode PHP tidak akan terlihat pada saat user memilih perintah “View Source” pada web browser yang mereka gunakan.

Pada prinsipnya server akan bekerja apabila ada permintaan dari client. Dalam hal ini client menggunakan kode-kode PHP untuk mengirimkan permintaan ke server. Sistem kerja dari PHP diawali dengan permintaan yang berasal dari halaman website oleh browser. Berdasarkan URL atau alamat website dalam jaringan internet, browser akan menemukan sebuah alamat dari webserver, mengidentifikasi halaman yang dikehendaki, dan menyampaikan segala informasi yang dibutuhkan oleh webserver.

Selanjutnya webserver akan mencari berkas yang diminta dan menampilkan isinya di browser. Browser yang mendapatkan isinya segera menerjemahkan kode HTML dan menampilkannya. Lalu bagaimana apabila yang dipanggil oleh user adalah halaman yang mengandung script PHP? Pada prinsipnya sama dengan memanggil kode HTML, namun pada saat permintaan dikirim ke web-server, web-server akan memeriksa tipe file yang diminta user. Jika tipe file yang diminta adalah PHP, maka akan memeriksa isi script dari halaman PHP tersebut.

Apabila dalam file tersebut tidak mengandung script PHP, permintaan user akan langsung ditampilkan ke browser, namun jika dalam

file tersebut mengandung script PHP, maka proses akan dilanjutkan ke modul PHP sebagai mesin yang menerjemahkan script-script PHP dan mengolah script tersebut, sehingga dapat dikonversikan ke kode-kode HTML lalu ditampilkan ke browser user.

## 2.9 MySql

MySQL merupakan software Relational Database Management System (RDBMS) atau server database yang dapat mengelola database dengan sangat cepat, dapat menampung data dalam jumlah sangat besar, dapat diakses oleh banyak user (multi-user), dan dapat melakukan suatu proses secara sinkron atau bersamaan (multi-threaded).

Lisensi MySQL terbagi menjadi dua, yaitu dapat menggunakan MySQL sebagai produk opensource dibawah General Public License (GNU) secara gratis atau dapat membeli lisensi dari versi komersialnya. MySQL versi komersial tentu memiliki nilai lebih atau kemampuan yang tidak disertakan pada versi gratis. Pada kenyataannya, untuk keperluan industri menengah kebawah, versi gratis dapat digunakan dengan baik (Achmad Munandar,2019).

### 2.9.1 MySqli

MySQLi merupakan salah satu ekstensi PHP untuk mengakses fungsional yang disediakan MySQL 4.1 ke atas. Jika pada tulisan sebelumnya mengakses MySQL dengan menggunakan MySQL Extension, MySQL Improved Extension ditujukan agar dapat menggunakan fitur MySQL versi 4.1.3 ke atas, sedangkan ekstensi MySQL lama diperuntukkan untuk versi MySQL sebelumnya.

Ekstensi MySQL lama akan berstatus deprecated pada rilis PHP 5.5 dan selanjutnya akan dibuang, untuk itu disarankan menggunakan Ekstensi MySQLi atau PDO MySQL untuk menulis kode-kode PHP yang baru. Ekstensi MySQL hanya dapat

digunakan untuk pemeliharaan kode-kode lama yang telah dikembangkan (Harison, Ahmad Syarif, 2016)

## 2.10 Rapid Application Diagram (RAD)

RAD adalah suatu pendekatan berorientasi objek terhadap pengembangan sistem yang mencakup suatu metode pengembangan serta perangkat-perangkat lunak. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi. Pada akhirnya, RAD sama-sama berusaha memenuhi syarat-syarat bisnis yang berubah secara cepat.

RAD adalah proses model perangkat lunak inkremental yang menekankan siklus pengembangan yang singkat. Model RAD adalah sebuah adaptasi “kecepatan tinggi” dari model *waterfall*, di mana perkembangan pesat dicapai dengan menggunakan pendekatan konstruksi berbasis komponen. Jika tiap-tiap kebutuhan dan batasan ruang lingkup proyek telah diketahui dengan baik, proses RAD memungkinkan tim pengembang untuk menciptakan sebuah “sistem yang berfungsi penuh” dalam jangka waktu yang sangat singkat.

Satu perhatian khusus mengenai metodologi RAD dapat diketahui, yakni implementasi metode RAD akan berjalan maksimal jika pengembang aplikasi telah merumuskan kebutuhan dan ruang lingkup pengembangan aplikasi dengan baik. (Theresa Ayu, 2016)

### 2.10.1 Fase dan Tahap Pengembangan Aplikasi

Terdapat tiga fase dalam RAD yang melibatkan penganalisis dan pengguna dalam tahap penilaian, perancangan, dan penerapan. Adapun ketiga fase tersebut adalah requirement planning (Perencanaan syarat-syarat), RAD *design workshop* (*Workshop Desain RAD*) dan implementation (Implementasi). Sesuai dengan metodologi RAD menurut Kendall (2010), berikut ini adalah tahap-tahap pengembangan aplikasi dari tiap fase pengembangan aplikasi.



**Gambar 2.8** Tahap Pengembangan RAD

### 1. *Requirements Planning* (Perencanaan Syarat-Syarat).

Dalam fase ini, pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.

### 2. *RAD Design Workshop* (Workshop Desain RAD)

Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Apabila seorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini

dapat mendorong pengembangan sampai pada tingkat terakselerasi.

### 3. *Implementation* (Implementasi)

Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan nonteknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diuji coba dan kemudian diperkenalkan kepada organisasi.

#### 2.10.2 Kelebihan dan Kekurangan RAD

Metode pengembangan sistem RAD relatif lebih sesuai dengan rencana pengembangan aplikasi yang tidak memiliki ruang lingkup yang besar dan akan dikembangkan oleh tim yang kecil. Namun, RAD pun memiliki kelebihan dan kekurangannya sebagai sebuah metodologi pengembangan aplikasi. Berikut ini adalah kelebihan metodologi RAD:

1. Penghematan waktu dalam keseluruhan fase proyek dapat dicapai
2. RAD mengurangi seluruh kebutuhan yang berkaitan dengan biaya proyek dan sumberdaya manusia.
3. AD sangat membantu pengembangan aplikasi yang berfokus pada waktu penyelesaian proyek.
4. Perubahan desain sistem dapat lebih berpengaruh dengan cepat dibandingkan dengan pendekatan SDLC tradisional.
5. Sudut pandang user disajikan dalam sistem akhir baik melalui fungsi-fungsi sistem atau antarmuka pengguna
6. RAD menciptakan rasa kepemilikan yang kuat di antara seluruh pemangku kebijakan proyek.



Sedangkan, mengacu pada pendapat Kendall (2010), maka dapat diketahui bahwa kekurangan penerapan metode RAD adalah sebagai berikut:

1. Dengan metode RAD, penganalisis berusaha mepercepat projek dengan terburu-buru.
2. Kelemahan yang berkaitan dengan waktu dan perhatian terhadap detail. Aplikasi dapat diselesaikan secara lebih cepat, tetapi tidak mampu mengarahkan penekanan terhadap permasalahan-permasalahan perusahaan yang seharusnya diarahkan.
3. RAD menyulitkan *programmer* yang tidak berpengalaman menggunakan prangkat ini dimana *programmer* dan *analyst* dituntut untuk menguasai kemampuan-kemampuan baru sementara pada saat yang sama mereka harus bekerja mengembangkan sistem.

### 2.11 Pengujian *Blackbox*

Pengujian *Blackbox* digunakan untuk menguji fungsi-fungsi perangkat lunak yang dirancang. Pengujian *Blackbox* berfokus pada persyaratan fungsional perangkat lunak. Dengan demikian, pengujian *Blackbox* memungkinkan perekayasa perangkat lunak mendapatkan serangkaian kondisi masukan yang sepenuhnya menggunakan semua persyaratan fungsional untuk suatu program. Pengujian *Blackbox* berusaha menemukan kesalahan dalam kategori sebagai berikut (Theresa Ayu, 2016):

1. Fungsi-fungsi yang tidak benar atau hilang.
2. Kesalahan antarmuka.
3. Kesalahan dalam struktur data atau akses basis data eksternal.
4. Kesalahan kinerja.
5. Inisialisasi dan kesalahan terminasi.



### 2.11.1 Proses dalam Blackbox Testing

Terdapat beberapa proses di dalam pengujian *blackbox testing* yaitu sebagai berikut:

1. Menganalisa kebutuhan dan spesifikasi dari perangkat lunak.
2. Pemilihan jenis *input* yang mungkin menghasilkan *output* yang benar.
3. Pengujian dilakukan dengan *input-input* yang benar-benar telah diseleksi.
4. Perbandingan *output* yang dihasilkan dengan *output* yang diterapkan.
5. Menentukan fungsionalitas yang harusnya ada pada perangkat lunak yang diuji.

Berikut ini merupakan tabel perbandingan antara metode pengujian *Blackbox Testing* dan *Whitebox Testing*.

**Tabel 2. 14** Perbandingan *Blackbox Testing* dan *Whitebox Testing* (Pressman, 2009)

Metode Pengujian	Kelebihan	Kekurangan
<i>Blackbox Testing</i>	<ul style="list-style-type: none"> <li>- <i>Software tester</i> dalam jumlah yang banyak dapat menguji program tersebut tanpa harus memiliki pengetahuan tentang programming.</li> <li>- Cocok untuk <i>source code</i> dengan skala besar.</li> <li>- Menguji program dari sudut pandang user.</li> </ul>	<ul style="list-style-type: none"> <li>- Pengujian tidak spesifik karena <i>software tester</i> tidak memiliki akses ke <i>source code</i>.</li> <li>- Pengujian tidak efisien karena <i>software tester</i> memiliki pengetahuan yang terbatas tentang program.</li> <li>- <i>Software tester</i> hanya menjalankan beberapa</li> </ul>

		skenario pengujian yang dipilih.
<i>Whitebox Testing</i>	<ul style="list-style-type: none"> <li>-Sebagai <i>software engineer</i> yang memiliki akses ke <i>source code</i>, hal ini menjadi sangat mudah untuk melakukan skenario pengujian secara efektif.</li> <li>- Baris kode yang tidak efisien dapat dihilangkan agar mencegah <i>bugs</i> pada program.</li> </ul>	<ul style="list-style-type: none"> <li>- Karena dibutuhkan <i>softwareengineer</i> yang berpengalaman dalam <i>Whitebox Testing</i> sehingga mengeluarkan biaya tambahan.</li> <li>- Terkadang sangat sulit melihat setiap baris kode untuk mencari <i>bugs</i> pada program yang akan diuji.</li> </ul>

Dari perbandingan metode pengujian perangkat lunak di atas, penulis memilih metode pengujian *Blackbox Testing* karena dengan *Blackbox Testing* pengguna tidak harus mengetahui tentang bahasa pemrograman, tetapi *user* hanya melihat *output* sistem sudah sesuai dengan apa yang di masukkan (*input*).

## **BAB III**

### **METODOLOGI PENELITIAN**

#### **3.1 Metode Pengumpulan Data**

Dalam penelitian ini, terdapat tiga tahap dalam melakukan pengumpulan data untuk menggali informasi yang berguna bagi penelitian ini. Tahap pengumpulan data tersebut meliputi studi lapangan, studi pustaka dan studi literature.

##### **3.1.1 Studi Lapangan**

###### **Observasi**

Pengumpulan data secara observasi dilakukan dengan melihat langsung proses dan kegiatan yang berjalan yang dilakukan di Bank Sampah Malaka Sari dalam memberikan pelayanan kepada nasabah untuk proses transaksi. Kegiatan pengamatan langsung ini dilakukan di bawah pengawasan pak Prakoso selaku pengurus Bank Sampah Malaka Sari

###### **Wawancara**

Metode Wawancara ini dilakukan dengan cara melakukan diskusi dengan beberapa narasumber yang sekaligus mengawasi penulis saat melakukan riset yaitu pak Prakoso. Wawancara dilakukan dengan mengajukan pertanyaan mengenai alur dari proses transaksi di bank sampah malaka sari, dari mulai Penimbangan sampah sampai dengan pencatatan dibuku tabungan nasabah. Selain itu peneliti mengusulkan kepada pihak bank sampah untuk dibuat sebuah sistem aplikasi yang dapat melakukan transaksi pada nasabah dan menyimpan data nasabah, serta pendataan jumlah sampah yang masuk.

Dari wawancara yang dilakukan dapat diketahui sistem yang sedang berjalan saat ini di bank sampah, masalah-masalah

yang mereka hadapi serta kebijakan-kebijakan yang ada di bank sampah.

### 3.1.2 Studi Pustaka

Metode ini dilakukan dengan mencari sumber referensi yang relevan mengenai topik yang akan dibahas, seperti buku teori, jurnal ilmiah, dan artikel atau tulisan dari internet.

### 3.1.3 Studi Literature

Metode ini dilakukan dengan menelusuri literatur terhadap beberapa jurnal yang berkaitan dengan Advanced Encryption standart (AES). dari jurnal-jurnal tersebut diambil beberapa kesimpulan seperti algoritma yang digunakan, kelebihan dan kekurangan setiap penelitian, dsb. Berikut adalah tabel studi literatur dari beberapa jurnal penelitian tersebut.

**Tabel 3. 1** Perbandingan *Studi Literatur* Sejenis

No	Judul	Objek (input)	Metode Proses	Hasil Penelitian
1	PENGAMANAN DATA MySQL PADA E-COMMERCE DENGAN ALGORITMA AES 256 (Kartika Imam Santoso, Wahyu Priyoatmoko, 2016)	Mengenkrip Data Customer E-commerce Input : Username, Password, Nama, Email, No.Hp, Alamat, Kota.	Proses enkripsi (penyandian data) pada data customer e-commerce menggunakan algoritma AES, yang nanti hasil enkripsinya akan disimpan di dalam database MySql	Enkripsi dengan AES 256 pada data MySQL E-Commerce yang menghasilkan ciphertext yang tidak sama meskipun kuncinya sama (statis).
2	ANALISIS KRIPTOGRAFI SIMETRIS AES DAN KRIPTOGRAFI ASIMETRIS RSA PADA ENKRIPSI CITRA	Enkripsi dan dekripsi data berupa <i>plain image</i> dengan membandingkan kecepatan dan kualitas enkripsi dengan 2 algoritma AES dan RSA	Pada proses enkripsi untuk algoritma asimetris RSA dibutuhkan masukan untuk kunci publik dan kunci privat, sedangkan untuk proses enkripsi AES	Proses enkripsi dan dekripsi pada algoritma RSA dipengaruhi oleh pasangan kunci. Dimana, semakin besar kunci publik maka akan semakin lama proses enkripsi, dan

	DIGITAL (Geby Geta Putri, Wiwin Styorini, Rizki Dian Rahayani, 2018)		memilih jenis AES yang akan digunakan	semakin besar kunci privat maka akan semakin lama proses dekripsi sedangkan, Proses enkripsi dan dekripsi pada algoritma AES dipengaruhi oleh panjang kunci.  Dimana semakin panjang kunci yang digunakan maka akan semakin banyak putaran yang dilalui dan semakin lama proses enkripsi dan dekripsi berlangsung.
3	IMPLEMENTASI ALGORITMA KRIPTOGRAFI RIJNDAEL (AES) UNTUK PENGAMANAN SISTEM SMS BANKING DAN INTERNET BANKING (I Wayan Ordiasa, 2015)	kode transaksi yang berupa <i>plain text</i> nantinya akan di enkripsi sebelum dikirim ke server, lalu proses dekripsinya dikirim oleh server	Input kode transaksi yang di enkripsi menggunakan algoritma Rijindael (AES), jika permintaan berhasil proses dekripsi akan dikirim ke client dari server, yang hasilnya berupa plain text sesuai dengan kode transaksi	algoritma AES berhasil diimplementasikan untuk kegiatan perbankan karena berkaitan kerahasiaan dan keamanan data nasabah
4	PERANCANGAN APLIKASI KRIPTOGRAFI FILE DENGAN METODE ALGORITMA ADVANCED ENCRYPTION STANDARD (AES). (Rahmat Tullah, Muhammad Iqbal	Masukan data berupa file/plain text lalu user memasukan input key sebagai password untuk melakukan dekripsi	Proses upload file yang akan di enkripsi dan disimpan di dalam database, lalu untuk proses dekripsinya dibutuhkan kunci yang sama saat melakukan proses enkripsi file tersebut	Tehnik dalam mengamankan sebuah file berupa excel, word dan pdf dapat dilakukan dengan menggunakan sebuah metode algoritma kriptografi.

	Dzulhaq, Yudi Setiawan, 2016)			
5	IMPLEMENTASI ALGORITME ADVANCE ENCRYPTION STANDARD (AES) PADA ENKRIPSI DAN DEKRIPSI QR-CODE (Dwi Qunita Putri Ambeq Paramarta, Ari Kusyanti, Mahendra Data, 2018)	User menginput plain text yang kemudian akan di proses menjadi Qr-Code, lalu Qr-code dan plain text di enkripsi menggunakan algoritma AES	Input user yang berupa plain text akan di ubah menjadi Qr-Code, lalu dua hasil tersebut kemudian di enkripsi menjadi cipher text dan Qr-Code yang sudah terenkripsi, untuk proses dekripsinya dibutuhkan key yang sama saat melakukan proses enkripsi. lalu Qr-Code yang sudah terenkripsi diproses menjadi plaintext	Algoritme AES dapat diterapkan pada QRCode. Algoritme AES akan memberikan aspek confidentiality, hal ini dapat dibuktikan dengan pengujian keamanan. Algoritme AES pada proses enkripsi menghasilkan output berupa ciphertext berupa karakter tidak jelas yang sulit dipahami dan pada proses dekripsi dilakukan dengan menscan ciphertext QRCode dan memasukkan key lalu sistem akan menjalankan proses dekripsi AES dan menghasilkan isi pesan asli atau plaintext

Berdasarkan hasil perbandingan beberapa literatur diatas dapat disimpulkan bahwa pada penelitian yang akan saya lakukan terdapat kelebihan dari penelitian-penelitian sebelumnya yang didapatkan dari referensi literatur diatas, yaitu penelitian ini menggunakan algoritma AES. Penelitian ini akan menggunakan algoritma untuk fungsi enkripsi pada transaksi..



### 3.2 Metode Pengembangan Sistem

Pada pengembangan sistem ini penulis menggunakan metode *Rapid Application Development (RAD)* dalam menganalisis, merancang (*design*) dan mengimplementasikan Sistem yang akan dibuat. Penulis menggunakan metode pengembangan *RAD* karena dengan *RAD* memungkinkan pengguna untuk aktif dan berpartisipasi dalam pengembangan sistem sehingga dapat memenuhi secara langsung permintaan *user*. Adapun tahapan yang dilakukan peneliti yaitu:

#### 1. *Requirement Planning (Perencanaan Syarat-Syarat)*

Dalam tahap ini penulis melakukan langkah-langkah sebagai berikut:

- a. Analisis masalah, yaitu melakukan analisis terhadap masalah yang terjadi pada sistem yang sedang berjalan pada Bank Sampah Malaka Sari.
- b. Gambaran umum organisasi, berisikan tentang instansi dan struktur organisasi dari Bank Sampah Malaka Sari
- c. Sistem yang sedang berjalan, berisikan tentang tampilan dari system yang sedang berjalan dan akan dikembangkan oleh penulis pada Bank Sampah Malaka Sari.
- d. Sistem usulan yang diusulkan untuk memperbaiki sistem yang lama.

#### 2. *RAD Design Workshop (Workshop Desain RAD)*

Dalam tahap ini penulis melakukan langkah-langkah sebagai berikut:

- a. Melakukan perancangan input, meliputi teks yang di-input sebagai plainteks yang akan di enkripsi ke dalam *database*.
- b. Merancang spesifikasi proses, menerjemahkan dalam algoritma dan mengimplementasikan dalam bentuk program

c. Merancang *Unified Modeling Language* (UML), yang terdiri dari :

1. Membuat *use case* diagram, bertujuan untuk mendeskripsikan usecase yang telah dibuat pada tahap pertama.
  2. Membuat *activity* diagram, bertujuan untuk membuat alur kerja dari satu aktivitas ke aktivitas lainnya.
  3. Membuat *sequence* diagram, bertujuan untuk menjelaskan interaksi objek yang disusun dalam suatu urutan waktu.
  4. Membuat *class* diagram, bertujuan untuk memperlihatkan himpunan kelas, *interface*, kolaborasi dan relasi.
- d. Merancang *database*, menentukan jumlah tabel, *coloumn*, *type coloumn* dan hubungan antar tabel.
- e. Merancang *interface* atau tampilan untuk mempermudah pengguna.
- f. Melakukan pengkodean program.

