Ubiquitous: Computers and its Applications Journal
Volume 4, Nomor 1, Juni 2021, 1-8

ISSN 2622-7983 online
ISSN 2622-7746 print

RANCANG BANGUN APLIKASI *SECURE MESSAGE* MENGGUNAKAN ALGORITMA *RIVEST CODE* 6 BERBASIS ANDROID

OKTAVIAN ANGGARA[1], **KHAIRIL ANAM**[2]

Teknik Informatika, Fakultas Teknik
Universitas Maarif Hasyim Latif, Sidoarjo, Indonesia
e-mail : [1]oktavian anggara14@gmail.com, [2]khairil_anam@dosen.umaha.ac.id

ABSTRAK

Komunikasi sangat diperlukan untuk mempermudah silaturahmi, untuk menjalin kerja sama dengan orang lain. Berkembangnya zaman dan perkembangan teknologi yang ada pada saat ini mendorong banyak orang untuk mampu menciptakan berbagai macam perangkat. Salah satunya perangkat yang bisa digunakan mengirim dan menerima dalam komunikasi atau informasi pada *smartphone,* untuk keamanan informasi maka dibutuhkan enkripsi menggunakan algoritma *rivest code 6.*

Kata kunci: android, *secure message,* sms, algoritma *rivest code* 6

PENDAHULUAN

Komunikasi butuh keamanan agar tidak dapat diketahui ataupun dibaca orang lain maka dibutuhkan enkripsi menggunakan algorima *rivest code* 6, enkripsi mengubah pesan asli menjadi kode atau *chipertext* untuk mengembalikan pesan ke bentuk asli atau *plaintext* maka diperlukan proses dekripsi

METODE PENELITIAN

Perancangan sistem dimulai dengan perancangan *flowchart, use case diagram, squence diagram, blok diagram,* perancangan dan desain aplikasi dimulai dengan perancangan detail dari halaman yang akan ditampilkan dalam aplikasi.

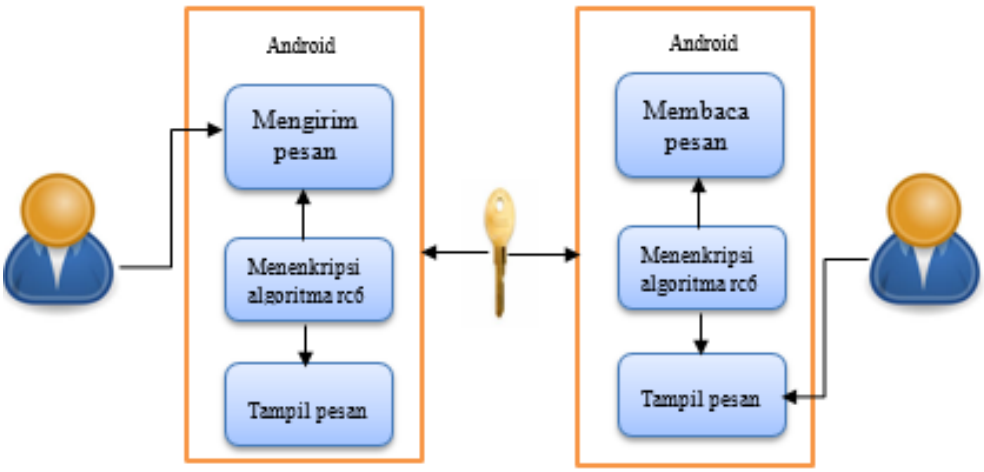
Blok diagram

Blok Diagram merupakan suatu peryataan gambar yang ringkas dari gabungan sebab dan akibat antara masukan dan keluaran dari suatu sistem untuk memperjelas konsep keseluruhan, penjelasan gambar blok diagram adalah:

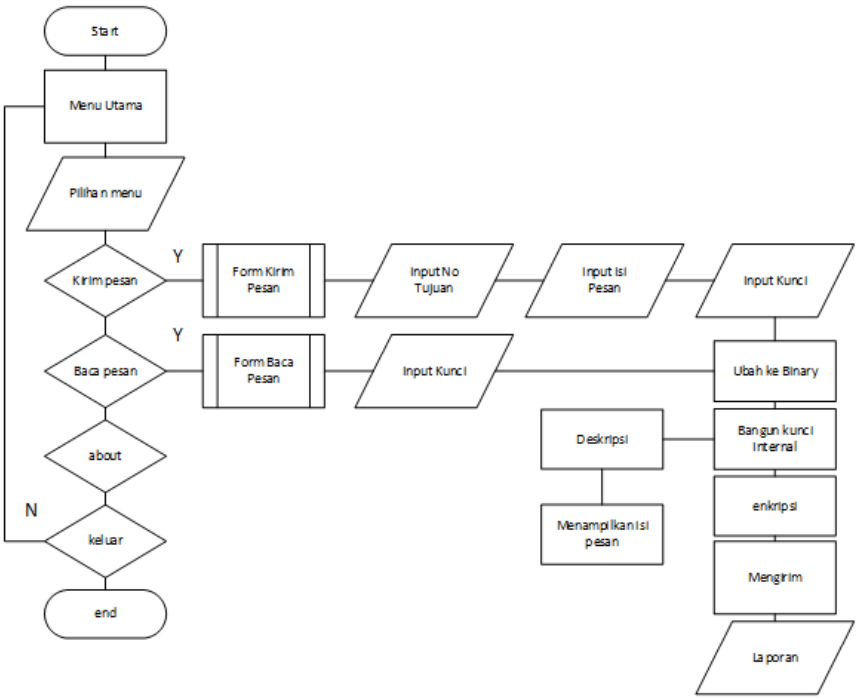
1. Pengirim memasukan kunci untuk mengenkripsi pesan yang akan dikirimkan kan.
2. Pesan yang sudah dikunci akan tampil sebelum dikirimkan.
3. Pengirim bisa memulai mengirimkan pesan kepada nomer yang dituju.
4. Pesan yang telah dikirim akan masuk dan tampil pada aplikasi penerima.
5. Penerima akan membaca pesan yang telah masuk dengan memasukan kunci untuk menenkripsi pesan menjadi pesan *plaintext*.

Flowchart

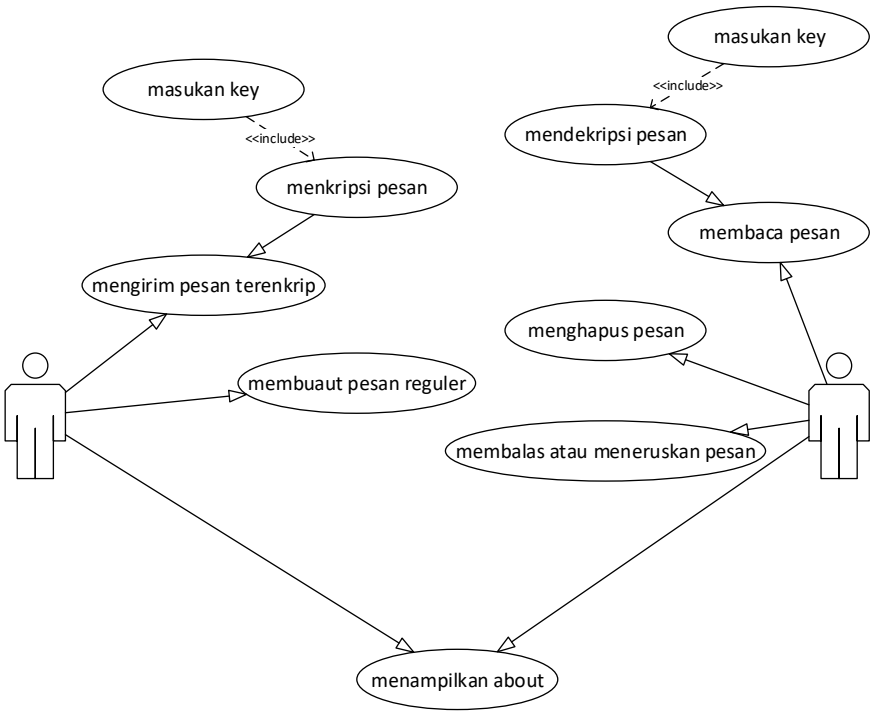
Flowchart merupakan serangkaian bagan-bagan yang menggambarkan alir program tentang urutan prosedur dalam *mobile application* enkripsi pesan menggunakan algoritma RC6.