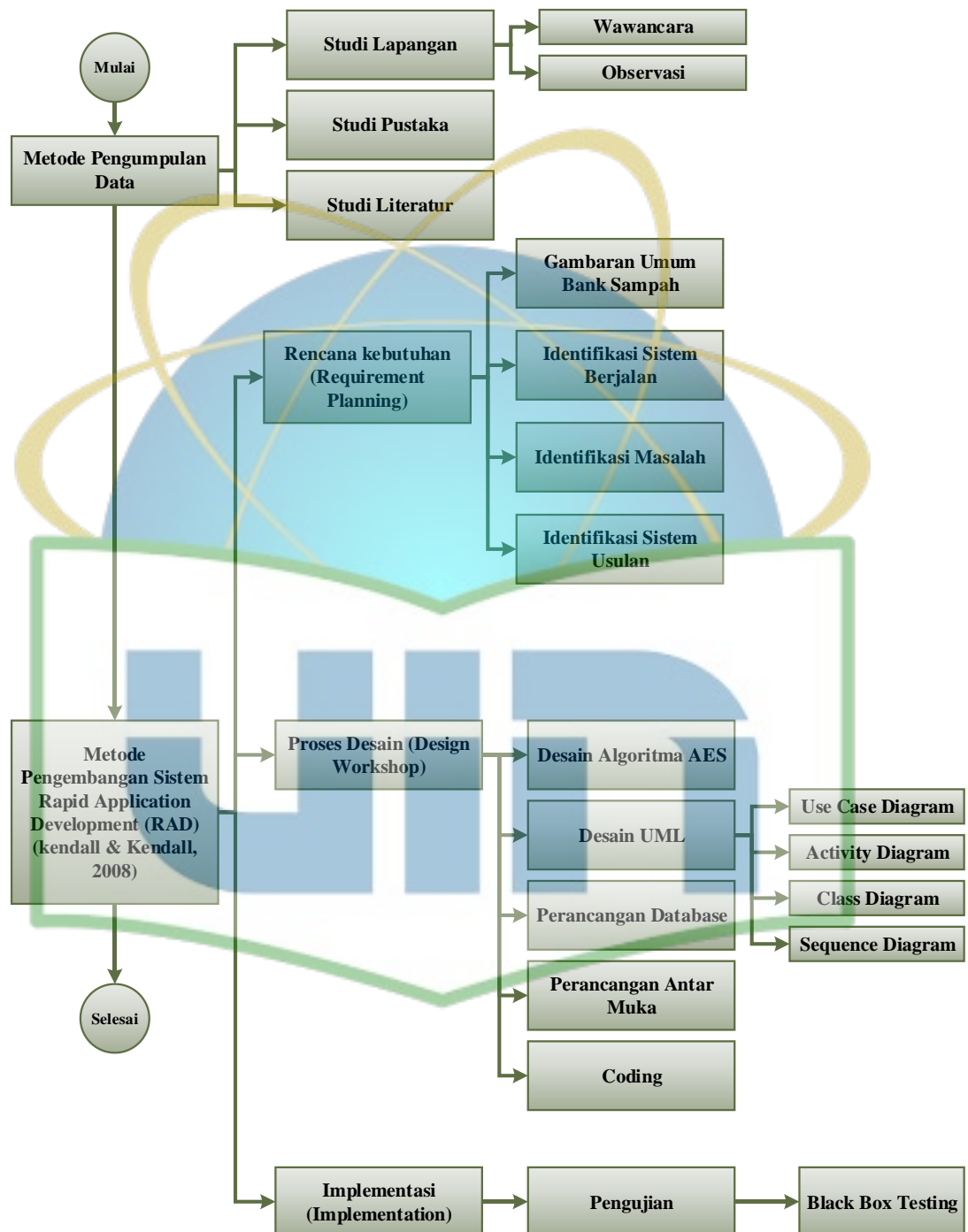
### **3. *Implementation* (Implementasi)**

Tahap ini merupakan tahap presentasi dari hasil perancangan ke dalam program. Dalam pembuatan Aplikasi Bank Sampah Malaka Sari, fase inplementasi terdiri dari tiga tahap, yaitu :

- a. Implementasi Perangkat Lunak
- b. Implementasi Perangkat Keras
- c. Implementasi Interface

### 3.3 Kerangka Berfikir Penelitian

Dalam tahap ini penulis akan menggambarkan sebuah kerangka berfikir penelitian sebagai berikut:



Gambar 3. 1 Kerangka Berfikir Penelitian

## BAB IV

### ANALISA Dan PEMBAHASAN

#### 4.1 *Requirements Planning* (Perencanaan Syarat-Syarat)

##### 4.1.1 Analisis Masalah

Bank Sampah Malaka Sari merupakan bank sampah yang berlokasi di Jakarta Timur yang sudah mendapatkan penghargaan *Gold* dari Pemprov DKI Jakarta melalui program “Jakarta Green and Clean” dengan jumlah nasabah yang melebihi 300 orang dan sampah yang terserap setiap bulannya mencapai 2-2.5 Ton. Setiap harinya Bank Sampah Malaka Sari melayani dan mencatat seluruh kegiatan transaksi bank sampah mulai dari pendaftaran nasabah baru, pendataan sampah yang masuk, kemudian pencatatan saldo di buku nasabah, hingga penarikan dana oleh nasabah. Akan tetapi semua proses implementasinya masih secara manual, sehingga terjadinya kesalahan (*human error*) dinilai masih cukup tinggi, kurangnya integritas data antara bank sampah dan nasabah akan mempengaruhi nasabah dalam berperan dalam kegiatan bank sampah, pendataan sampah yang masuk sering tidak valid terhadap jumlah sampah yang terserap.

Peneliti mengusulkan untuk meningkatkan efektifitas dan efisiensi dari kegiatan bank sampah perlu dibuat sistem yang dapat memonitoring seluruh kegiatan bank sampah dan melengkapi sistem tersebut dengan keamanan data yang baik, untuk menghindari adanya kejahatan yang dilakukan oleh pihak ketiga, seperti manipulasi data dan pencurian data, demi mendukung pengamanan aplikasi tersebut maka perlu diterapkan sebuah algoritma yang memiliki teknik pengamanan yang cukup baik. Algoritma AES menjadi solusi sebagai metode enkripsi dalam

melakukan pengamanan data yang hasil enkripsinya akan tersimpan di dalam database MySql.

#### **4.1.2 Tinjauan Bank Sampah**

Tinjauan terhadap organisasi menjelaskan tentang visi dan misi organisasi, tujuan dibentuknya organisasi dan struktur organisasi.

##### **4.1.2.1 Visi, Misi dan Tujuan Bank Sampah**

###### **Visi**

bank sampah sebagai wadah untuk mewujudkan masyarakat yang peduli terhadap lingkungan.

###### **Misi**

1. Mengajak masyarakat untuk peduli terhadap lingkungan.
2. Memberikan pendidikan terhadap masyarakat agar sadar tentang pentingnya menjaga lingkungan dan kesehatan.
3. Memberdayakan masyarakat dengan memanfaatkan sampah.

###### **Tujuan**

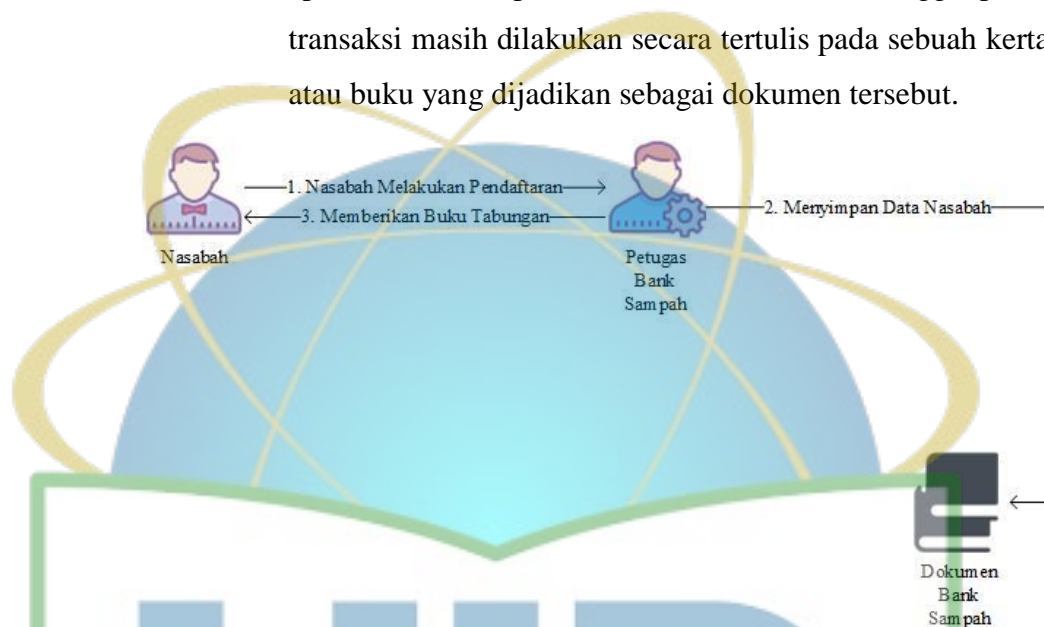
Tujuan utama pendirian bank sampah mekar sari adalah untuk membantu menangani pengolahan sampah di Jakarta. Tujuan bank sampah selanjutnya adalah untuk menyadarkan masyarakat akan lingkungan yang sehat, rapi, dan bersih. Bank sampah juga didirikan untuk mengubah sampah menjadi sesuatu yang lebih berguna dalam masyarakat, misalnya untuk kerajinan dan pupuk yang memiliki nilai ekonomis.

#### **4.1.3 Identifikasi Sistem Berjalan**

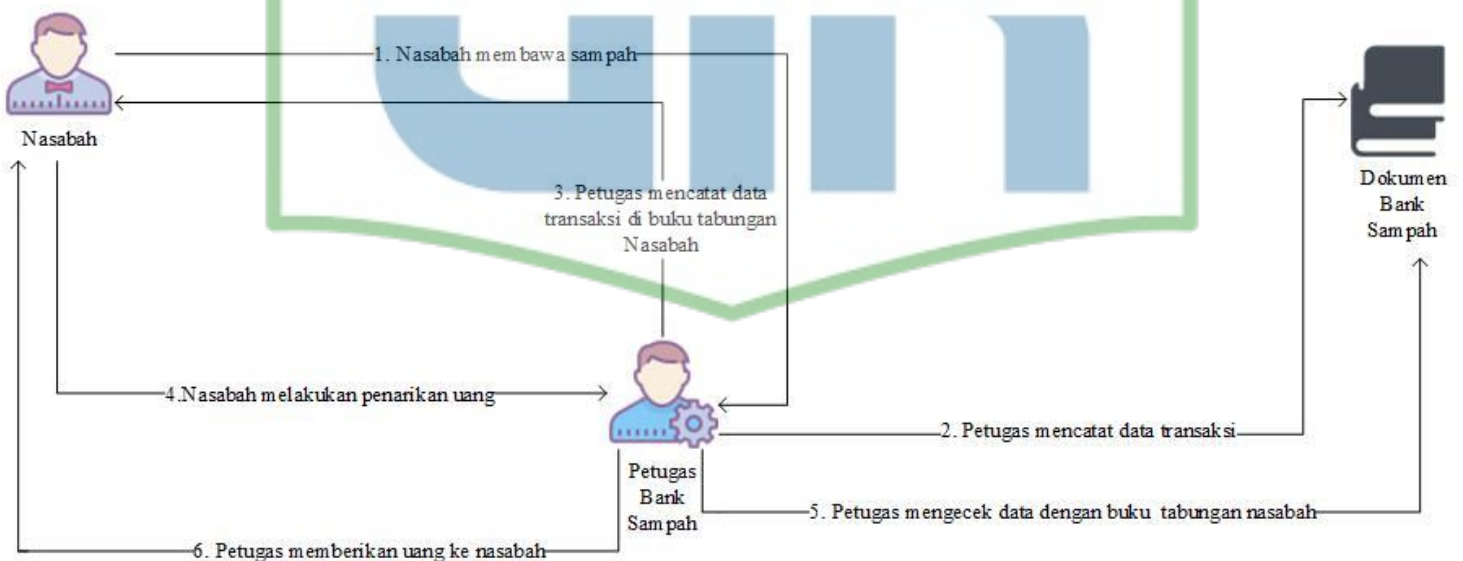
Proses sistem berjalan yang ada pada Bank Sampah Malaka Sari adalah sebagai berikut:

### 1. Prosedur Pengelolaan Data

Proses pengelolaan data seperti, data nasabah, data transaksi, data saldo bank sampah dan data sampah yang ditampung oleh bank sampah, masih dilakukan secara manual yaitu dengan pencatatan tertulis tanpa sebuah aplikasi. Proses pendaftaran nasabah baru hingga proses transaksi masih dilakukan secara tertulis pada sebuah kertas atau buku yang dijadikan sebagai dokumen tersebut.



2. **Gambar 4.1** Proses Sistem Berjalan Alur Pendaftaran



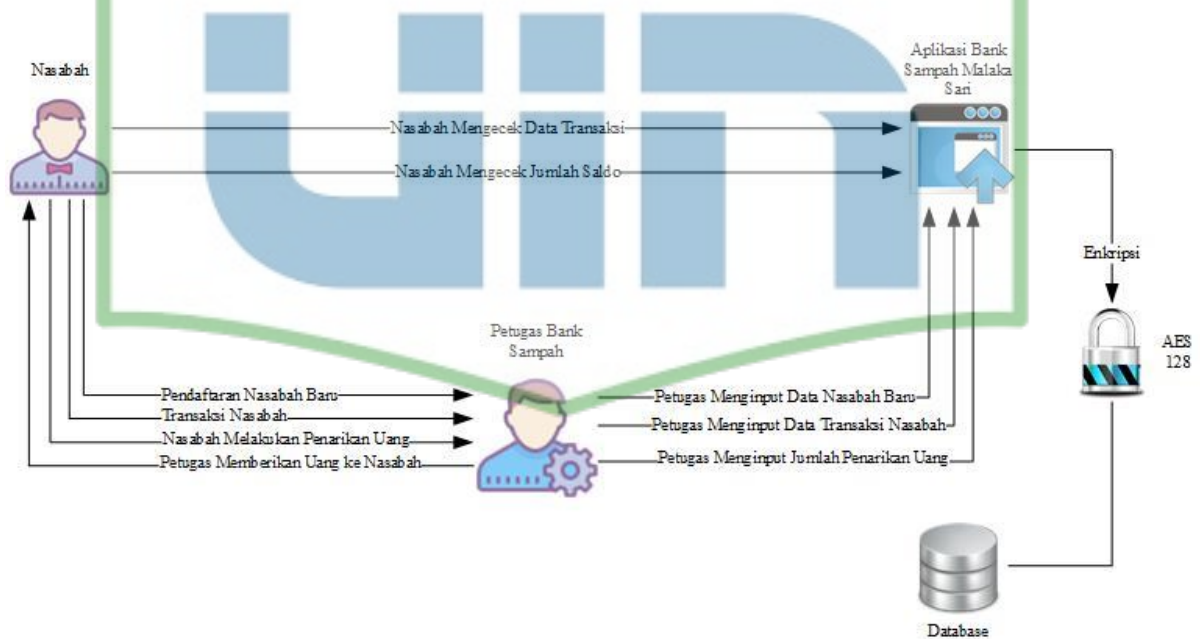
**Gambar 4.2** Proses Sistem Berjalan Bank Sampah



Proses penyimpanan data yang dilakukan di bank sampah malaka sari masih dengan cara penyimpanan berkas berupa dokumen / buku yang masih berantakan disebabkan media penyimpanan yang digunakan hanya sebuah lemari penyimpanan dokumen dan belum memiliki media penyimpanan data berupa aplikasi yang jauh lebih aman.

#### 4.1.4 Identifikasi Sistem Usulan

Dengan melihat segala permasalahan dan berdasarkan hasil analisa diatas, maka peneliti mengusulkan sebuah solusi permasalahan terhadap kelemahan dan kendala yang dihadapi tersebut. Usulan sistem yang akan dibuat oleh penulis yaitu, dengan memfasilitasi bank sampah malaka sari dengan komputerisasi. Dalam hal ini penulis akan membangun sebuah program aplikasi bank sampah yang akan dilengkapi dengan teknik pengamanan, yang diharapkan dapat meningkatkan dari segi mutu, kinerja dan pelayanan kepada nasabah.



**Gambar 4.3** Proses Sistem Berjalan Bank Sampah

## 4.2 Workshop Design

### 4.2.1 Perancangan Algoritma AES (Advanced Encrytion System)

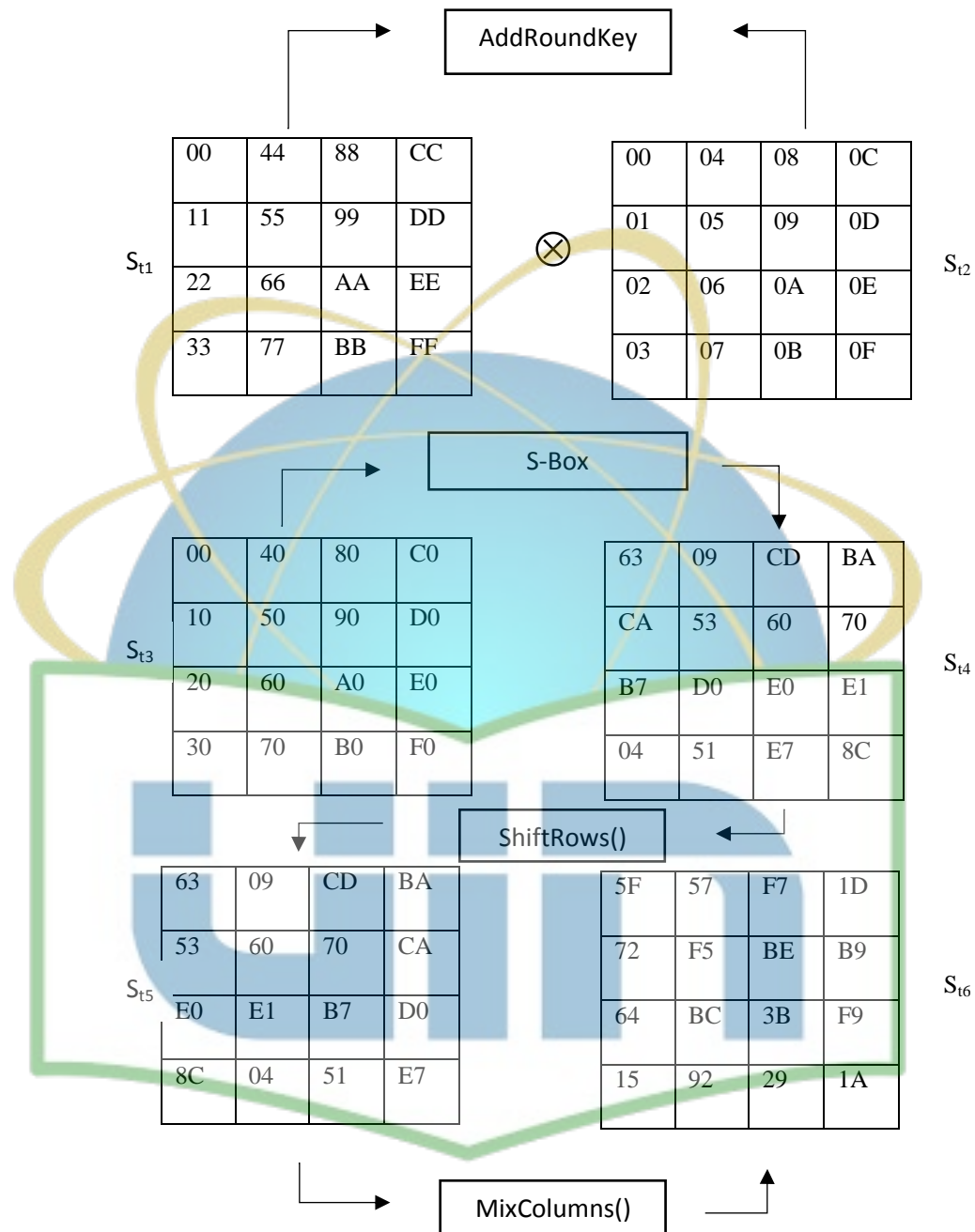
Penerapan algoritma AES (*Advanced Encrytion System*) pada aplikasi adalah untuk melindungi data nasabah dan bank sampah dengan cara melakukan enkripsi pada data transaksi tersebut. Pada aplikasi bank sampah terdapat fitur transaksi yang digunakan untuk pengamanan data di record database, proses enkripsi dan dekripsi diterapkan saat melakukan proses transaksi, sistem akan secara otomatis mengenkripsi data transaksi ke dalam database, setelah masuk dan tersimpan kedalam database data transaksi tersebut akan berubah menjadi data teks acak (ciphertext) yang sulit diartikan ke teks biasa.

Ketika data transaksi yang ada pada database akan dipanggil dan ditampilkan kedalam aplikasi kembali, maka fungsi dekripsi akan terpanggil sehingga akan merubah menjadi tesk biasa seperti pada proses melakukan fitur transaksi pada sistem dan pengguna aplikasi akan dapat membaca informasi transaksi dan detail transaksi tanpa mengetahui proses enkripsi dan dekripsi tersebut sedang berjalan.

Pada contoh kali ini peneliti akan menerapkan proses enkripsi dengan metode AES yang di implementasikan pada system ini:

Plain text: 00 11 22 33 44 55 66 77 88 99 AA BB CC DD EE FF

Key: 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F



Langkah pertama yaitu mengcopy plaintext sebagai  $S_{t1}$  dan kunci sebagai  $S_{t2}$ .  $S_{t3}$  didapat dari proses *AddRoundKey* antara  $S_{t1}$  dan  $S_{t2}$  yang dikonversikan kedalam bentuk biner, sehingga dihasilkan:

{ '00' ; '10' ; '20' ; '30' ; '40' ; '50' ; '60' ; '70' ; '80' ; '90' ; 'A0' ; 'B0' ; 'C0' ; 'D0' ; 'E0' ; 'F0' }

Langkah selanjutnya SubBytes() yaitu mensubstitusikan  $S_{t3}$  dalam bentuk heksadesimal kedalam table S-Box sehingga menghasilkan  $S_{t4}$ . Dimana diketahui  $S_{r,c}$  sebagai state 3 serta r (row) merupakan baris dan c (column).  $S_{t5}$  merupakan hasil dari proses ShiftRows dengan menggeser secara cyclic sebagai berikut:

$S_{t4}$	63	09	CD	BA
	CA	53	60	70
	B7	D0	E0	E1
	04	51	E7	8C

→

$S_{t5}$	63	09	CD	BA
	53	60	70	CA
	E0	E1	B7	D0
	8C	04	51	E7

Langkah selanjutnya MixColumns,  $S_{t6}$  dihasilkan dari perkalian antara koefisien { '02' ; '03' ; '01' ; '01' } yang ditetapkan AES dengan  $S_{t5}$  (per - word) operasi yang dilakukan sebagai perkalian matriks dengan mempresentasikan ke dalam bentuk polinomial sehingga mendapatkan persamaan, sebagai berikut:

$$W_0 = 6353E08C$$

$$W_1 = 0960E104$$

$$W_3 = CD70B751$$

$$W_4 = BACAD0E7$$

Sebagai contoh  $W_0 = 6353E08C$

$$S'_{0,c} = ([02] \bullet 63) \otimes ([03] \bullet 53) \otimes E0 \otimes 8C$$

$$S'_{1,c} = 63 \otimes ([02] \bullet 53) \otimes ([03] \bullet E0) \otimes 8C$$

$$S'_{2,c} = 63 \otimes 53 ([02] \bullet E0) \otimes ([03] \bullet 8C)$$

$$S'_{3,c} = ([03] \bullet 63) \otimes 53 \otimes 63 \otimes ([02] \bullet 8C)$$

$$1. S'_{0,c} = ([02] \bullet 63) \otimes ([03] \bullet 53) \otimes E0 \otimes 8C$$

$$\begin{aligned} '02 \bullet 63' &= (x) \cdot (x^6 + x^5 + x + 1) = x^7 + x^6 + x^2 + x = 1100 \\ &0110 \end{aligned}$$

$$\begin{aligned} '03 \bullet 53' &= (x + 1) \cdot (x^6 + x^4 + x + 1) \\ &= (x^7 + x^5 + x^2 + x) + (x^6 + x^5 + x + 1) \\ &= x^7 + x^6 + x^5 + x^4 + x^2 + x + 1 = 1111 \ 0101 \end{aligned}$$

$$'01 \bullet E0' = (1) \cdot (x^7 + x^6 + x^5) = x^7 + x^6 + x^5 = 1110 \ 0000$$

$$'01 \bullet 8C' = (1) \cdot (x^7 + x^3 + x^2) = x^7 + x^3 + x^2 = 1000 \ 1100$$