Gambar 1. Blok Diagram



Gambar 2. *Flowchart*



Gambar 3. *Use Case Diagram*

Penjelasan gambar 2 sebagai berikut:

1. Pada menu utama aplikasi terdapat 3 pilihan menu diantaranya menu kirim pesan, menu baca pesan, menu about.
2. Pada menu kirim pesan akan muncul form buat kirim pesan *user* dapat memasukan nomer tujuan, menulis pesan yang akan dikirimkan, input kunci untuk menkripsi pesan yang akan dikirimkan
3. *User* dapat membaca pesan masuk pada menu baca pesan disini penerima tinggal memasukan kunci yang sama dengan pengirim dapat mengenkripsi pesan ke bentuk *plainteks*

Use case diagram

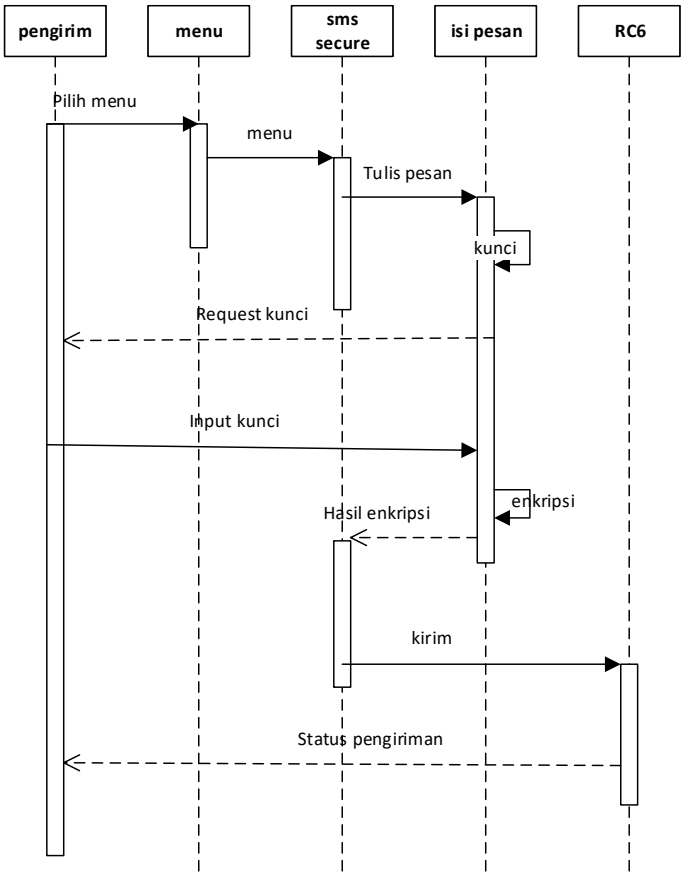
Use case diagram tidak menunjukan detail kasus penggunaan, ini hanya merangkum beberapa hubungan antara kasus penggunaan aktor dan sistem itu. Tidak menunjukan urutan langkah-langkah yang dilakukan untuk mencapai tujuan dari setiap kasus penggunaan. Penulis merancang aplikasi *secure message* yang sederhana agar

memudahkan *user* yang akan menggunakan aplikasi, pengirim dapat memasukan kunci, nomer telepon tujuan, menuliskan pesan sedangkan penerima juga sama peneima juga dapat membaca pesan yang masuk memasukan kunci dan membalas pesan.