## 2. $63 \otimes ([02] \bullet 53) \otimes ([03] \bullet E0) \otimes 8C$

$$'01 \bullet 63' = (1) \cdot (x^7 + x^6 + x^5 + 1) = x^7 + x^6 + x^5 + 1 = 0110 \ 0011$$

$$'02 \bullet 53' = (x) \cdot (x^6 + x^4 + x + 1) = x^7 + x^5 + x^2 + x = 1010 \ 0110$$

$$'03 \bullet 53' = (x + 1) \cdot (x^6 + x^4 + x^5) = (x^8 + x^7 + x^6) + (x^7 + x^6 + x^5)$$

$$= (x^8 + x^5) \text{ modulo } (x^8 + x^4 + x^3 + x + 1)$$

$$= x^5 + x^4 + x^3 + x + 1$$

$$= 0011 \ 1011$$

$$'01 \bullet 8C = (1) \cdot (x^7 + x^3 + x^2) = x^7 + x^3 + x^2 = 1000 \ 1100$$

## 3. $S'_{2,c} = 63 \otimes 53 ([02] \bullet E0) \otimes ([03] \bullet 8C)$

$$'01 \bullet 63' = (1) \cdot (x^6 + x^5 + x + 1) = x^6 + x^5 + x + 1 = 0110 \ 0011$$

$$'01 \bullet 53' = (1) \cdot (x^6 + x^4 + x + 1) = x^6 + x^4 + x + x = 0101 \ 0011$$

$$\begin{aligned} '02 \bullet E0' &= (x) \cdot (x^7 + x^6 + x^5) = x^8 + x^7 + x^6 \\ &= (x^8 + x^7 + x^6) \text{ modulo } (x^8 + x^4 + x^3 + x + 1) \\ &= x^7 + x^6 + x^4 + x^3 + x + 1 \\ &= 1101 \ 1011 \end{aligned}$$

$$\begin{aligned} '03 \bullet 8C' &= (x + 1) \cdot (x^7 + x^3 + x^2) = (x^8 + x^4 + x^3) + (x^7 + x^3 + x^2) \\ &= (x^8 + x^7 + x^4 + x^2) \text{ modulo } (x^8 + x^4 + x^3 + x + 1) \\ &= x^7 + x^3 + x^2 + x + 1 \end{aligned}$$

$$= 1000\ 1111$$

$$4. ([03] \bullet 63) \otimes 53 \otimes 63 \otimes ([02] \bullet 8C)$$

$$\begin{aligned} '03 \bullet 63' &= (x + 1) \cdot (x^6 + x^5 + x + 1) \\ &= (x^7 + x^6 + x^2 + x) + (x^6 + x^5 + x + 1) = x^7 + x^5 + x^2 + 1 \\ &= 1010\ 0101 \end{aligned}$$

$$'01 \bullet 53' = (1) \cdot (x^6 + x^4 + x + 1) = x^6 + x^4 + x + 1 = 0101\ 0011$$

$$'01 \bullet E0' = (1) \cdot (x^7 + x^6 + x^5) = x^7 + x^6 + x^5 = 1110\ 0000$$

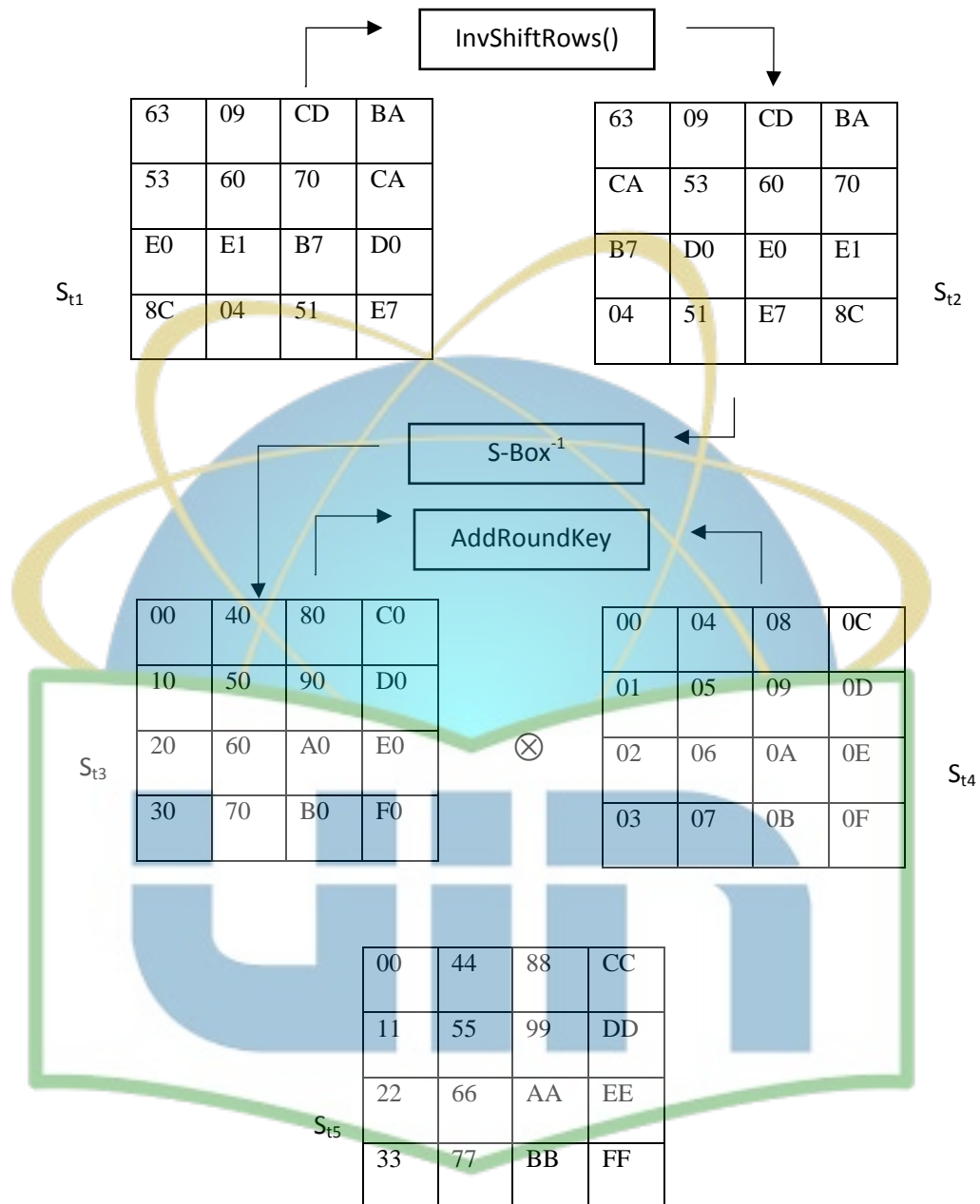
$$\begin{aligned} '02 \bullet 8C' &= (x) \cdot (x^7 + x^3 + x^2) = (x^8 + x^4 + x^3) \\ &= (x^8 + x^4 + x^3) \text{ modulo } (x^8 + x^4 + x^3 + x + 1) \\ &= x + 1 \\ &= 1000\ 1111 \end{aligned}$$

lalu dijumlah dengan metode XOR  $\otimes$ , dapat dicontohkan sebagai berikut:

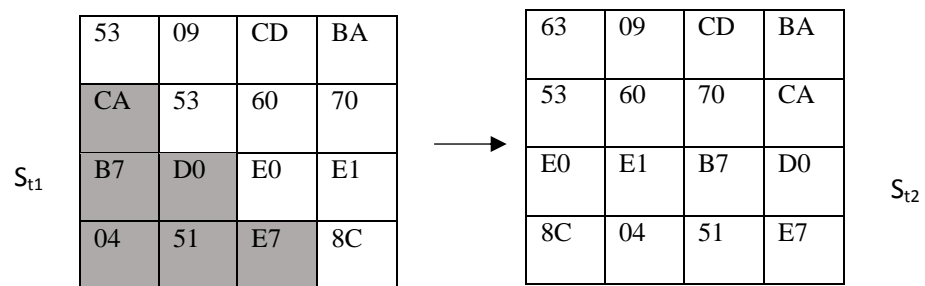
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} 63 \\ 53 \\ E0 \\ 8C \end{bmatrix} = \begin{bmatrix} 5F \\ 72 \\ 64 \\ 15 \end{bmatrix}$$

Proses dekripsi merupakan penterjemahan ciphertext menjadi bentuk semula atau plain text. Berikut ini peneliti akan mensimulasikan pada round ke – 10 yang merupakan invers dari cipher yang mana proses MixColumns tidak diikutsertakan pada round ini. Dijelaskan sebagai berikut:





Invers ShiftRows ini dengan menggeser secara cyclic  $S_{t1}$  menjadi  $S_{t2}$  sebagai berikut:



Langkah selanjutnya invers SubBytes yaitu mensubstitusikan  $S_{t2}$  dalam bentuk heksadesimal ke dalam table S-Box<sup>-1</sup> sehingga menghasilkan  $S_{t3}$ .

Langkah terakhir yaitu AddRoundKey dengan mengoperasikan XOR antara  $S_{t3}$  dan  $S_{t4}$ . Sehingga dihasilkan  $S_{t5}$  sebagai plain text, sehingga kembali ke bentuk aslinya atau plain text:

{'00'; '11'; '22'; '33'; '44'; '55'; '66'; '77'; '88'; '99'; 'AA'  
; 'BB'; 'CC'; 'DD'; 'EE'; 'FF' }

#### 4.2.2 Perancangan Model UML

Sebagaimana yang telah dijelaskan pada bab sebelumnya perancangan pemodelan objek mengguakan UML untuk merancang dan pengembangan aplikasi. Selanjutnya akan dijelaskan UML apa saja yang digunakan dalam penelitian ini.

##### 4.2.2.1 Use Case Diagram

*Use case diagram* mendeskripsikan interaksi aktor didalam Sistem. Hal ini diperlukan agar sistem dapat digunakan sesuai kebutuhan. Adapun langkah-langkah dalam membuat *usecase diagram* adalah sebagai berikut:

##### Identifikasi Aktor

Pengidentifikasian terhadap aktor diperlukan agar sistem dapat digunakan sesuai kebutuhan. Berikut ini identifikasi aktor pada Aplikasi Sistem Bank Sampah Malaka Sari :

Tabel 4.1 Identifikasi Aktor

No	Nama Aktor	Deskripsi
1	Admin	Orang yang memiliki hak untuk mengatur hak akses pengguna sistem ( <i>user</i> ) dan mengelola seluruh data didalam aplikasi seperti, mengelola data nasabah, melakukan transaksi nasabah dan memonitoring data bank sampah.
2	Nasabah	Orang yang memiliki hak akses untuk dapat mengecek data saldo dan data transaksi

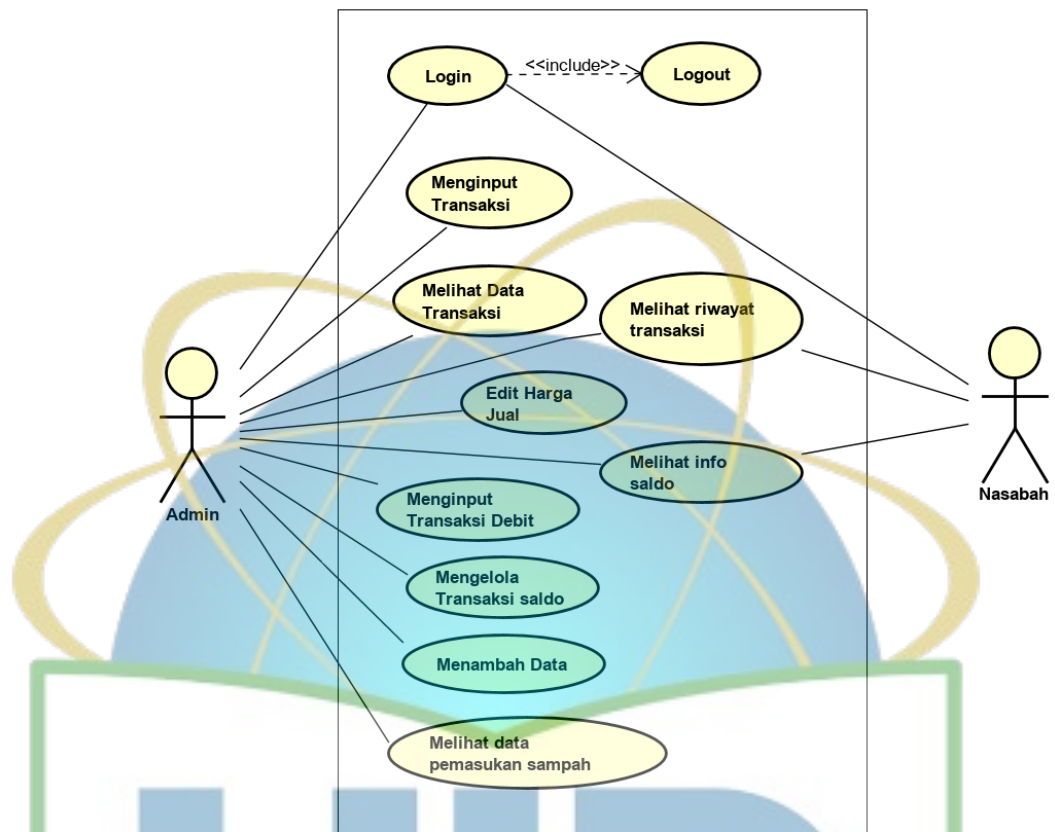
Tabel 4.2 Identifikasi Use Case

No	Nama <i>Use Case</i>	Deskripsi	Aktor
1	<i>Login</i>	<i>Use case</i> menggambarkan kegiatan aktor untuk masuk kedalam sistem dengan memasukkan <i>username</i> dan <i>password</i> .	Admin dan Nasabah
2	Menginput Transaksi	<i>Use case</i> menggambarkan kegiatan user untuk menginput data	Admin

		transaksi bank sampah	
3	Melihat Data Transaksi	<i>Use case</i> menggambarkan kegiatan user untuk melihat jumlah transaksi yang dilakukan oleh masing – masing nasabah	Admin
4	Menginput Debit	<i>Use case</i> menggambarkan kegiatan user untuk menginput transaksi debit nasabah	Admin
5	Menambah Data	<i>Use case</i> menggambarkan kegiatan user untuk menambah jumlah nasabah	Admin
6	Mengelola Transaksi Saldo	<i>Use case</i> menggambarkan kegiatan user untuk menginput transaksi pemasukan dan pengeluaran uang bank sampah	Admin
7	Edit Harga	<i>Use case</i>	Admin

	Jual	menggambarkan kegiatan <i>user</i> untuk mengedit harga jual sampah	
8	Melihat Riwayat Transaksi	<i>Use case</i> menggambarkan kegiatan <i>user</i> untuk melihat transaksi terakhir yang dilakukan oleh nasabah	Admin dan Nasabah
9	Melihat Info Saldo	<i>Use case</i> menggambarkan kegiatan <i>user</i> untuk melihat jumlah saldo	Nasabah
10	Melihat Data Pemasukan Sampah	<i>Use case</i> menggambarkan kegiatan <i>user</i> untuk melihat data pemasukan dalam kurun waktu per bulan	Admin
11	Logout	<i>Use case</i> menggambarkan kegiatan aktor untuk keluar dari sistem	Admin dan Nasabah

Berdasarkan Aktor dan fungsi yang terdapat dalam aplikasi maka digambarkan *use case* seperti berikut :



Gambar 4.4 Use Case Diagram

#### 4.2.2.1.1 Narasi Use Case

Dari *use case* pada gambar akan dijelaskan dengan narasi sebagai berikut :

Tabel 4.3 Deskripsi Use Case Login

<b>UseCase Name</b>	Login
<b>UseCase Id</b>	1
<b>Actor</b>	Admin dan Nasabah
<b>Description</b>	Use case menggambarkan kegiatan user untuk masuk kedalam sistem.



<b>Precondotion</b>	User memasukkan <i>username</i> dan <i>password</i> untuk dapat masuk kedalam system	
<b>Typical Course and Event</b>	<b>Actor</b>	
	<b>Action</b>	<b>System Reponse</b>
	1. Masukkan <i>username</i> dan <i>password</i> .	2. Validasi data <i>username</i> dan <i>password</i> .
	3. Masuk ke home page	
<b>Post Condition</b>	User masuk kedalam sistem.	

Tabel 4.4 Deskripsi Use Case Input Transaksi

<b>UseCase Name</b>	Menginput Transaksi
<b>UseCase Id</b>	2
<b>Actor</b>	Admin
<b>Description</b>	Use case menggambarkan kegiatan user untuk menginput data transaksi bank sampah
<b>Precondotion</b>	Admin menginput data nasabah, tanggal transaksi, jenis sampah, dan berat timbangan

<i>Typical Course and Event</i>	<i>Actor Action</i>	<i>System Reponse</i>
	1. Memilih menu <i>manajemen nasabah</i>	2. Menampilkan submenu <i>manajemen nasabah</i>
	3. Plih menu <i>transksi nasabah</i>	4. Menampilkan <i>form transaksi nasabah</i>
	5.masukkan data dan klik proses	6. menampilkan pesan “transaksi nasabah berhasil”
<i>Post Condition</i>	Hasil transaksi disimpan di dalam database	

Tabel 4.5 Deskripsi Use Case Lihat Data Transaksi

<i>UseCase Name</i>	Melihat data transaksi
<i>UseCase Id</i>	3
<i>Actor</i>	Admin
<i>Description</i>	<i>Use case</i> menggambarkan kegiatan user untuk melihat jumlah transaksi yang dilakukan oleh masing – masing nasabah
<i>Precondotion</i>	Pilih menu manajemen nasabah – data

	transaksi nasabah	
<b>Typical Course and Event</b>	<b>Actor Action</b>	<b>System Reponse</b>
	1. Memilih menu <i>manajemen nasabah</i> .	2. Menampilkan submenu <i>manajemen nasabah</i>
	3. Plih menu <i>data transaksi nasabah</i>	4. Menampilkan <i>table data transaksi dari masing – masing nasabah</i>
<b>Post Condition</b>	System menampilkan data transaksi dari masing – masing nasabah	

Tabel 4.5 Deskripsi Use Case Input Debit

<b>UseCase Name</b>	Menginput Debit	
<b>UseCase Id</b>	4	
<b>Actor</b>	Admin	
<b>Description</b>	<i>Use case</i> menggambarkan kegiatan <i>user</i> untuk menginput transaksi debit nasabah	
<b>Precondotion</b>	Admin menginput jumlah nominal uang	
<b>Typical Course and Event</b>	<b>Actor Action</b>	<b>System Reponse</b>

	1. Memilih menu <i>manajemen nasabah</i>	2. Menampilkan submenu <i>manajemen nasabah</i>
	3. Pilih menu <i>debit nasabah</i>	4. Menampilkan <i>table</i> yang berisi riwayat <i>transaksi debit nasabah</i>
	5. pilih menu <i>transaksi debit</i>	6. menampilkan <i>form transaksi debit</i>
	7.masukkan data dan klik proses	8. menampilkan pesan “transaksi debit <i>nasabah</i> berhasil”
<b>Post Condition</b>	Hasil transaksi disimpan di dalam database	

Tabel 4.6 Deskripsi Use Case Tambah Data

<b>UseCase Name</b>	Menambah Data	
<b>UseCase Id</b>	5	
<b>Actor</b>	Admin	
<b>Description</b>	<i>Use case</i> menggambarkan kegiatan user untuk menambah jumlah nasabah	
<b>Precondotion</b>	Admin menginput nama nasabah, no telepon dan alamat nasabah	
<b>Typical Course</b>	<b>Actor</b>	<b>System Reponse</b>

<i>and Event</i>	<i>Action</i>	
	1. Memilih menu <i>manajemen nasabah</i>	2. Menampilkan submenu <i>manajemen nasabah</i>
	3. Pilih menu <i>data nasabah</i>	4. Menampilkan <i>table</i> yang berisi <i>data nasabah</i>
	5. pilih menu <i>tambah nasabah</i>	6. menampilkan <i>form tambah nasabah</i>
	7. masukan data dan klik simpan	8. menampilkan pesan “tambah nasabah berhasil”
<i>Post Condition</i>	Hasil transaksi disimpan di dalam database	

Tabel 4.7 Deskripsi Use Case Lihat Data Transaksi

<i>UseCase Name</i>	Mengelola Transaksi Saldo
<i>UseCase Id</i>	6
<i>Actor</i>	Admin
<i>Description</i>	<i>Use case</i> menggambarkan kegiatan <i>user</i> untuk menginput transaksi pemasukan dan pengeluaran uang bank sampah

<b>Precondotion</b>	Admin menginput tanggal transaksi, jumlah nominal uang dan keterangan transaksi	
<b>Typical Course and Event</b>	<b>Actor Action</b>	<b>System Reponse</b>
	1. Memilih menu <i>bank sampah</i>	2. Menampilkan submenu <i>bank sampah</i>
	3. Plih menu <i>saldo</i>	4. Menampilkan <i>table yang berisi data pemasukan dan pengeluaran uang bank sampah</i>
	4.pilih menu <i>pemasukan saldo</i>	6. menampilkan <i>form pemasukan saldo</i>
	7. masukan data dan klik proses	8. menampilkan pesan “tambah saldo bank sampah berhasil”
	9. pilih menu <i>pengeluaran saldo</i>	10. menampilkan form <i>pengeluaran saldo</i>
	11. masukan data dan klik proses	12. menampilkan pesan “pengeluaran saldo bank sampah berhasil”
<b>Post Condition</b>	Hasil transaksi disimpan di dalam	



	database
--	----------

Tabel 4.8 Deskripsi Use Case Edit Harga Jual

<b>UseCase Name</b>	Edit Harga Jual	
<b>UseCase Id</b>	7	
<b>Actor</b>	Admin	
<b>Description</b>	Use case menggambarkan kegiatan user untuk mengedit harga jual jenis sampah	
<b>Precondotion</b>	Admin menginput nominal harga jual baru untuk 1 jenis sampah	
<b>Typical Course and Event</b>	<b>Actor</b>	<b>System Reponse</b>
	<b>Action</b>	
	1. Pilih menu <i>Bank Sampah</i>	2. Menampilkan submenu <i>bank sampah</i>
	3. Plih menu <i>harga jual</i>	4. Menampilkan <i>table</i> yang berisi <i>harga jual</i> dari masing – masing <i>sampah</i>
	5. pilih menu <i>edit harga jual</i>	6. menampilkan form <i>edit harga jual</i> bank sampah
	7. masukan data dan klik <i>simpan</i>	8. menampilkan pesan “harga jual sampah berhasil diubah”

<b>Post Condition</b>	Harga jual yang baru telah di simpan di dalam database
-----------------------	--

**Tabel 4.9** Deskripsi Use Case Lihat Riwayat Transaksi

<b>UseCase Name</b>	Melihat Riwayat Transaksi	
<b>UseCase Id</b>	8	
<b>Actor</b>	Admin dan Nasabah	
<b>Description</b>	<i>Use case</i> menggambarkan kegiatan user untuk melihat transaksi terakhir yang dilakukan oleh nasabah	
<b>Precondotion</b>	Pilih menu manajemen nasabah – data transaksi nasabah	
<b>Typical Course and Event</b>	<b>Actor</b>	
	<b>Action</b>	<b>System Reponse</b>
	1. Pilih menu <i>Bank Sampah</i>	2. Menampilkan submenu <i>bank sampah</i>
	3. Plih menu <i>riwayat transaksi nasabah</i>	4. Menampilkan <i>table</i> yang berisi transaksi terakhir yang telah dilakukan oleh nasabah
<b>Post Condition</b>	System menampilkan transaksi terakhir yang telah dilakukan oleh nasabah	

Tabel 4.10 Deskripsi Use Case Lihat Data Pemasukan Sampah

<b>UseCase Name</b>	Melihat Data Pemasukan Sampah	
<b>UseCase Id</b>	9	
<b>Actor</b>	Admin	
<b>Description</b>	Use case menggambarkan kegiatan user untuk melihat data pemasukan dalam kurun waktu per bulan	
<b>Precondotion</b>	Pilih menu home	
<b>Typical Course and Event</b>	<b>Actor Action</b>	<b>System Reponse</b>
	1. Pilih menu home	2. Menampilkan grafik dr jumlah pemasukan sampah dalam kurun waktu per bulan
<b>Post Condition</b>	System menampilkan grafik dari data pemasukan sampah	

Tabel 4.11 Deskripsi Use Case Log Out

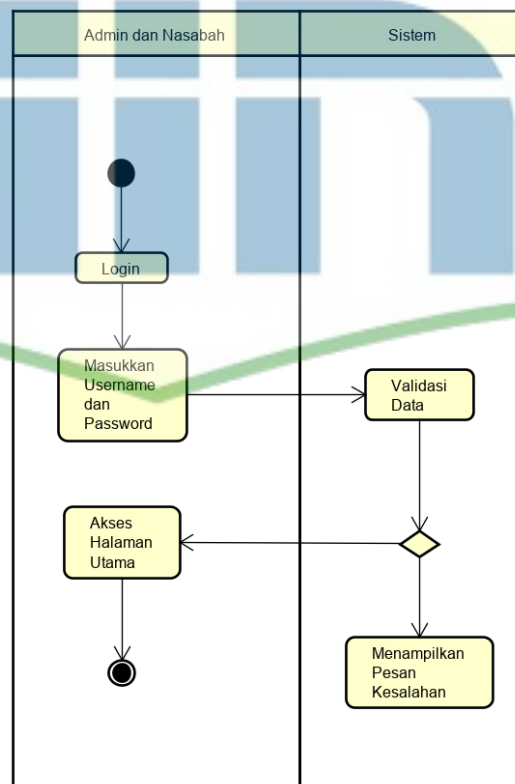
<b>UseCase Name</b>	Log Out
<b>UseCase Id</b>	10
<b>Actor</b>	Admin
<b>Description</b>	Use case menggambarkan kegiatan user untuk dapat keluar dari aplikasi
<b>Precondotion</b>	Admin harus melakukan proses login

	terlebih dahulu	
<b>Typical Course and Event</b>	<b>Actor Action</b>	<b>System Reponse</b>
	1. Pilih menu <i>sign out</i>	2. Menampilkan halaman <i>log in</i>
<b>Post Condition</b>	Aktor berhasil keluar dari sistem	

#### 4.2.2.2 Activity Diagram

Setelah perancangan use case selesai maka akan dilanjutkan dengan perancangan *activity diagram* untuk menggambarkan kegiatan sistem. Berikut adalah penggambaran *activity diagram*:

##### 1. Acitivty Diagram Login



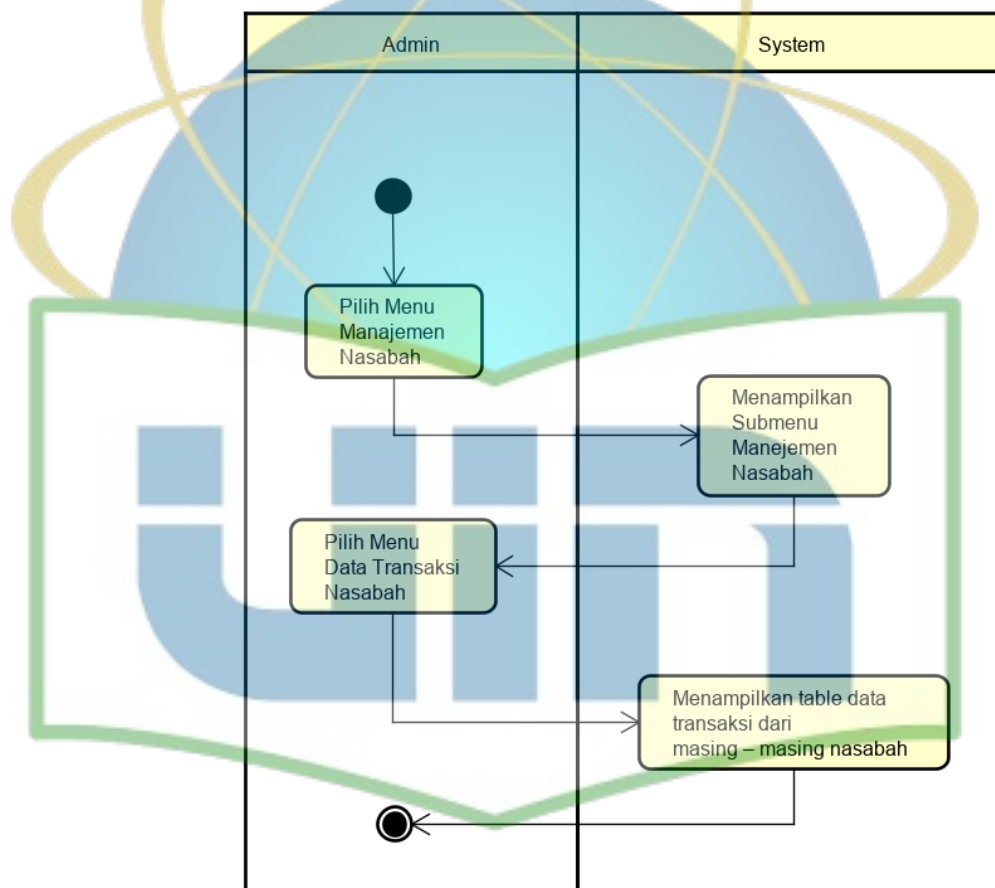
**Gambar 4.5** Activity Diagram Login



Keterangan:

Alur dan proses dari *activity diagram* menginput transaksi adalah aktor membuka menu manajemen nasabah dan memilih sub menu transaksi nasabah lalu masukkan data yang diperlukan untuk proses transaksi kemudian pilih proses, data akan otomatis tersimpan di dalam database.

### 3. Melihat Data Transaksi

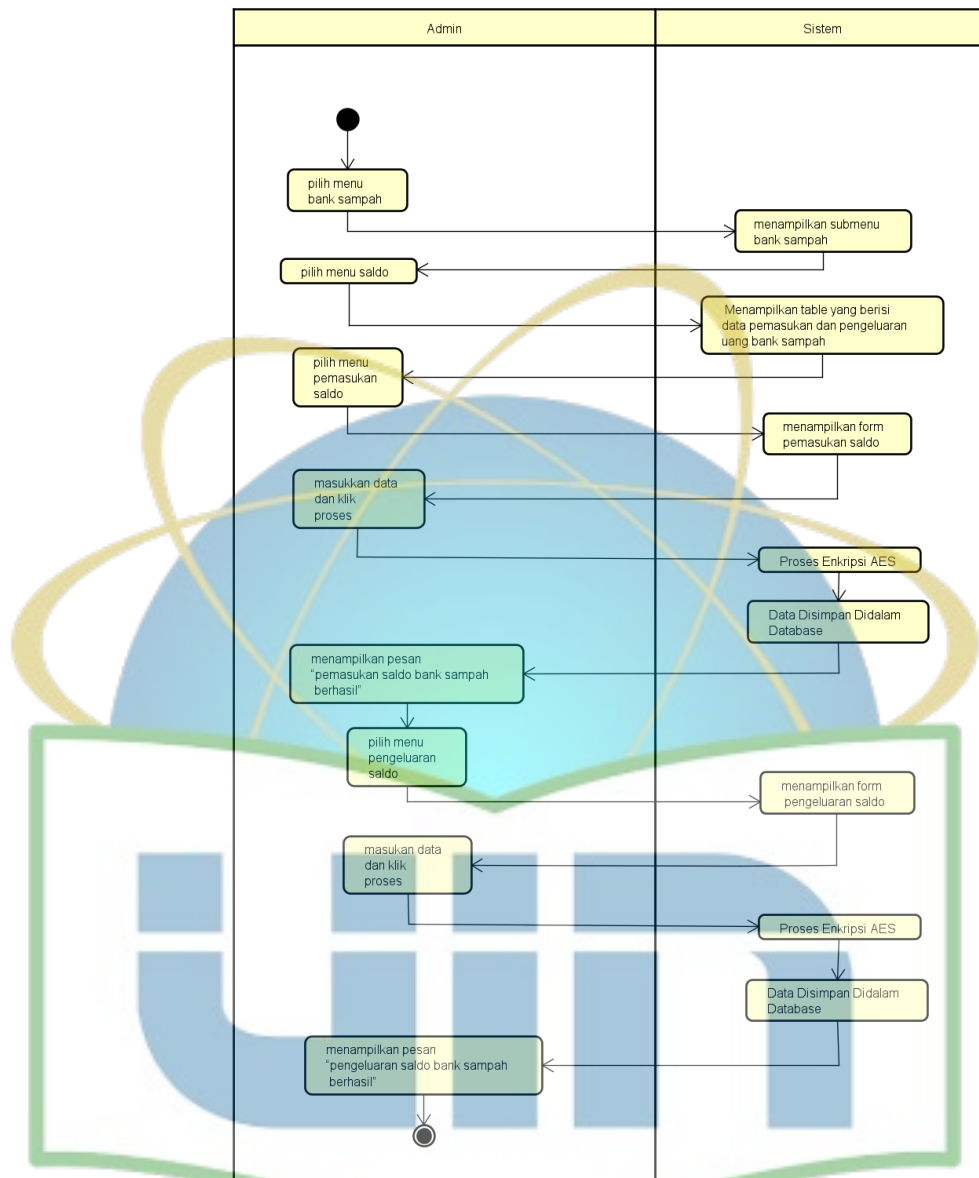


**Gambar 4.7** Activity Diagram Data Transaksi

Alur dan proses dari *activity diagram* melihat data transaksi adalah system menampilkan data transaksi dari masing – masing nasabah yang sudah melakukan transaksi di bank sampah.



#### 4. Menginput Debit



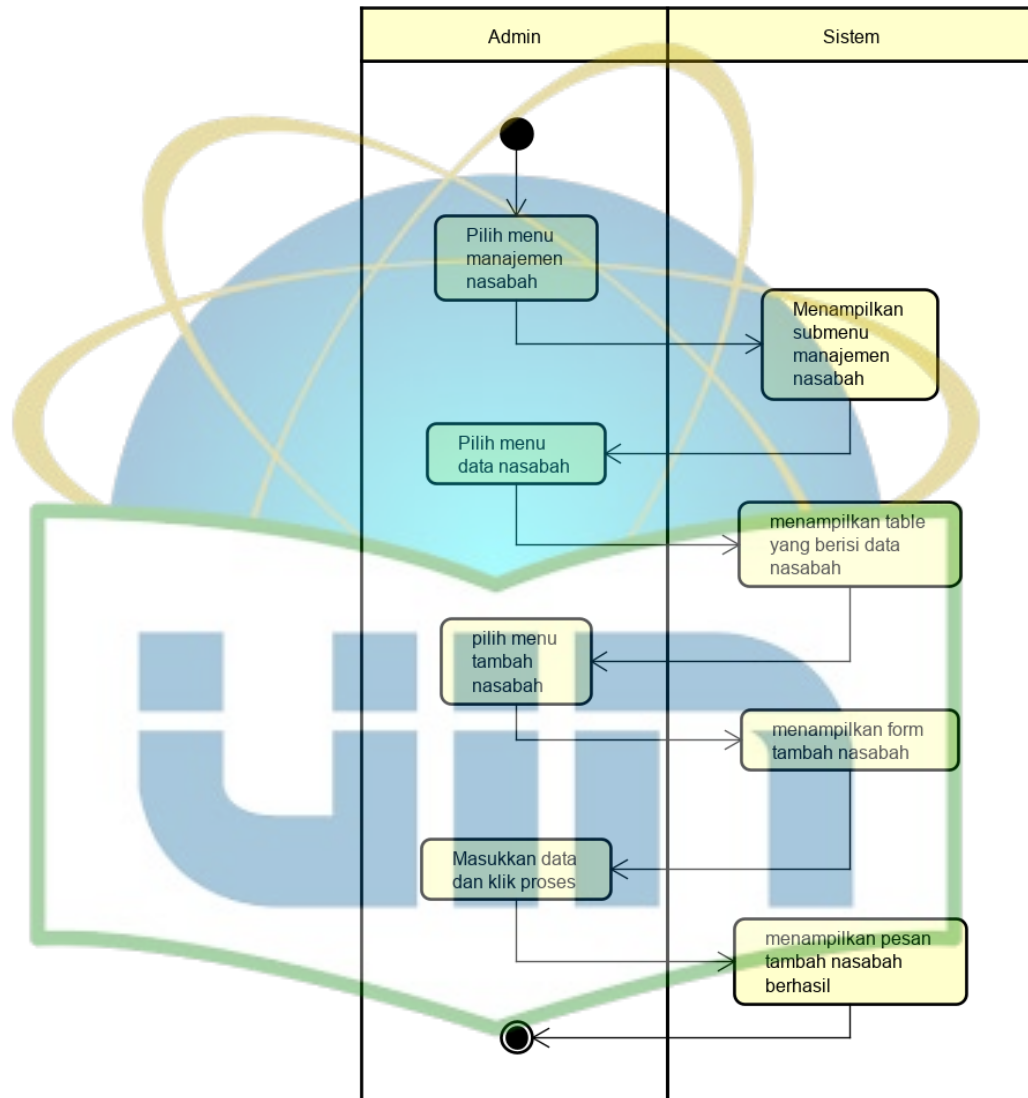
**Gambar 4.8** Activity Diagram Input Debit

Keterangan:

Alur dan proses dari *activity diagram* menginput debi adalah Aktor membuka menu manajemen nasabah dan memilih submenu debit nasabah kemudian system akan menampilkan jumlah saldo dari masing - masing nasabah lalu aktor akan menginput jumlah nominal uang yang nasabah

perlu, selanjutnya klik simpan lalu data akan tersimpan di dalam database. Selanjutnya system menampilkan table data transaksi debit yang sudah dilakukan oleh masing – masing nasabah.

## 5. Menambah Data

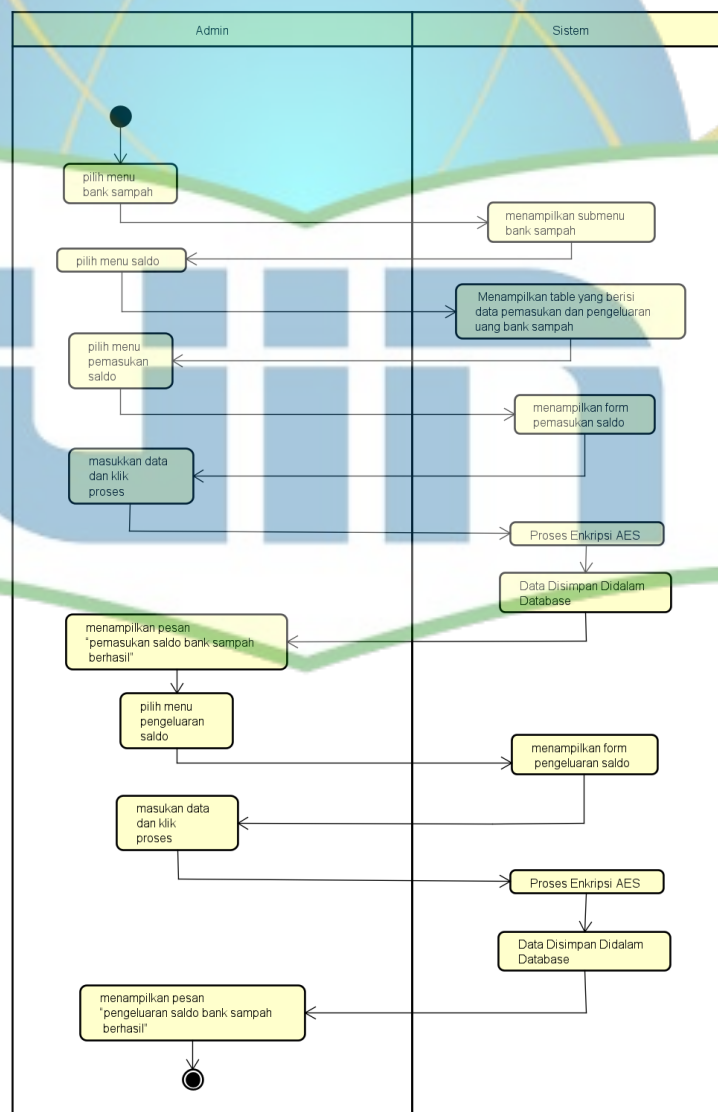


**Gambar 4.9** Activity Diagram Tambah Data

### Keterangan:

Alur dan proses dari *activity diagram* menambah data adalah Aktor membuka menu manajemen nasabah dan memilih submenu data nasabah kemudian system akan menampilkan form yang berisi nama nasabah, no telepon dan alamat, kemudian aktor masukkan data dan klik simpan, data nasabah baru akan disimpan di dalam database, kemudian system menampilkan jumlah nasabah dan data dari masing – masing nasabah.

### 6. Mengelola Transaksi Saldo

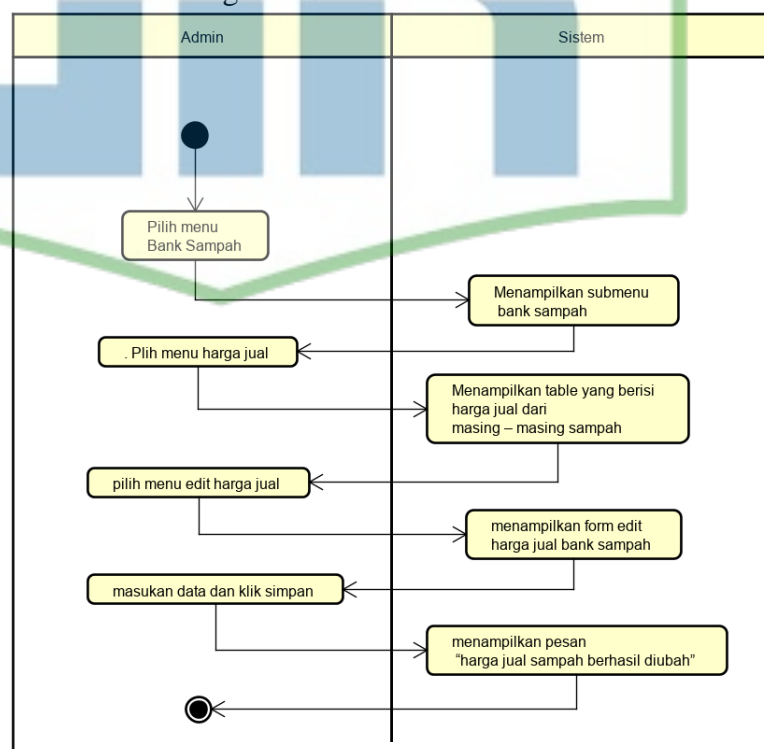


**Gambar 4.10** Activity Diagram Transaksi Saldo

### Keterangan:

Alur dan proses dari *activity diagram* mengelola transaksi saldo adalah Aktor membuka menu bank sampah dan memilih submenu saldo kemudian system akan menampilkan table data rincian transaksi pemasukan dan pengeluaran uang bank sampah. lalu pilih pemasukan saldo, system akan menampilkan form yang berisi jumlah saldo bank sampah, tanggal transaksi, jumlah yang di input dan keterangan transaksi, kemudian klik simpan, lalu data akan disimpan di dalam database. Dan saldo akan bertambah, Jika aktor memilih pengurangan saldo, system akan menampilkan form yang berisi jumlah saldo bank sampah, tanggal transaksi, jumlah yang di input dan keterangan transaksi, kemudian klik simpan, lalu data akan disimpan di dalam database. Dan saldo akan berkurang.

### 7. Edit Harga Jual

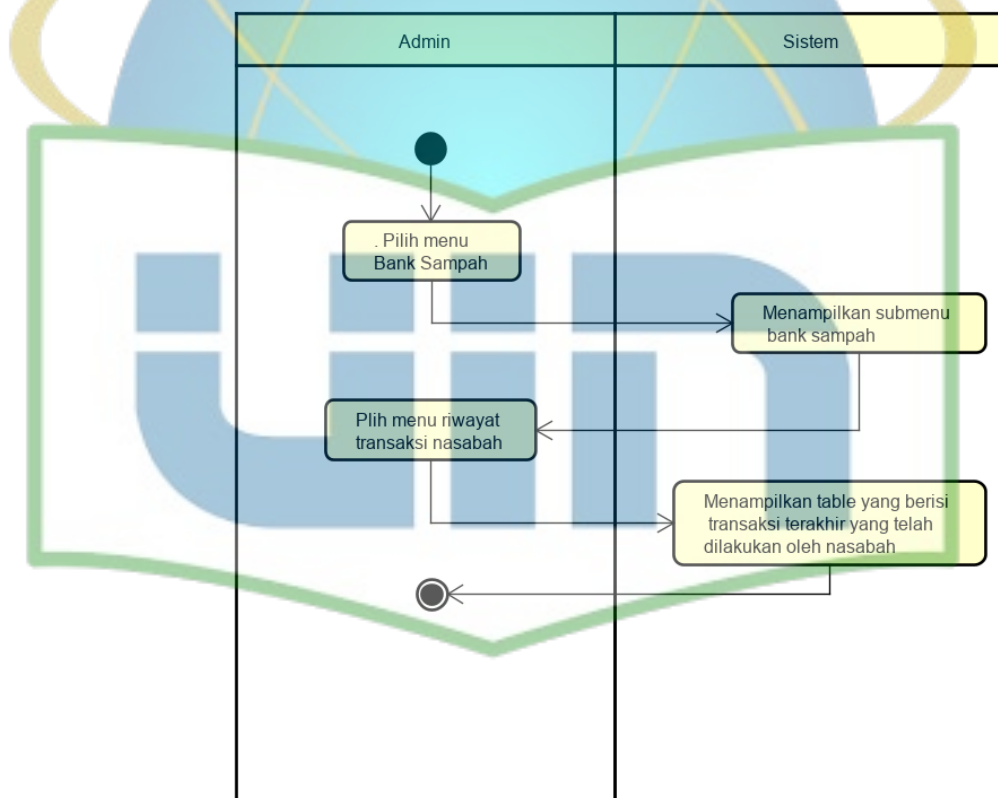


Gambar 4.11 Activity Diagram Edit Harga Jual

Keterangan:

Alur dan proses dari *activity diagram* edit harga jual adalah Aktor membuka menu bank sampah dan memilih submenu data edit harga jual kemudian system akan menampilkan table yang berisi harga dari masing – masing sampah, kemudian klik edit lalu system akan menampilkan form harga satuan sampah per kilo, kemudian aktor menginput harga jual baru, lalu system akan mengupdate harga baru ke dalam database.