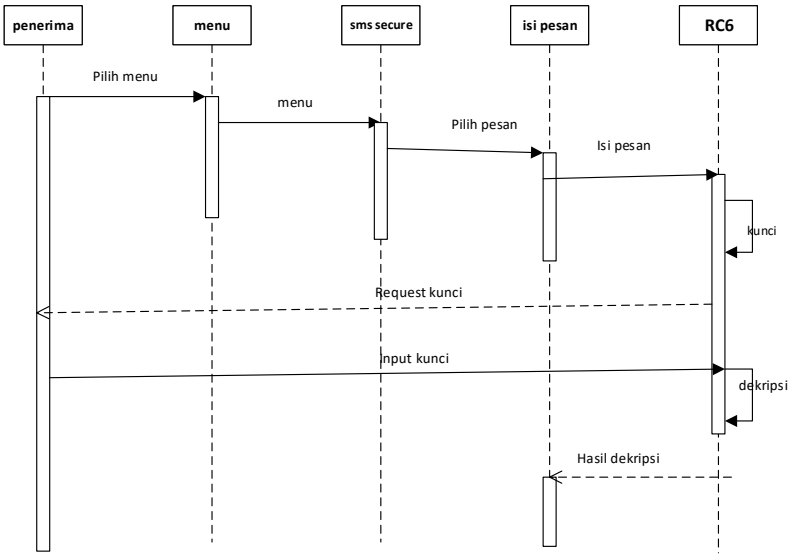
Squence diagram

Sequence diagram kadang disebut diagram acara atau skenario acara. Diagram urutan menunjukan sebagai garis-garis paralel berbagai proses atau objek yang hidup secara bersamaan dan sebagai panah horisontal pesan dipertukarkan di antara mereka dalam urutan yang terjadi. Dalam tahap rancang *squence diagram* ini apa saja yang dapat dikerjakan oleh sistem terhadap aplikasi yang dibuat.

Kebalikan proses diatas berikut ini adalah *squence diagram dekripsi* pesan yang akan diterapkan dalam aplikasi *secure message.*



Gambar 4. Diagram Squence Kirim Sms Enkripsi



Gambar 5. *Diagram squence* baca sms dekripsi

Sistem akan menampilkan pilihan menu yang ada pada aplikasi kemudian yang dipilih oleh user akan dimunculkan oleh sistem selanjutnya sistem akan meminta user buat masukan kunci yang berfungsi untuk membuka pesan yang telah dikirimkan

HASIL DAN PEMBAHASAN

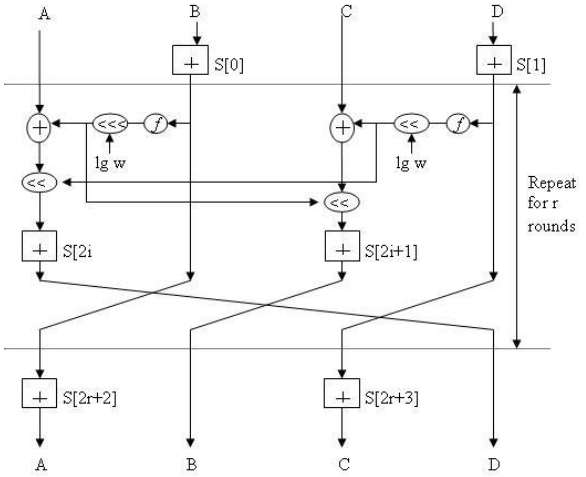
Penulisan Persamaan Algoritma *Rivest Code 6*

Rivest code 6 adalah pengembangan dari sistem *rivest code 5,* algoritma menggunakan ukuran blok hinga 128 bit dengan ukuran kunci yang digunakan bervariasi antara 128, 192 dan 256 bit, algoritma dilengkapi dengan beberapa parameter w/r/b, "w" merupakan ukuran kata dalam satuan bit, parameter "r" bilangan bukan negatif yang menunjukan banyaknya iterasi, parameter "b" menunjukan ukuran kunci enkripsi

Tabel 1. Aturan Operasi Dasar

No.	Operasi	Deskripsi
1	a + b	operasi penjumlahan bilangan integer
2	a – b	operasi pengurangan bilangan integer
3	a ⊕ b	operasi exclusive or (xor)
4	a × b	operasi perkalian bilangan integer
5	a <<< b	a dirotasikan ke kiri sebanyak variabel kedua b
6	a >>> b	a dirotasikan ke kanan sebanyak variabel kedua b