#### 8. Melihat Riwayat Transaksi



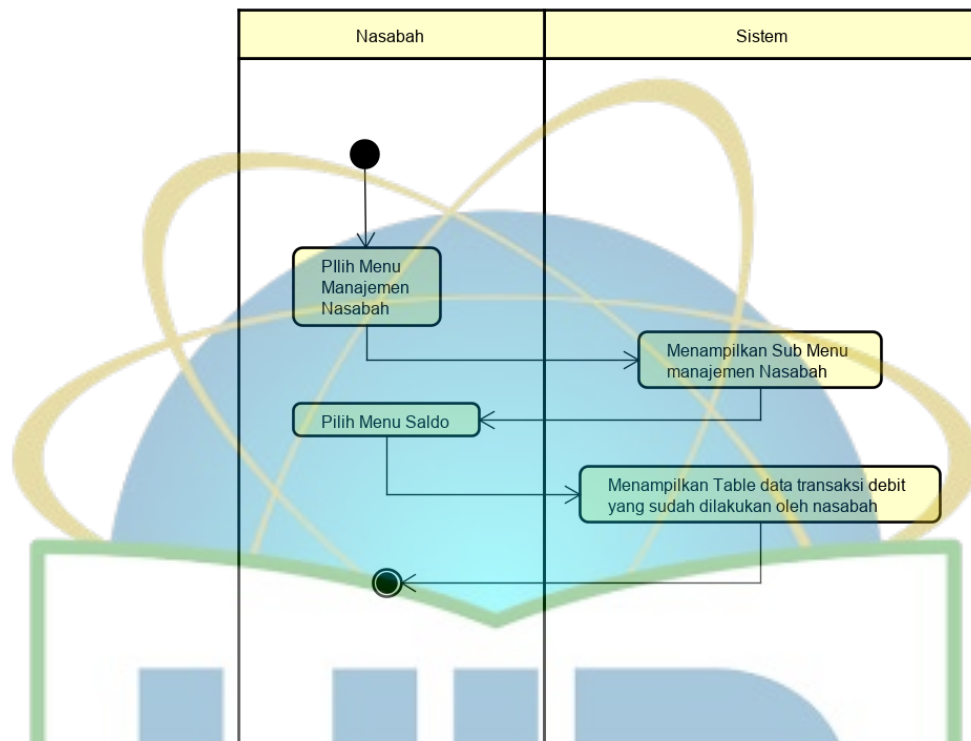
**Gambar 4.12** Activity Diagram Transaksi Saldo

Keterangan:

Alur dan proses dari *activity diagram* melihat riwayat transaksi adalah Aktor membuka menu

bank sampah dan memilih riwayat transaksi nasabah kemudian system akan menampilkan table data rincian transaksi yang dilakukan oleh nasabah

## 9. Melihat Info Saldo



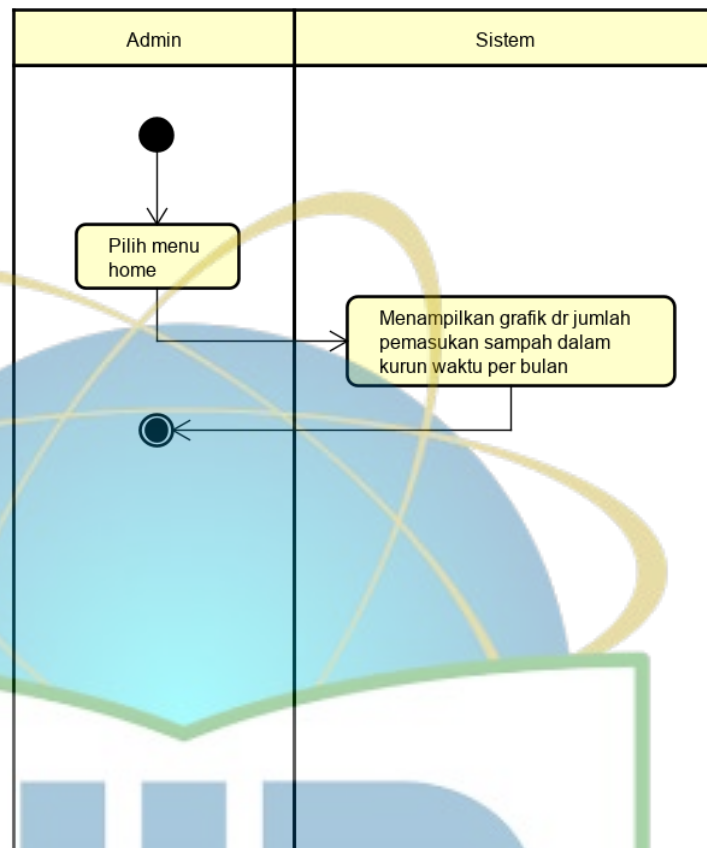
**Gambar 4.13** Activity Diagram Info Saldo

### Keterangan:

Alur dan proses dari *activity diagram* melihat melihat info saldo adalah Aktor membuka menu manajemen nasabah dan memilih sub menu info saldo kemudian system akan menampilkan table data rincian transaksi debit yang sudah dilakukan oleh nasabah.



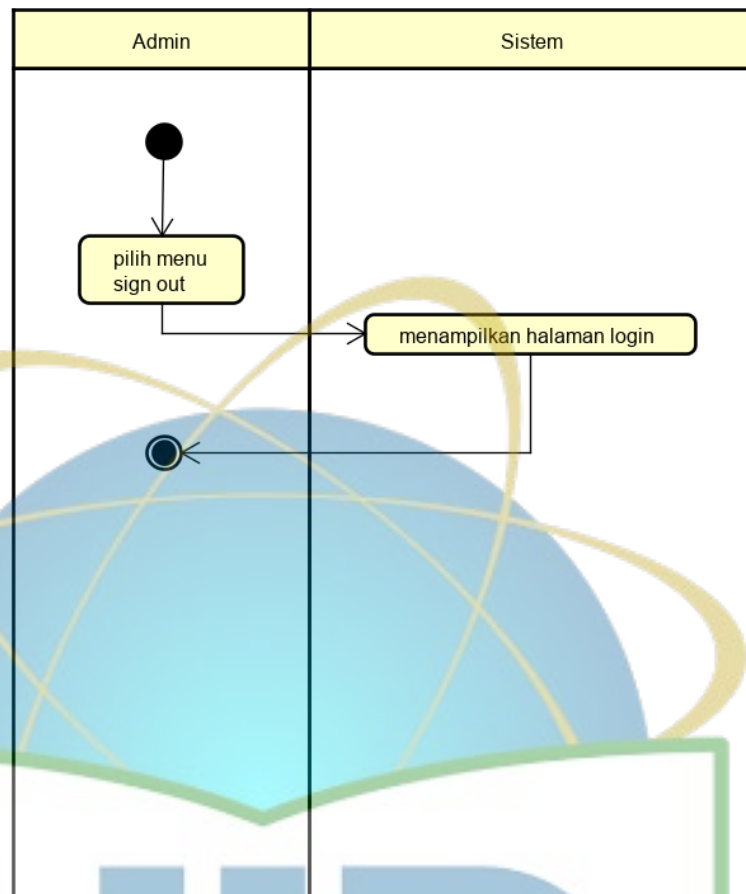
## 10. Melihat Data Pemasukan Sampah



**Gambar 4.14** Activity Diagram Data Pemasukan Sampah

Alur dan proses dari *activity diagram* melihat data pemasukan sampah adalah Aktor pilih menu home, lalu system akan menampilkan grafik pemasukan sampah dalam kurun waktu per bulan.

## 11. Log Out



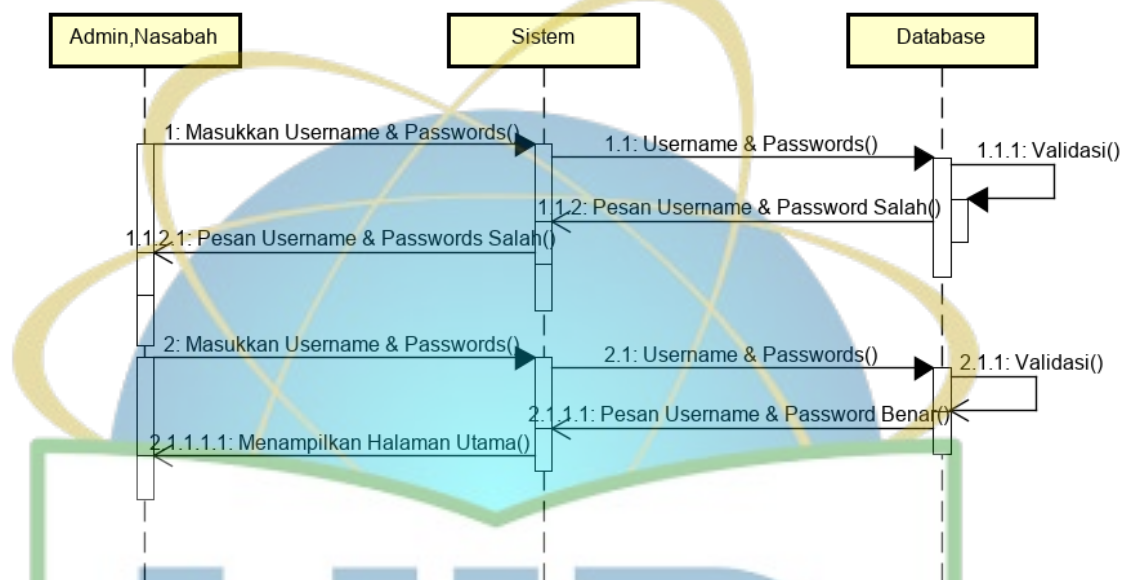
**Gambar 4.15** Activity Diagram Log Out

Alur dan proses dari *activity diagram* log out adalah Aktor pilih menu sign out, lalu aktor akan keluar dari system aplikasi, dan kemudian system akan menampilkan halaman login.

#### 4.2.2.3 Sequence Diagram

Perancangan *sequence diagram* untuk menggambarkan diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu. Berikut adalah penggambaran *sequence diagram*:

##### 1. Sequence Diagram Login

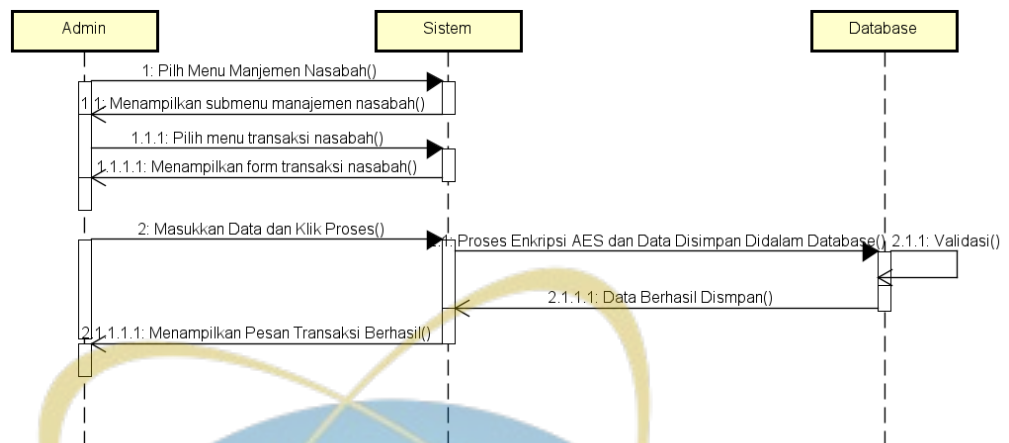


**Gambar 4.16** Sequence Diagram Log in

Keterangan :

*Sequence Diagram* diatas menunjukan proses admin dan nasabah sebagai *user* memasukkan *username* dan *password* kedalam sistem. Jika *username* dan *password* tidak sesuai, maka sistem akan menampilkan pesan “*username* dan *password* yang dimasukkan salah” dan *login* gagal. Jika *username* dan *password* benar, maka sistem akan menampilkan tampilan utama halaman sistem aplikasi bank sampah

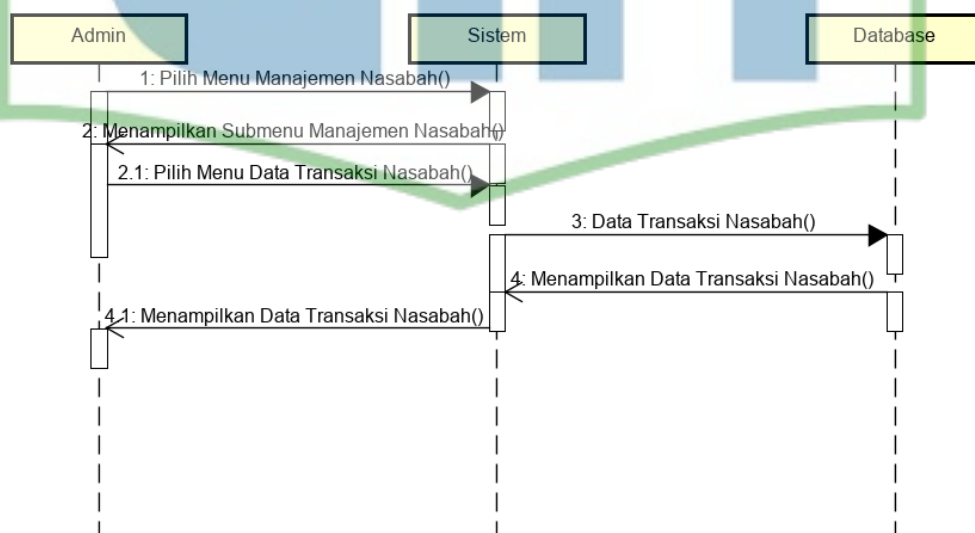
## 2. Sequence Input Transaksi



**Gambar 4.17** Sequence Diagram Input Transaksi

*Sequence Diagram* diatas menunjukan proses Admin menginput transaksi bank sampah, untuk menjalankan proses ini admin diharuskan menginput data nasabah, tanggal transaksi, jenis sampah dan berat timbangan, kemudian data diproses dan disimpan ke dalam database, lalu sistem akan menampilkan pesan “transaksi berhasil” jika semua input data valid.

## 3. Sequence Melihat Data Transaksi

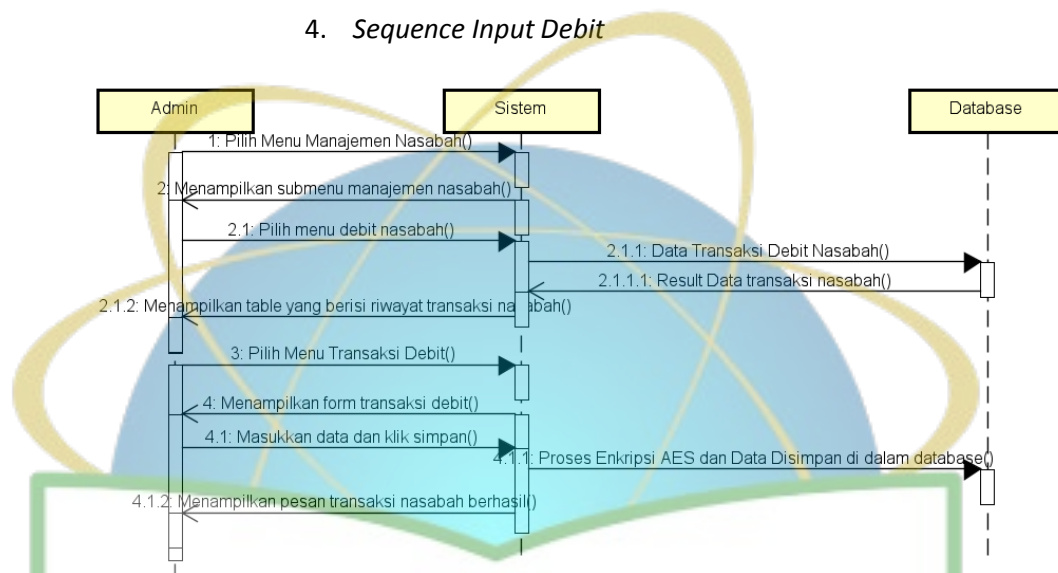


**Gambar 4.18** Sequence Diagram Data Transaksi

Keterangan :

*Sequence Diagram* diatas menunjukan proses Admin untuk mengetahui data transaksi dari masing – masing nasabah yang sudah melakukan transaksi di bank sampah.

#### 4. *Sequence Input Debit*

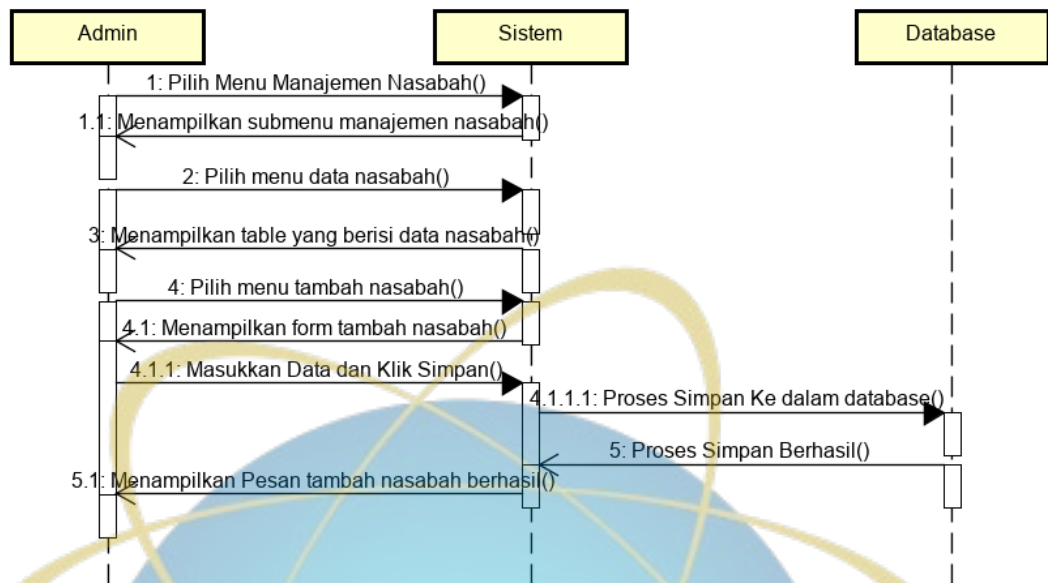


**Gambar 4.19** Sequence Diagram Input Debit

Keterangan :

*Sequence Diagram* diatas menunjukan proses admin untuk mengetahui nasabah yang sudah melakukan debit dan untuk nasabah yang ingin melakukan debit. Admin harus menginput data nasabah kemudian menginput jumlah nominal penarikan, lalu system akan memproses data tersebut, jika valid data akan disimpan di dalam database.

### 5. Sequence Menambah Data

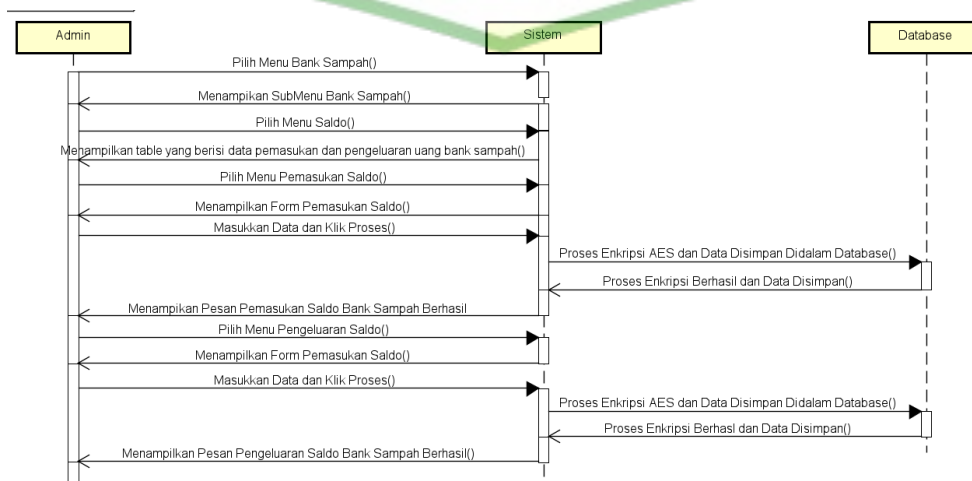


Gambar 4.20 Sequence Diagram Tambah Data

Keterangan :

*Sequence Diagram* diatas menunjukkan proses admin untuk mengetahui jumlah dan data nasabah serta admin dapat menambah jumlah nasabah. Untuk melakukan proses ini, admin harus menginput nama, no telp, dan alamat nasabah, kemudian di proses dan disimpan di dalam database. Yang nanti nya nasabah dapat melakukan login setelah terdaftar di system.

### 6. Sequence Transaksi Saldo

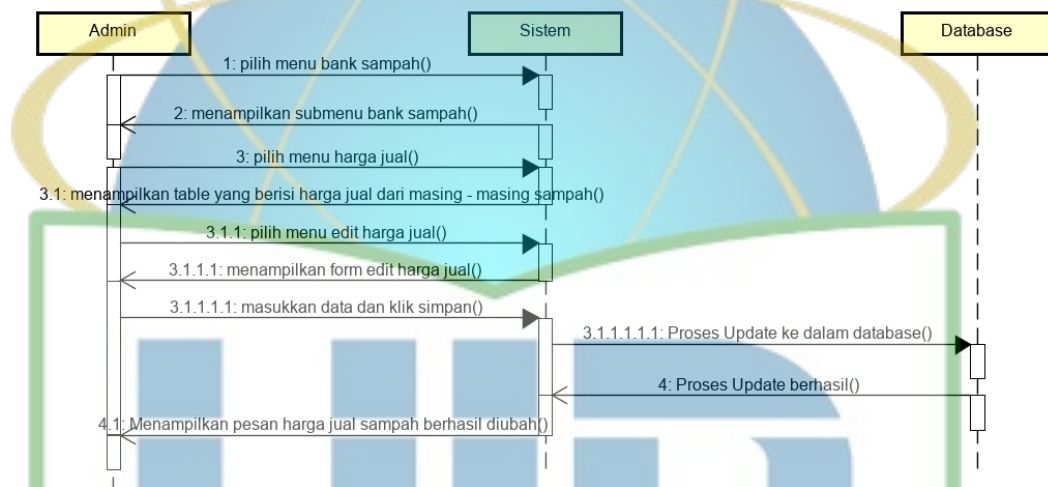




Keterangan :

*Sequence Diagram* diatas menunjukan proses admin untuk mengetahui jumlah saldo bank sampah dan melakukan penarikan dan penambahan saldo bank sampah. Untuk melakukan penarikan dan penambahan admin harus menginput tanggal transaksi, jumlah nominal uang, dan keterangan transaksi, lalu data akan di proses, jika valid data akan disimpan di dalam database.

### 7. *Sequence Edit Harga Jual*

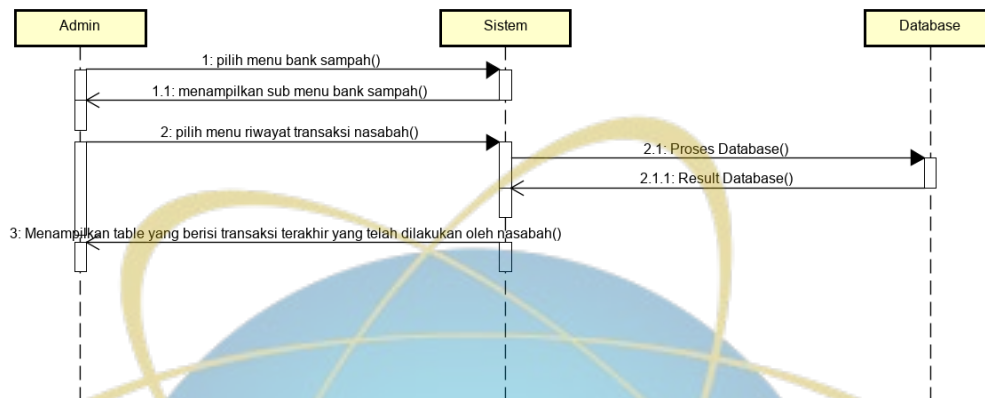


**Gambar 4.22** Sequence Diagram Edit Harga Jual

Keterangan :

*Sequence Diagram* diatas menunjukan proses admin untuk mengedit harga jual sampah, admin harus memilih sampah yang akan dirubah harganya, lalu klik edit, kemudian admin menginput harga jual sampah baru klik simpan lalu system akan proses data tersebut ke dalam database, dan system akan menampilkan pesan “harga jual sampah berhasil diubah”

### 8. Sequence Melihat Riwayat Transaksi

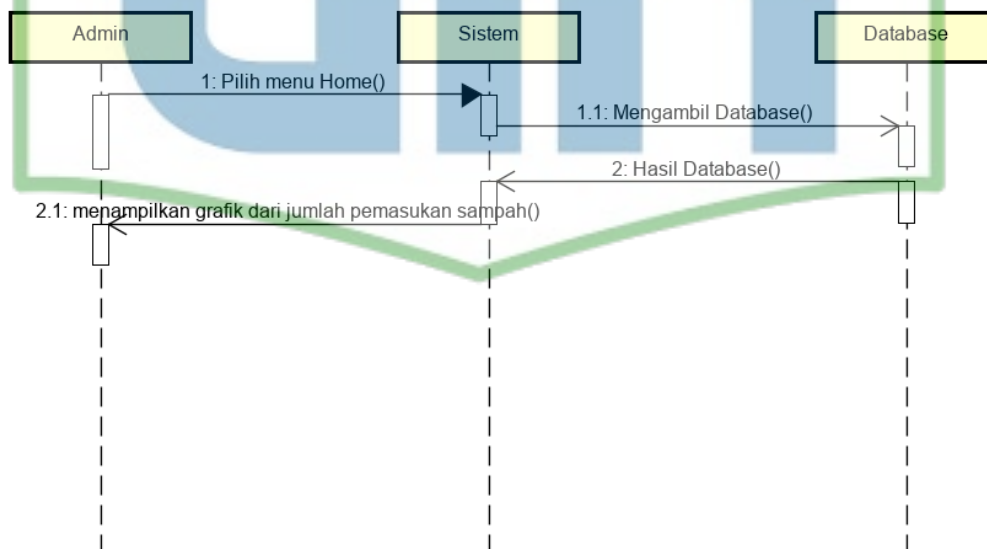


**Gambar 4.23** Sequence Diagram Riwayat Transaksi

Keterangan :

*Sequence Diagram* diatas menunjukan proses admin dan nasabah, system akan menampilkan table data rincian transaksi yang sudah dilakukan oleh nasabah.

### 9. Sequence Data Pemasukan Sampah



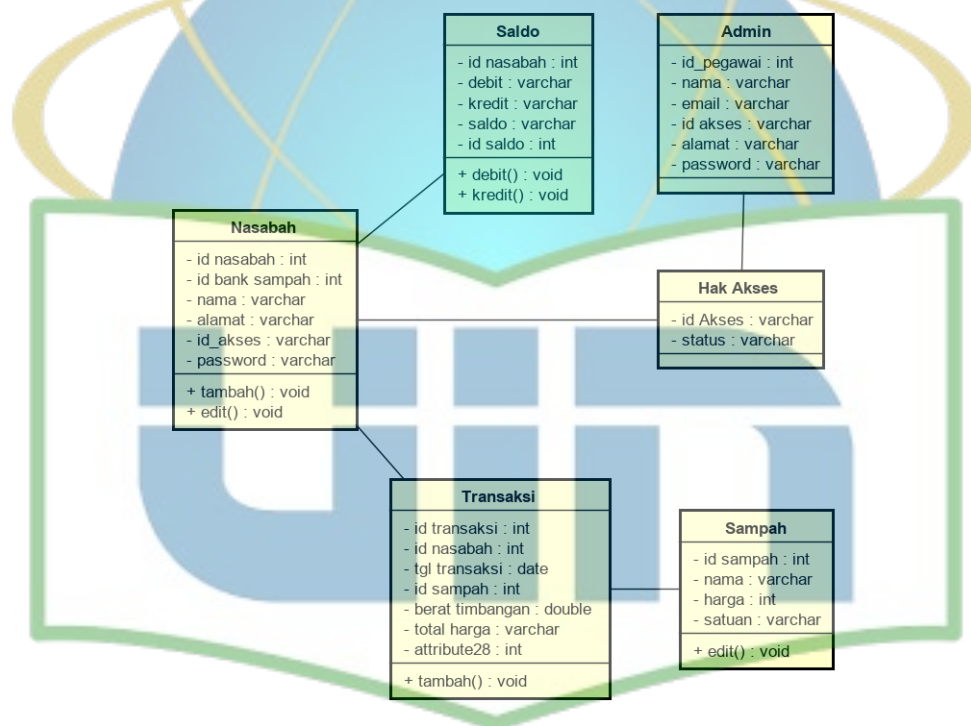
**Gambar 4.24** Sequence Diagram Data Pemasukan

Keterangan :

*Sequence Diagram* diatas menunjukan proses admin untuk melihat data pemasukan sampah per bulan dalam bentuk grafik.

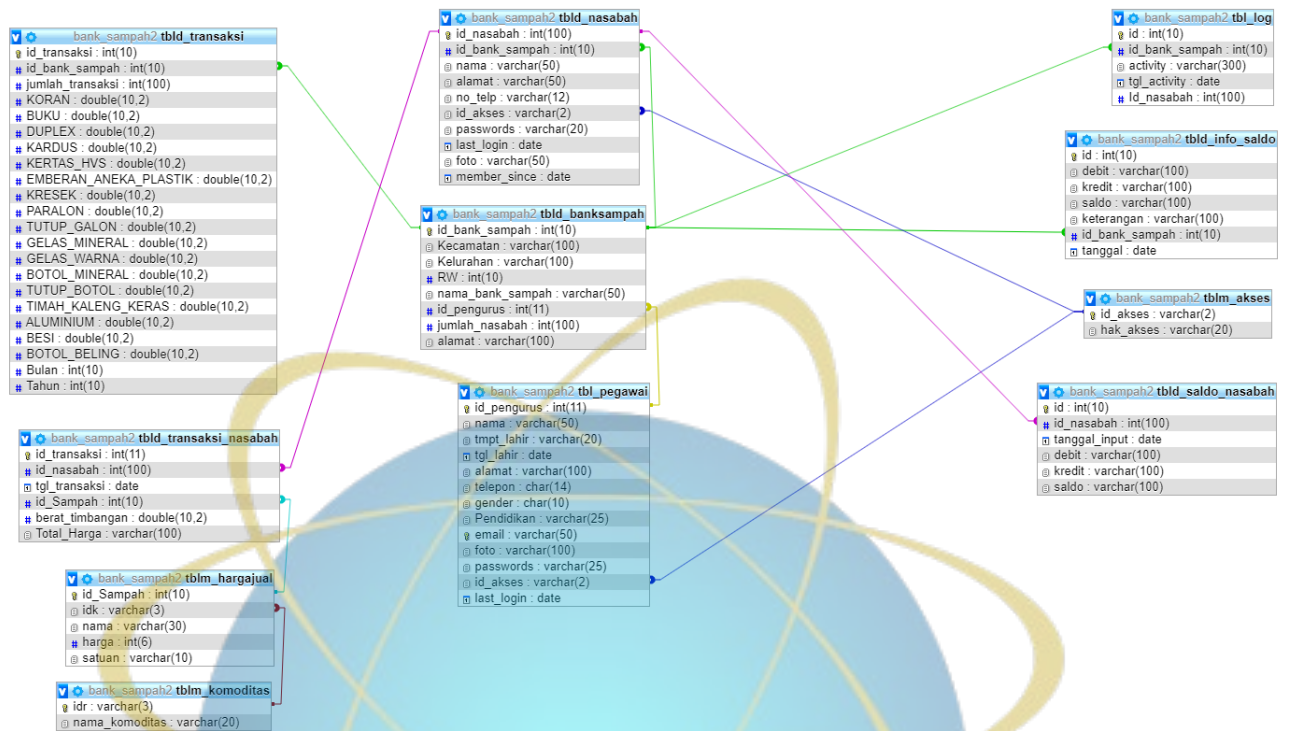
#### 4.2.2.4 Class Diagram

*Class Diagram* ini menjelaskan hubungan antar *table* di dalam model *desain* dari suatu sistem. Aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem, juga diperlihatkan pada *class diagram*, seperti yang di tunjukan pada Gambar 4.25



**Gambar 4.25** Class Diagram

### 4.2.3 Database Schema



Gambar 4.26 Database Schema

#### 4.2.3.1 Spesifikasi Database

Berikut merupakan nama-nama tabel yang digunakan dalam database Aplikasi Bank Sampah Malaka Sari. Tabel tersebut berisi data pegawai sebagai berikut:

Nama Tabel : tbl\_pegawai

Primary Key : id\_pegawai

Foreign Key : id\_akses

Tipe File : Master

Tabel 4.12 Table Pengurus

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_pegawai	int	11	Id Pegawai
2	nama	Varchar	50	Nama Pegawai
3	Tmpt_lahir	Varchar	20	Tempat

				Kelahiran
4	Tgl_lahir	Date	-	Tanggal Lahir
5	Alamat	Varcabar	100	Alamat Pegawai
6	Telepon	Char	14	Telepon Pegawai
7	Gender	Char	10	Jenis Kelamin
8	Pendidikan	Varchar	25	Jenjang Pendidikan
9	Email	Varchar	50	Email Pegawai
10	Foto	Varchar	100	No kartu pasien
11	Passwords	Varchar	25	Password Pegawai
12	Id_akses	Varchar	2	Id akses
13	Last_login	date	-	Terakhir login di aplikasi

Nama Tabel : tbl\_log

Primary Key : id

Foreign Key : id\_bank\_sampah

Id\_nasabah

Tipe File : Master

**Tabel 4.13** Table Log

No	Nama Field	Type	Panjang Field	Keterangan
1	Id	int	10	Id Log
2	Id_Bank_Sampah	int	10	Id Bank Sampah

3	Activity	Varchar	300	Rincian Aktivitas
4	Tgl_activity	Date	-	Tanggal Aktivitas
5	Id_Nasabah	int	100	Id Nasabah

Nama Tabel : tbl\_komoditas

Primary Key : idr

Foreign Key : -

Tipe File : Master

**Tabel 4.14** Table Komoditas

No	Nama Field	Type	Panjang Field	Keterangan
1	Idr	varchar	3	Id komoditas
2	Nama komoditas	varchar	20	Nama komoditas

Nama Tabel : tblm\_hargajual

Primary Key : id\_sampah

Foreign Key : idk

Tipe File : Master

**Tabel 4.15** Table Harga Jual Sampah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_sampah	int	10	Id Sampah
2	Idk	varchar	3	Id Komoditas
3	Jenis Sampah	Varchar	30	Jenis Sampah
4	Harga	int	6	Harga Sampah



5	Satuan	Varchar	10	Satuan Harga
---	--------	---------	----	--------------

Nama Tabel : tblm\_akses

Primary Key : id\_akses

Foreign Key : -

Tipe File : Master

**Tabel 4.16** Table Id Akses

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_akses	varchar	2	Id Akses
2	Hak Akses	varchar	20	Hak Akses

Nama Tabel : tbld\_transaksi\_nasabah

Primary Key : id\_transaksi

Foreign Key : id\_nasabah

id\_sampah

Tipe File : Master

**Tabel 4.17** Table Transaksi Nasabah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_transaksi	Int	11	Id Transaksi
2	Id_nasabah	int	100	Id Nasabah
3	Tgl_transaksi	date	-	Tanggal Transaksi
4	Id_sampah	int	6	Id Sampah
5	Berat_Timbangan	double	10,2	Berat Timbangan Sampah
6	Total_Harga	Varchar	100	Berat timbangan

				dikali harga satuan
--	--	--	--	---------------------

Nama Tabel : tbld\_transaksi

Primary Key : id\_transaksi

Foreign Key : id\_bank\_sampah

Tipe File : Master

**Tabel 4.18** Table Data Pemasukan Sampah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_transaksi	Int	10	Id Transaksi
2	Id_bank_sampah	int	10	Id Bank Sampah
3	Jumlah_transaksi	int	100	Jumlah Transaksi Per bulan
4	Koran	Double	10,2	Total data pemasukan sampah koran
5	Buku	Double	10,2	Total data pemasukan sampah buku
6	Duplex	Double	10,2	Total data pemasukan sampah duplex
7	Kardus	Double	10,2	Total data pemasukan

				sampah kardus
8	Kertas_HVS	Double	10,2	Total data pemasukan sampah kertas hvs
9	Emberan_Aneka_Plastik	Double	10,2	Total data pemasukan sampah emberan aneka plastik
10	Kresek	Double	10,2	Total data pemasukan sampah kresek
11	Paralon	Double	10,2	Total data pemasukan sampah paralon
12	Tutup_Galon	Double	10,2	Total data pemasukan sampah tutup galon
13	Gelas_Mineral	Double	10,2	Total data pemasukan sampah gelas mineral
14	Gelas_Warna	Double	10,2	Total data pemasukan sampah gelas

				warna
15	Botol_Mineral	Double	10,2	Total data pemasukan sampah botol mineral
16	Tutup_Botol	Double	10,2	Total data pemasukan sampah tutup botol
17	Timah_Kaleng_Keras	Double	10,2	Total data pemasukan sampah timah kaleng keras
18	Aluminum	Double	10,2	Total data pemasukan sampah aluminum
19	Besi	Double	10,2	Total data pemasukan sampah besi
20	Botol_Beling	Double	10,2	Total data pemasukan sampah botol beling
21	Bulan	Int	10	Bulan Data Pemasukan Sampah

22	Tahun	int	10	Tahun Data Pemasukan Sampah
----	-------	-----	----	-----------------------------

Nama Tabel : tbld\_saldo\_nasabah

Primary Key : id

Foreign Key : id\_nasabah

Tipe File : Master

**Tabel 4.19** Table Saldo Nasabah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id	Int	10	Id Transaksi
2	Id_nasabah	int	100	Id Nasabah
3	Tgl_input	date	-	Tanggal Transaksi
4	debit	Varchar	100	Debit Nasabah
5	kredit	varchar	100	Kredit Nasabah
6	Saldo	Varchar	100	Saldo Nasabah

Nama Tabel : tbld\_nasabah

Primary Key : id\_nasabah

Foreign Key : id\_bank\_sampah

id\_akses

Tipe File : Master

**Tabel 4.20** Table Nasabah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_nasabah	Int	100	Id nasabah

2	Id_bank_sampah	int	10	Id bank sampah
3	Nama	Varchar	50	Nama nasabah
4	Alamat	varchar	50	Alamat nasabah
5	No_Telp	varchar	12	No telepon nasabah
6	Id_akses	Varchar	2	Id akses nasabah
7	passwords	varchar	20	Password nasabah
8	Last_login	date	-	Login terakhir nasabah
9	Foto	varchar	50	Foto nasabah
10	Member_since	date	-	Lama nasabah bergabung

Nama Tabel : tbld\_info\_saldo

Primary Key : id

Foreign Key : id\_bank\_sampah

Tipe File : Master

Tabel 4.21 Table Saldo Bank Sampah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id	Int	10	Id Transaksi
2	debit	Varchar	100	Debit bank sampah
3	kredit	varchar	100	Kredit bank sampah



4	Saldo	Varchar	100	Saldo bank sampah
5	Keterangan	varchar	100	Keterangan transaksi
6	Id_bank_sampah	int	10	Id_bank_sampah
7	tanggal	date	-	Tanggal transaksi

Nama Tabel : tbl\_bank\_sampah

Primary Key : id\_bank\_sampah

Foreign Key : -

Tipe File : Master

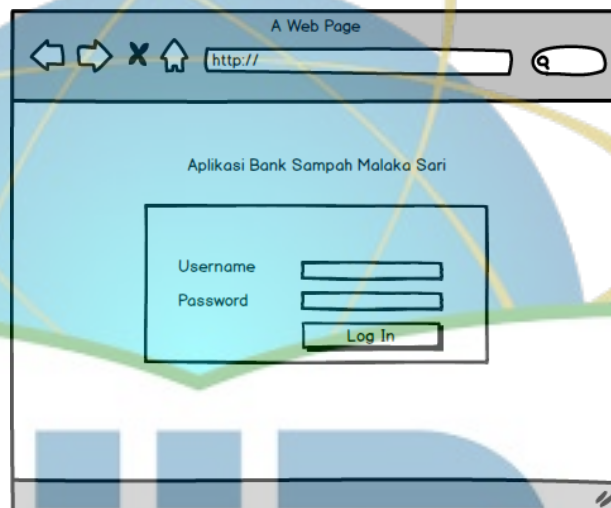
**Tabel 4.22** Table Bank Sampah

No	Nama Field	Type	Panjang Field	Keterangan
1	Id_bank_sampah	Int	10	Id Transaksi
2	Kecamatan	Varchar	100	Kecamatan bank sampah
3	Kelurahan	varchar	100	Kelurahan bank sampah
4	RW	int	10	RW bank sampah
5	Nama bank sampah	varchar	50	Nama bank sampah
6	Id_pengurus	int	11	Id_pengurus
7	Jumlah_nasabah	Int	100	Jumlah nasabah

8	Alamat	Varchar	100	Alamat bank sampah
---	--------	---------	-----	--------------------

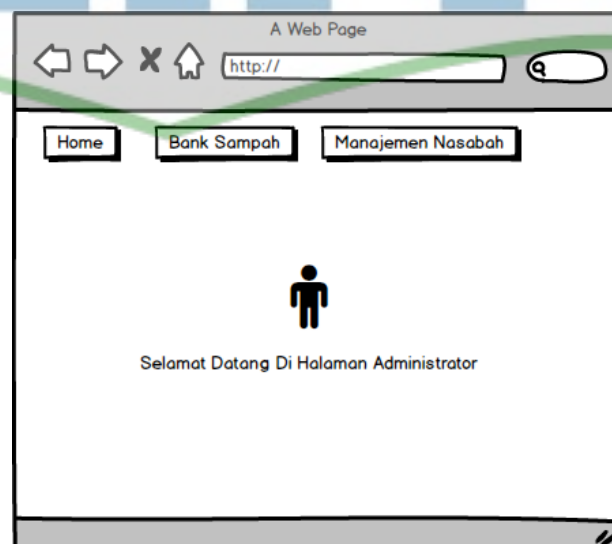
#### 4.2.4 Perancangan User interface

Tampilan antarmuka pengguna dibuat untuk memudahkan dalam pembangunan Sistem Informasi Bank Sampah yaitu dengan membuat rancangan bagi setiap pengguna sistem. Berikut ini rancangan Sistem Informasi Bank Sampah Malaka Sari.



**Gambar 4.27** Halaman Login

Halaman ini merupakan halaman login yang dapat digunakan oleh pengguna aplikasi seperti, admin dan nasabah.



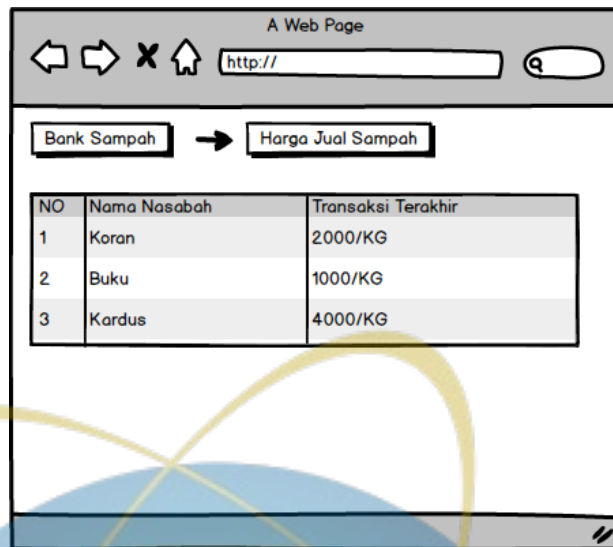
**Gambar 4.28** Halaman Admin

Halaman ini merupakan tampilan dari halaman utama seorang admin, dimana seluruh menu yang ada didalamnya dapat diakses seluruhnya oleh admin.



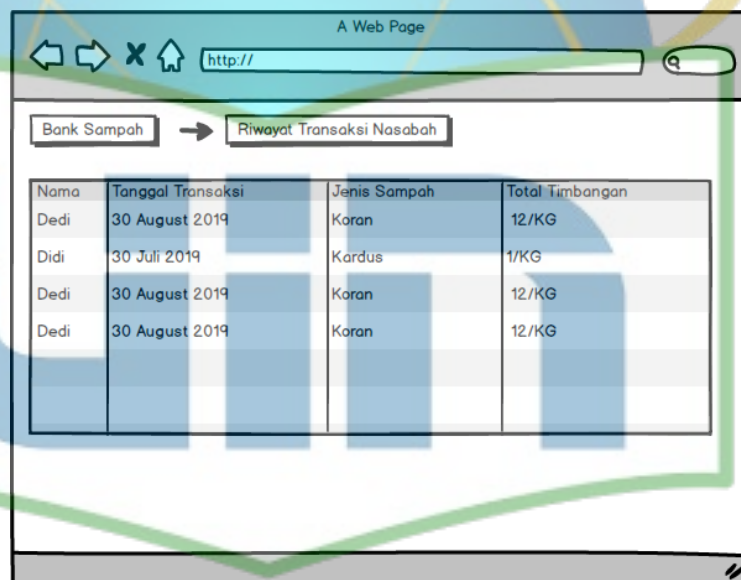
Gambar 4.29 Halaman Saldo Bank Sampah

Halaman ini merupakan salah satu menu yang ada di halaman admin yaitu, saldo bank sampah, dalam menu ini admin dapat memonitoring setiap pengeluaran dan pendapatan yang diperoleh dari bank sampah.



**Gambar 4.30** Halaman Harga Jual Sampah

Halaman ini merupakan halaman harga jual sampah, di halaman ini admin dapat merubah harga jual sampah.



**Gambar 4.31** Halaman Riwayat Transaksi

Halaman ini merupakan halaman riwayat transaksi nasabah, di halaman ini admin dapat mengetahui transaksi terakhir yang dilakukan oleh bank sampah.

**Gambar 4.32** Halaman Transaksi Nasabah

Halaman ini merupakan halaman transaksi nasabah yang hanya dapat diakses oleh admin, di halaman ini admin menginput nasabah, tanggal transaksi, jenis sampah dan berat timbangan.

NO	Nama Nasabah	Transaksi Terakhir	Data Transaksi
1	Yogi	05-July-2019	Data
2	Fahmi	07-July-2019	Data
3	Adit	08-July-2019	Data

**Gambar 4.33** Data Transaksi Nasabah

Halaman ini merupakan halaman data transaksi nasabah, halaman ini dapat melihat seluruh transaksi yang dilakukan oleh masing – masing nasabah.

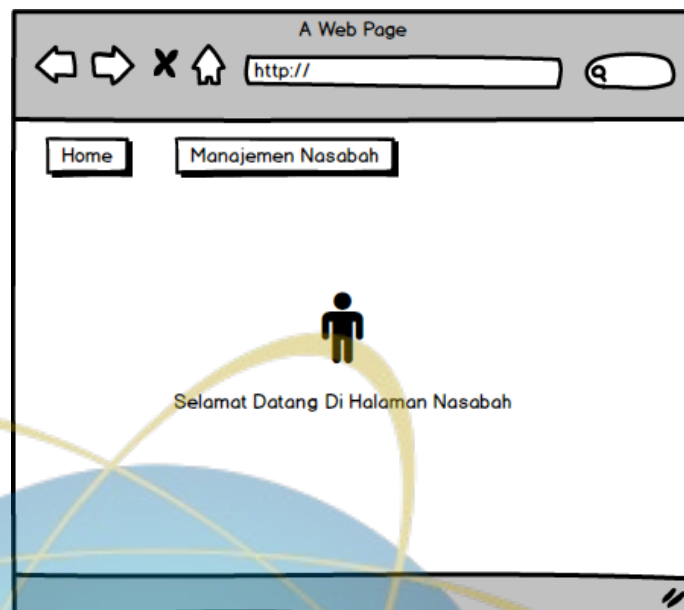
**Gambar 4.34** Halaman Debit Nasabah

Halaman ini merupakan halaman transaksi debit nasabah, halaman ini akan melakukan transaksi terhadap nasabah untuk dapat mencairkan uang dari hasil transaksi bank sampah.

**Gambar 4.35** Halaman Tambah Data Nasabah

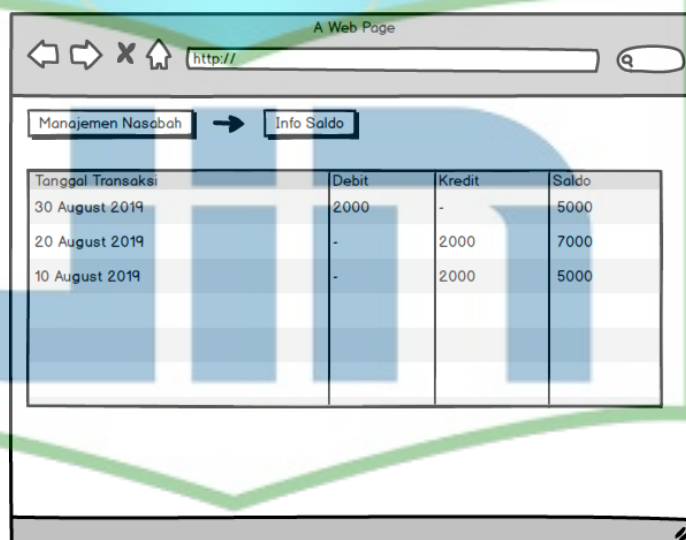
Halaman ini merupakan halaman tambah data nasabah, halaman ini akan melakukan input data nasabah baru yang dilakukan oleh admin.





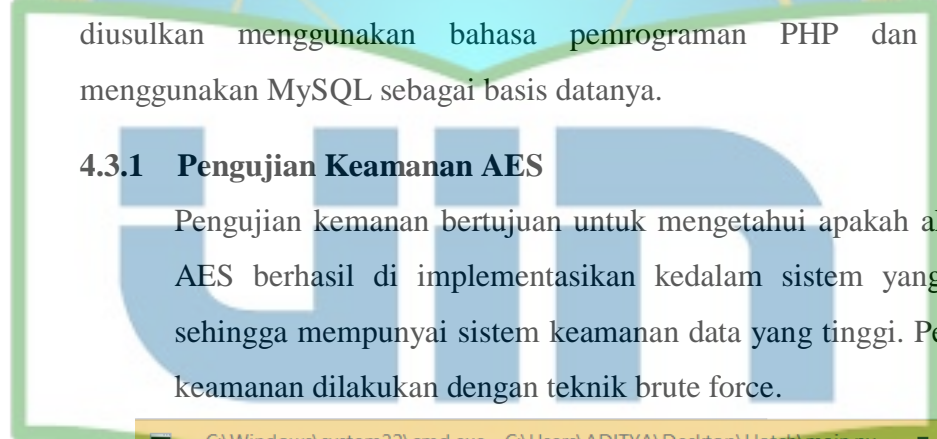
**Gambar 4.36** Halaman Nasabah

Halaman ini hanya dapat diakses oleh nasabah setelah melakukan login. Yang didalamnya terdapat menu info saldo dan data transaksi nasabah.



**Gambar 4.37** Halaman Info Saldo Nasabah

Halaman ini merupakan halaman info saldo nasabah, yang didalamnya menampilkan saldo atau tabungan dan pencairan uang yang dilakukan oleh nasabah



**Gambar 4.39** Cmd Brute Force

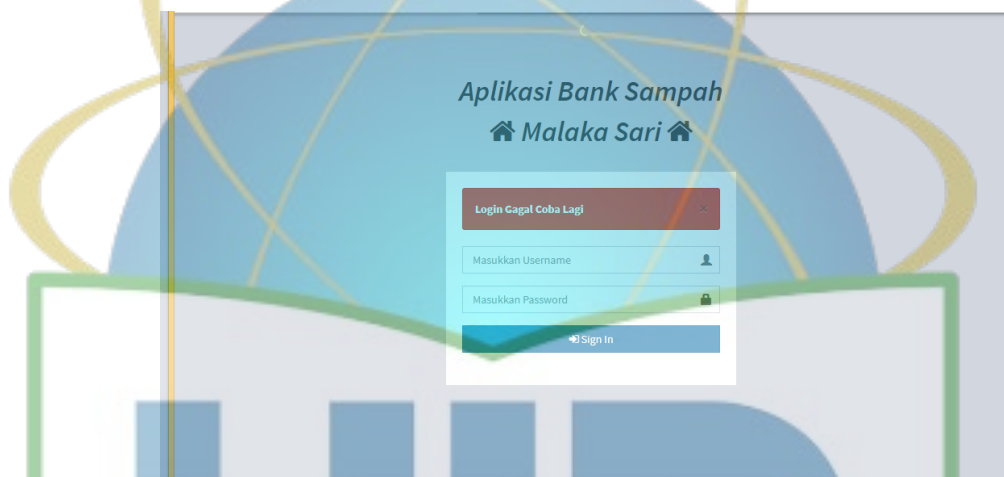
diusulkan menggunakan bahasa pemrograman PHP dan dengan menggunakan MySQL sebagai basis datanya.

### 4.3.1 Pengujian Keamanan AES

```
DevTools listening on ws://127.0.0.1:57629/devtools/browser/7a6b945a-c3e5-4086-9
e24-59d13fie2141
+ [1m+ [91m
[1m[37m[91m--> [92mU.1.0
+ [91m[37m[91m--> [92mncoded by Metachar
+ [91m[37m[91m--> [92m brute-force tool
+ [92m+ [1m
[~] + [37mEnter a website: http://localhost/$kripsi_tes3/Home_Page.php
+ [92m[~] + [37mChecking if site exists + [92m[OK] + [37m
+ [92m[~] + [37mEnter the username selector: body > div.login-box > div.login-box-
body > form > div:nth-child(1) > input
+ [92m[~] + [37mEnter the password selector: body > div.login-box > div.login-box-
body > form > div:nth-child(2) > input
+ [92m[~] + [37mEnter the Login button selector: body > div.login-box > div.login-
box-body > form > div.row > div > button
+ [92m[~] + [37mEnter the username to brute-force: pegawai3@gmail.com
+ [92m[~] + [37mEnter a directory to a password list: C:\Users\ADITYA\Desktop\Hate
h\passlist.txt
```

**UIN Syarif Hidayatullah Jakarta**

Ini merupakan tahap menggunakan brute force hatch, untuk menjalankan software ini perlu menggunakan command prompt (cmd) kemudian sesuaikan direktori tempat brute force hatch disimpan. Lalu ketik *python main.py*. setelah itu masukan target url yang ingin di tes, lalu masukkan type field untuk username,password dan button login. Kemudian input username yang ingin di tes dan pilih direktori untuk *passlist.txt*, kemudian enter.



**Gambar 4.40** Pengujian Sistem Aplikasi

Jika semua parameter di isi dengan benar, maka akan muncul pop up url target dari yang di input kan tadi. Pop up seperti gambar di atas, dan secara otomatis brute force akan menginput username dan password secara terus – menerus sampai username dan password ditemukan.



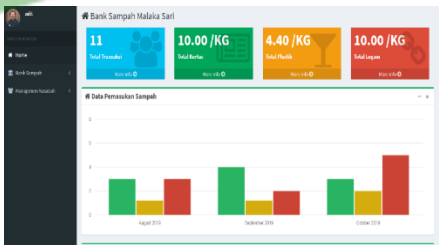
**Gambar 4.41** Serangan Brute Force

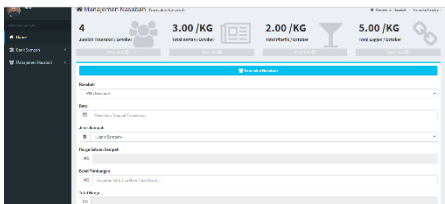
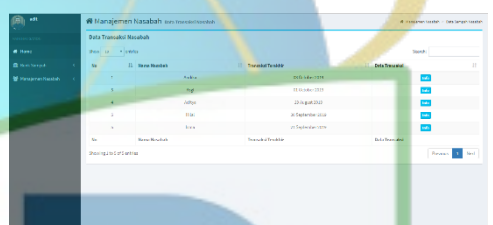
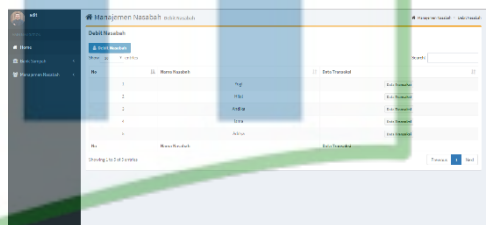
Setelah dilakukan pengujian dengan menggunakan brute force hatch, pada aplikasi bank sampah serangan brute force butuh waktu yang sangat lama dan tidak dapat diselesaikan untuk menembus keamanan yang sudah di enkripsi oleh AES.

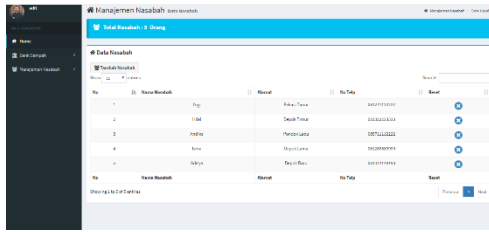
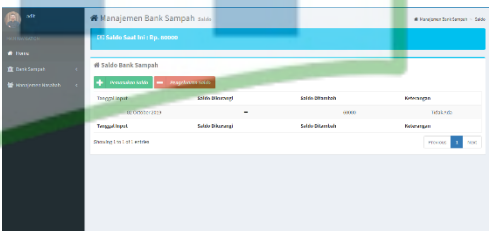
#### 4.3.2 Pengujian Sistem

Pengujian ini menggunakan *black-box testing* dengan melakukan *test-case* yaitu dengan masuk kedalam aplikasi dan memasukkan data dan selanjutnya melihat *output* (keluaran) apakah sesuai dengan yang diharapkan.

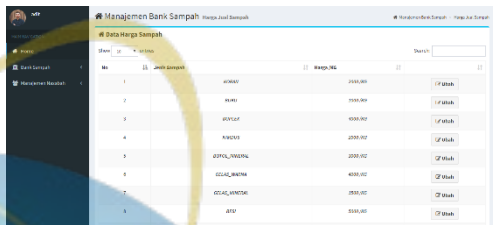

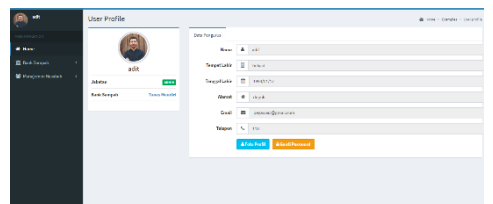
Table 4.23 Tabel Pengujian Level Admin

No	Bentuk Pengujian	Hasil yang Diharapkan	Hasil
1	Isi <i>username</i> & <i>password</i> (data benar)	Masuk halaman menu utama Admin	
2	Isi <i>username</i> & <i>password</i> (data salah)	Muncul peringatan <i>username</i> dan <i>password</i> salah	
3	Menu Home	-Menampilkan aktivitas terakhir yang dilakukan oleh aplikasi -menampilkan data transaksi bank sampah per bulan	

3	Pilih menu Manajemen Nasabah – Transaksi Nasabah	<ul style="list-style-type: none"> <li>- Menampilkan data transaksi per bulan</li> <li>- Input data nasabah, tanggal transaksi, jenis sampah, berat timbangan, kemudian data diproses dan tersimpan di dalam database</li> </ul>	
4	Pilih menu Manajemen Nasabah – Data Transaksi Nasabah	Menampilkan data nasabah yang sudah melakukan transaksi di dalam aplikasi	
5	Pilih menu Manajemen Nasabah – Debit Nasabah	<ul style="list-style-type: none"> <li>- Menampilkan data nasabah yang sudah melakukan penarikan uang</li> <li>- Input data nasabah, dan nominal uang, kemudian data diproses dan tersimpan di dalam database</li> </ul>	

6	Pilih menu Manajemen Nasabah – Data Nasabah	<ul style="list-style-type: none"> <li>- Menampilkan total jumlah nasabah</li> <li>- Menampilkan tabel data nasabah berupa foto, nama, alamat, dan no telepon secara keseluruhan</li> <li>- Input nama nasabah, no telepon dan alamat nasabah, kemudian data diproses dan tersimpan di dalam database</li> <li>- Mereset password nasabah</li> </ul>	
7	Pilih menu Bank Sampah – Saldo	<ul style="list-style-type: none"> <li>- Menampilkan jumlah saldo</li> <li>- Input nominal uang, tanggal transaksi, keterangan kemudian data diproses dan disimpan di dalam database</li> </ul>	



		- Menampilkan data transaksi saldo bank sampah secara keseluruhan	
8	Pilih menu Bank Sampah – Harga Jual Sampah	<ul style="list-style-type: none"> <li>- Menampilkan data harga jual sampah</li> <li>- Input harga jual baru untuk setiap jenis sampah, kemudian data diproses dan diperbaharui di dalam database</li> </ul>	
9	Pilih menu Bank Sampah – Riwayat Transaksi Nasabah	- Menampilkan data transaksi nasabah yang sudah melakukan transaksi di dalam aplikasi berdasarkan tanggal transaksi	
10	Pilih menu Profil	<p>Mengganti foto</p> <p>Mengganti password lama dengan password yang baru</p>	

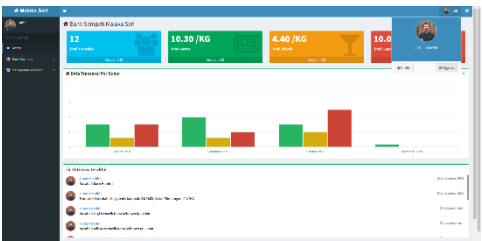


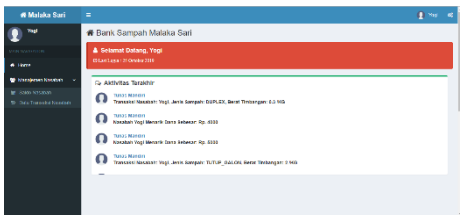
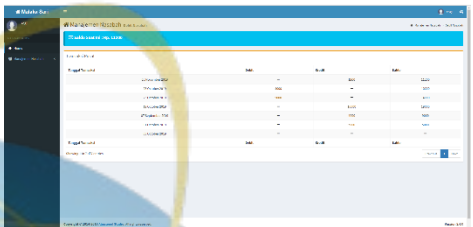
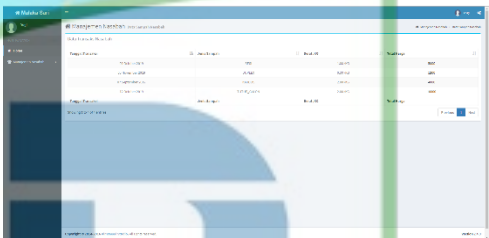
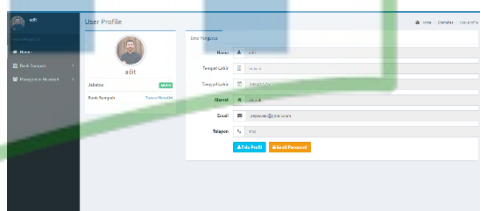
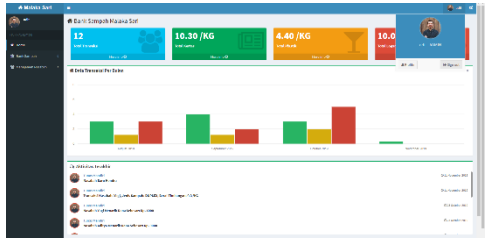
11	Pilih menu Sign Out	Keluar dari sistem aplikasi	
----	------------------------	--------------------------------	--

Table 4.24 Tabel Pengujian Level Nasabah

No	Bentuk Pengujian	Hasil yang Diharapkan	Hasil
1	Isi <i>username</i> & <i>password</i> (data benar)	Masuk halaman menu utama Admin	
2	Isi <i>username</i> & <i>password</i> (data salah)	Muncul peringatan <i>username</i> dan <i>password</i> salah	

3	Menu Home	-Menampilkan aktivitas terakhir yang dilakukan oleh nasabah	
4	Pilih menu Manajemen Nasabah – Saldo Nasabah	- Menampilkan data saldo nasabah - Menampilkan data transaksi debit dan kredit yang telah dilakukan oleh nasabah	
5	Pilih menu Manajemen Nasabah – Data Transaksi Nasabah	Menampilkan secara keseluruhan transaksi yang telah dilakukan oleh nasabah	
7	Pilih menu Profil	Mengganti foto Mengganti password lama dengan password yang baru	

8	Pilih menu Sign Out	Keluar dari sistem aplikasi	
---	---------------------	-----------------------------	--

### 4.3.3 Implementasi Perangkat

Perangkat keras yang digunakan untuk mendukung sistem ini minimal dengan spesifikasi sebagai berikut:

#### 1. Perangkat Keras

- Intel Core i5
- Ram 16 GB
- Vga Gtx 1060
- Hardisk Seagate 2TB
- 64-bit Operating System

#### 2. Perangkat Lunak

- MySQL versi 10.3.16
- XAMPP versi 3.2.4
- phpMyAdmin 7.1.

## BAB V

### HASIL dan PEMBAHASAN

#### 5.1 Hasil Output

Berikut merupakan printscreen program sebagai hasil output dan penjelasan dari aplikasi yang telah dibuat:

##### 5.1.1 Halaman Login



**Gambar 5.1** Halaman Login

Keterangan Gambar:

Halaman login, berisikan tampilan field username dan password untuk dapat masuk ke dalam sistem.

##### 5.1.2 Halaman Utama Admin

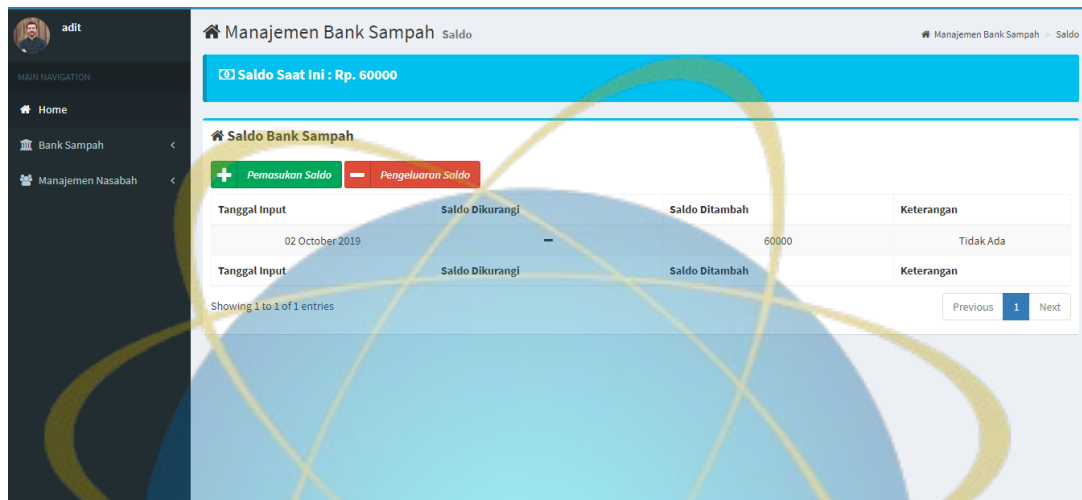


**Gambar 5.2** Halaman Utama Admin

Keterangan Gambar :

Halaman Utama Admin, Berisi Tampilan data pemasukan sampah per bulan dan total keseluruhan transaksi yang sudah di lakukan di dalam aplikasi.

### 5.1.3 Halaman Saldo Bank Sampah

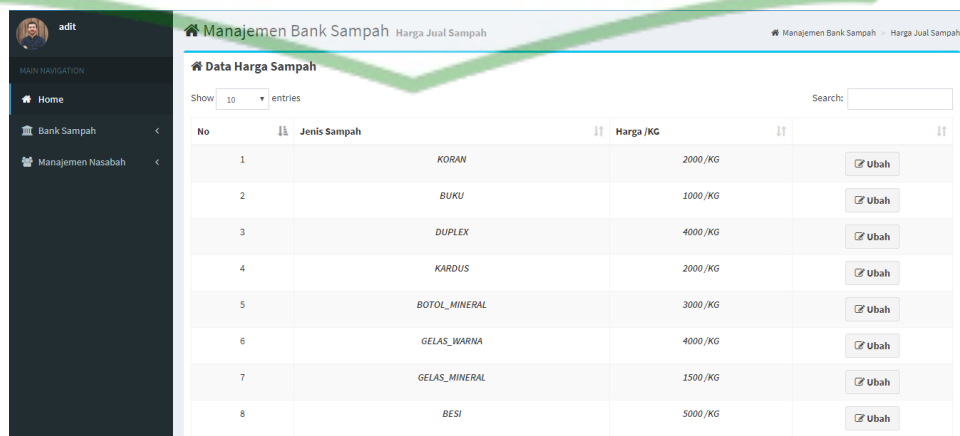


Gambar 5.3 Halaman Saldo Bank Sampah

Keterangan Gambar :

Halaman Saldo Bank Sampah. Halaman yang hanya dapat diakses oleh admin, berisi detail transaksi uang yang telah dilakukan oleh bank sampah, admin dapat melakukan pemasukan dan pengeluaran uang bank sampah.

### 5.1.4 Halaman Edit Harga Jual

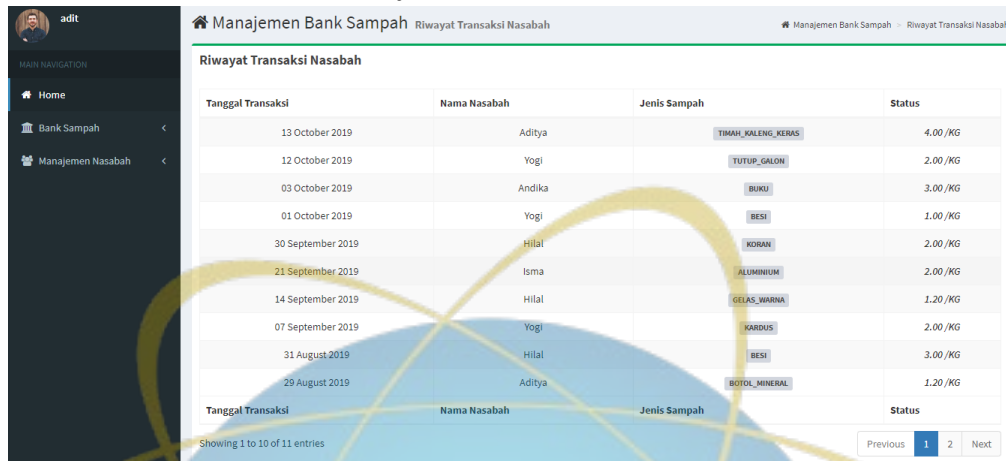


Keterangan Gambar 5.4 Halaman Edit Harga Jual



Halaman Edit Harga Jual. Halaman yang hanya dapat diakses oleh admin, admin dapat merubah harga jual sampah.

### 5.1.5 Halaman Riwayat Transaksi Nasabah



Tanggal Transaksi	Nama Nasabah	Jenis Sampah	Status
13 October 2019	Aditya	TIMAH, KALENG, KERAS	4.00 /KG
12 October 2019	Yogi	TUTUP, GALON	2.00 /KG
03 October 2019	Andika	BUKU	3.00 /KG
01 October 2019	Yogi	BESI	1.00 /KG
30 September 2019	Hilal	KORAN	2.00 /KG
21 September 2019	Ismi	ALUMINIUM	2.00 /KG
14 September 2019	Hilal	GEJAS, BAKAR	1.20 /KG
07 September 2019	Yogi	KARDUS	2.00 /KG
31 August 2019	Hilal	BESI	3.00 /KG
29 August 2019	Aditya	BOTOL, MINERAL	1.20 /KG

Gambar 5.5 Riwayat Transaksi Nasabah

Keterangan Gambar :

Halaman Riwayat Transaksi Nasabah, Halaman yang berisi riwayat data transaksi yang sudah dilakukan oleh nasabah.

### 5.1.6 Halaman Transaksi Nasabah



No	Nama Nasabah	Transaksi Terakhir	Data Transaksi
1	Andika	03 October 2019	<a href="#">Info</a>
3	Yogi	01 October 2019	<a href="#">Info</a>
4	Aditya	29 August 2019	<a href="#">Info</a>
5	Hilal	30 September 2019	<a href="#">Info</a>
5	Ismi	21 September 2019	<a href="#">Info</a>

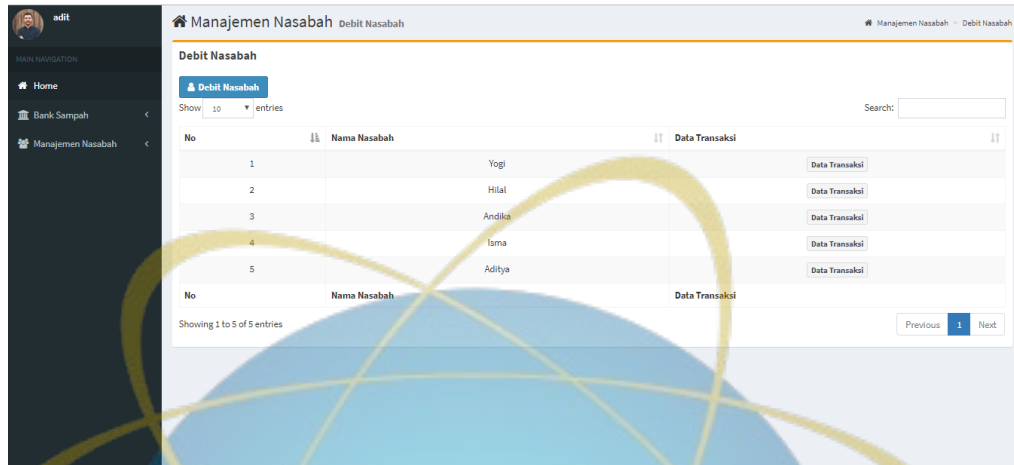
Gambar 5.6 Halaman Transaksi Nasabah

Keterangan Gambar :

Halaman Transaksi Nasabah. Halaman yang hanya dapat diakses oleh admin, berisi form yang harus di input yaitu data nasabah, tanggal transaksi, jenis sampah dan berat timbangan. Selain itu

admin dapat mengetahui total transaksi dan total pemasukan sampah.

### 5.1.7 Halaman Data Transaksi Nasabah



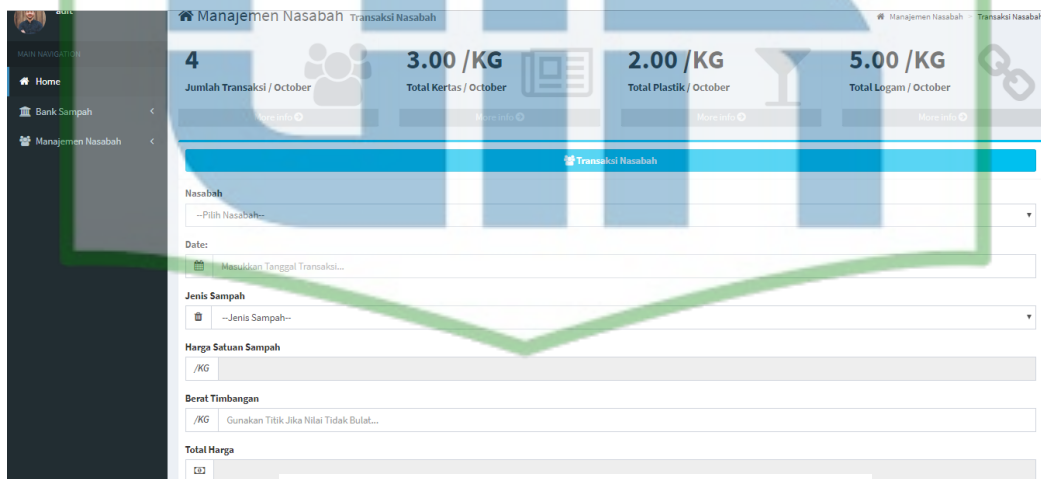
No	Nama Nasabah	Data Transaksi
1	Yogi	Data Transaksi
2	Hilal	Data Transaksi
3	Andika	Data Transaksi
4	Isma	Data Transaksi
5	Aditya	Data Transaksi

**Gambar 5.7** Halaman Data Transaksi Nasabah

Keterangan Gambar :

Halaman Data Transaksi Nasabah. Halaman yang hanya dapat diakses oleh admin, berisi detail transaksi dari masing – masing nasabah.

### 5.1.8 Halaman Debit Nasabah



4	3.00 /KG	2.00 /KG	5.00 /KG
Jumlah Transaksi / October	Total Kertas / October	Total Plastik / October	Total Logam / October

**Transaksi Nasabah**

Nasabah: --Pilih Nasabah--

Date: --Masukkan Tanggal Transaksi--

Jenis Sampah: --Jenis Sampah--

Harga Satuan Sampah: /KG

Berat Timbangan: /KG (Gunakan Titik Jika Nilai Tidak Bulat...)

Total Harga: Rp

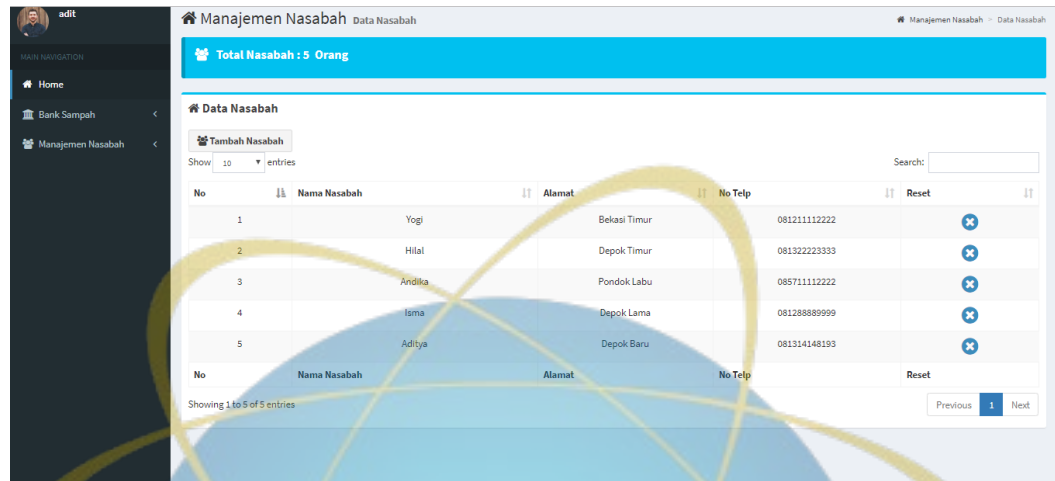
**Gambar 5.8** Halaman Debit Nasabah

Keterangan Gambar :

Halaman Debit Nasabah. Halaman yang hanya dapat diakses oleh admin, untuk melakukan proses debit admin harus menginput data

nasabah dan jumlah nominal penarikan. Selain itu halaman ini berisi detail transaksi debit yang telah dilakukan oleh nasabah.

#### 5.1.10 Halaman Tambah Nasabah

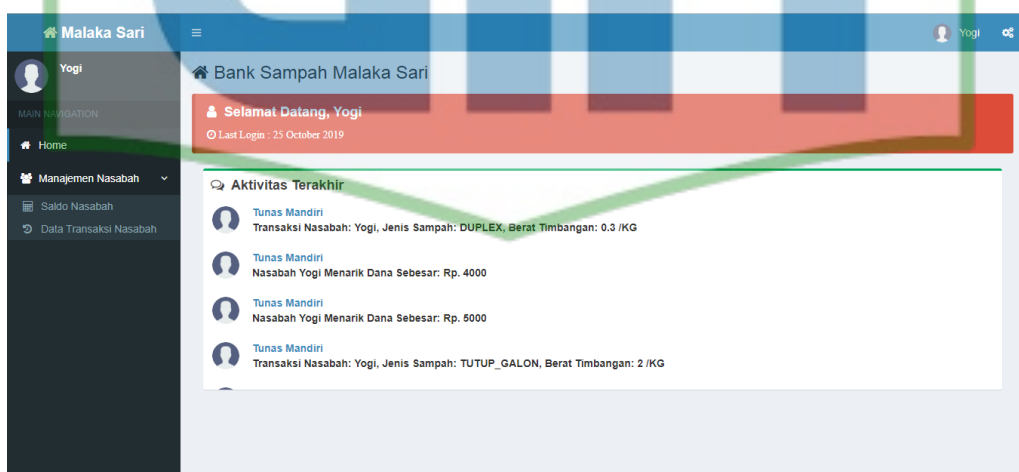


**Gambar 5.9** Halaman Tambah Nasabah

Keterangan Gambar :

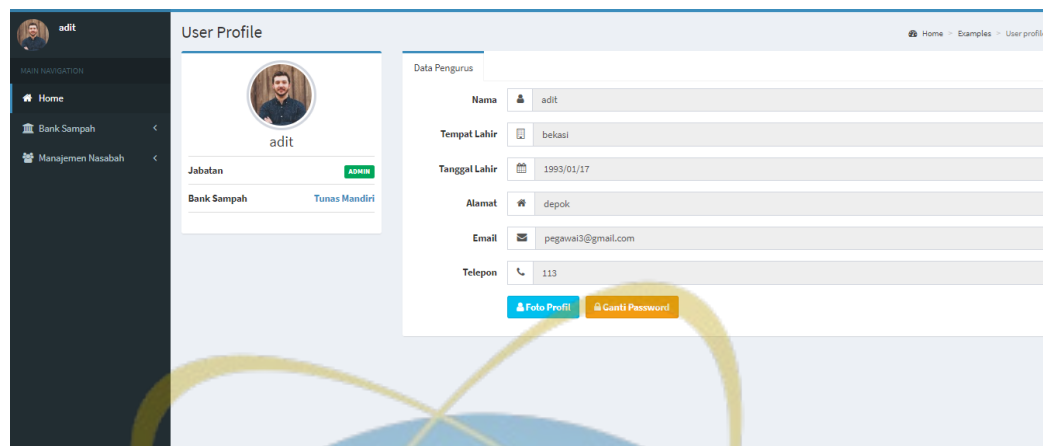
Halaman TambahNasabah. Halaman yang hanya dapat diakses oleh admin, berisi jumlah dan data nasabah serta reset password. Untuk melakukan proses tambah nasabah, admin harus menginput nama, no telepon yang nantinya dijadikan username untuk nasabah melakukan login dan alamat nasabah.

#### 5.1.10 Halaman Utama Nasabah



**Gambar 5.10** Halaman Utama Nasabah

### 6.1.10 Halaman Profil User



**Gambar 5.11** Halaman Profil User

Keterangan Gambar :

Halaman Profil, halaman ini dapat diakses baik nasabah maupun admin. Berisi data profile user, di halaman ini user dapat mengganti foto dan mengganti password.

## **BAB VI**

### **KESIMPULAN dan SARAN**

#### **6.1 Kesimpulan**

Berdasarkan hasil dan pembahasan untuk menjawab rumusan masalah yang ada di dalam penelitian ini, maka dapat ditarik beberapa kesimpulan sebagai berikut:

1. Algoritma AES dinyatakan aman dalam mengamankan data transaksi nasabah karena sulit untuk ditembus oleh serangan brute force dan juga memerlukan waktu yang sangat lama untuk menemukan kunci yang benar.
2. Proses enkripsi dan dekripsi pada algoritma AES dipengaruhi oleh panjang kunci. AES menetapkan panjang kunci adalah 128 bit, 192 bit, dan 256 bit, Dimana semakin panjang kunci yang digunakan maka akan semakin banyak putaran yang dilalui dan semakin lama proses enkripsi dan dekripsi berlangsung
3. Aplikasi Bank Sampah Malaka Sari mampu memberikan alternatif untuk mengelola data bank sampah dengan baik karena adanya integritas data antara bank sampah dengan nasabah.

#### **6.2 Saran**

Aplikasi yang penulis buat tentu saja masih belum sempurna, masih banyak hal yang dapat dikembangkan guna membuat manfaat aplikasi menjadi lebih baik lagi untuk kedepannya. Oleh karena itu penulis juga menyampaikan beberapa saran untuk pengembangan selanjutnya, yaitu:

1. Sistem nantinya dapat dikembangkan dengan menambahkan algoritma lainnya, tidak hanya menggunakan algoritma AES saja tetapi dapat menambahkan algoritma keamanan lainnya yang memiliki sistem pengamanan yang lebih sempurna.

2. Penelitian selanjutnya diharapkan dapat mengembangkan aplikasi menjadi lebih baik lagi sehingga aplikasi terlihat lebih dinamis dan moderen.
3. Penelitian selanjutnya diharapkan menggunakan metode pengembangan sistem lainnya seperti, waterfall, metode prototyping, dll.





## Daftar Pustaka

- Rinaldi Munir. (2019). Kriptografi Edisi Kedua.
- Budi Raharjo. (2018). Modul Pemrograman Web.
- Betha Sidik. (2017). Pemrograman Web dengan PHP 7.
- Kartika Iman Santoso, Wahyu Priyoatmoko, (2016). Pengamanan Data MySQL Pada E-Commerce Dengan Algoritma AES 256.
- Geby Geta Putri, Wiwin Styorini, Rizki Dian Rahayani, (2018), Analisis Kriptografi Simetris AES dan Kriptografi Asimetris RSA pada Enkripsi Citra Digital.
- Meilisa Dwiyantri Marali, Fajar Pradana, Bayu Priyambadha, (2018). Pengembangan Sistem Aplikasi Transaksi Bank Sampah Online Berbasis Web.
- Theresa Ayu, (2016). Implementasi Algoritma Rivest Shamir Adleman (RSA) dan Algoritma Knuth Morris Pratt (KMP) pada Aplikasi Klinik.
- Donzilo Antonio Meko, (2018). Perbandingan Algoritma DES, AES, IDEA, dan Blowfish dalam Enkripsi dan Dekripsi Data.
- Yayasan Unilever Indonesia, (2013). Buku Panduan Sistem Bank Sampah.
- Sumandri, (2017). Studi Model Algoritma Kriptografi Klasik dan Modern.
- Syamsinar, (2017). Implementasi Kombinasi Algoritma Asimetris Rivest Shamir Adleman Dan Algoritma Simetris Advanced Encryption Standard Pada Aplikasi Pesan Singkat.
- Rahmat Tullah, Muhammad Iqbal Dzulhaq, Yudi Setiawan, (2016). Perancangan Aplikasi Kriptografi File Dengan Metode Algoritma Advanced Encryption Standard (AES).
- Dwi Qunita Putri Ambeq Paramarta, Ari Kusyanti, Mahendra, (2018). Implementasi Algoritma Advance Encryption Standard (AES) pada Enkripsi

dan Dekripsi QR-Code.

Wayan Odiasa, (2015). Implementasi Algoritma Kriptografi Rijndael untuk Pengamanan Sistem Sms Banking dan Internet Banking.

Anih Sri Suryani, (2014). Peran Bank Sampah Dalam Efektivitas Pengelolaan Sampah.



## Lampiran

### Lampiran 1: Hasil Wawancara

#### 1. Identitas Responded

Tanggal : 18 Oktober 2018

Responden : Bpk. Prakoso

Jabatan : Pengurus Bank Sampah

Tempat : Bank Sampah Malaka Sari – Jakarta Timur

#### 2. Daftar Pertanyaan

1. Dalam satu rw mencakup berapa rt? Dan berapa total nasabah yang berperan dalam kegiatan bank sampah?

Jawaban: 1 Rw mencakup 18 RT, dan jumlah nasabah kurang lebih sudah mencapai 300 orang.

2. Dalam pengoperasian bank sampah saat ini, apakah sudah terintegrasi oleh sistem komputerisasi?

Jawaban: Belum, masih menggunakan buku besar dalam kegiatan transaksi.

3. Apakah mekanisme yang berjalan saat ini dirasa sudah cukup atau diperlukan peningkatan?

Jawaban: Perlu Peningkatan dalam system komputerisasi.

4. Apakah ada kendala selama ini dengan sistem yang sudah berjalan?

Jawaban: Ada, dalam pendataan bank sampah.

5. Setujukah anda jika dibuat sebuah sistem sebagai pusat transaksi untuk meningkatkan efektifitas?

Jawaban: Sangat Setuju.

**Lampiran 2: Source Code AES**

```

<?php
/**
Aes encryption
*/
class AES {

    protected $key;
    protected $data;
    protected $method;

    /**
     * Available OPENSSEL RAW DATA | OPENSSEL ZERO PADDING
     *
     * @var type $options
     */
    protected $options = 0;

    /**
     *
     * @param type $data
     * @param type $key
     * @param type $blockSize
     * @param type $mode
     */

    function __construct($data = null, $key = null, $blockSize = null, $mode = 'CBC') {
        $this->setData($data);
        $this->setKey($key);
    }

```

```

    $this->setMethod($blockSize, $mode);
}
/**
 *
 * @param type $data
 */
public function setData($data) {
    $this->data = $data;
}
/**
 *
 * @param type $key
 */
public function setKey($key) {
    $this->key = $key;
}
/**
 * CBC 128 192 256
 * CBC-HMAC-SHA1 128 256
 * CBC-HMAC-SHA256 128 256
 * CFB 128 192 256
 * CFB1 128 192 256
 * CFB8 128 192 256
 * CTR 128 192 256
 * ECB 128 192 256
 * OFB 128 192 256
 * XTS 128 256
 * @param type $blockSize
 * @param type $mode
 */

```

```

public function setMethod($blockSize, $mode = 'CBC') {
    if($blockSize==192 && in_array("", array('CBC-HMAC-SHA1','CBC-HMAC-
SHA256','XTS'))){
        $this->method=null;
        throw new Exception('Invalid block size and mode combination!');
    }
    $this->method = 'AES-' . $blockSize . '-' . $mode;
}
/**
 *
 * @return boolean
 */
public function validateParams() {
    if ($this->data != null &&
        $this->method != null ) {
        return true;
    } else {
        return FALSE;
    }
}
//it must be the same when you encrypt and decrypt
protected function getIV() {
    return '1234567890123456';
    //return mcrypt_create_iv(mcrypt_get_iv_size($this->cipher, $this->mode),
    MCRYPT_RAND);
    return openssl_random_pseudo_bytes(openssl_cipher_iv_length($this->method));
}
/**
 * @return type
 * @throws Exception

```



```

    */
    public function encrypt() {
        if ($this->validateParams()) {
            return trim(openssl_encrypt($this->data, $this->method, $this->key, $this->options, $this->getIV()));
        } else {
            throw new Exception('Invalid params!');
        }
    }
}

/**
 * @return type
 * @throws Exception
 */
public function decrypt() {
    if ($this->validateParams()) {
        $ret=openssl_decrypt($this->data, $this->method, $this->key, $this->options, $this->getIV());

        return trim($ret);
    } else {
        throw new Exception('Invalid params!');
    }
}
}
?>

```