Dalam prosesnya akan didapatkan (A,B,C,D)=(B,C,D,A) yang dapat diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register di sisi kiri,

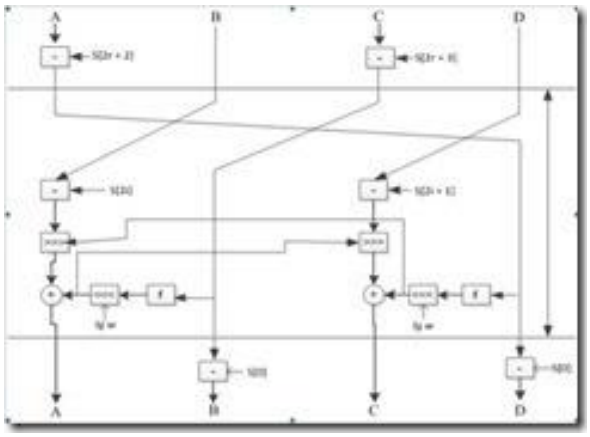


Gambar 6. Algoritma Enkripsi *Rivest Code* 6

Proses enkripsi dengan *rivest code 6* dijelaskan pada notasi (A,B,C,D)=(B,C,D,A) Algoritma adalah sebagai berikut:

```
B = B+S(0);
D = D+S(1);
For i = i to r do
t = (B x (2 x B + 1))<<<5
u = (D x (2 x D + 1))<<<5
A = ((A xor t)<<<u) + S (2 x i);
C = ((C xor u)<<<t) + S (2 x i+1);
(A,B,C,D) = (B.C,D,A)
A = A + S [(2 x r) + 2];
C = C + S [(2 x r) + 3];
```

Kebalikan dari proses enkripsi yaitu proses dekripsi seperti yang terllihat pada gambar 4.2 algoritma dekripsi *rivest code 6*.



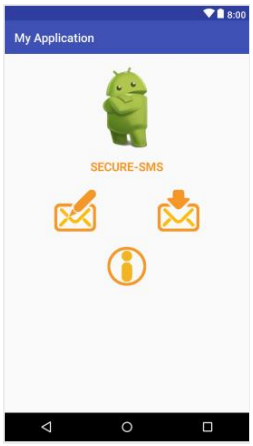
Gambar 7. Algoritma *Rivest Code 6*

Sama seperti proses enkripsi proses dekripsi dengan *rivest code 6* dijelaskan pada notasi (A,B,C,D)=(D,A,B,C) Algoritma adalah sebagai berikut:

```
C = C – S [2r + 3]
A = A – S[2r + 2]
For i = r down to 1 do
(A,B,C,D) = (D,A,B,C)
u = (D x (2D + 1))<<<5
t= (B x (2B + 1))<<<5
C = ((C – S[2i + 1]>>>t) xor u
A = ((A – S[2i])>>>u) xor t
D = D – S[1]
B = B – S[0]
```

Layout utama

Layout menurut bahasa memiliki arti tata letak. Di layout utama ini memiliki 3 menu yang dapat digunakan oleh *user*, menu kirim pesan, menu baca pesan, menu *about.*



Gambar 8. Gambar Layout Utama

Menu kirim pesan

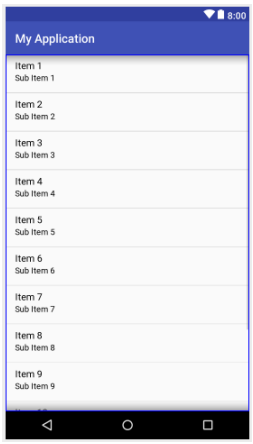
Pada tampilan utama aplikasi *secure message* terdapat salah satu menu yaitu menu kirim pesan yang memiliki bentuk gambar kotak

pesan dengan pensil, di dalam menu kirim pesan terdapat nomor tujuan, input kunci, input pesan, tombol gembok, hasil pesan dan tombol kirim pesan.

Gambar 9. Menu Kirim Pesan

Menu baca pesan

Menu baca pesan berfungsi untuk menerima dan menampilkan pesan yang telah dikirimkan oleh pengirim, menu baca pesan yang ditampilkan pada layout utama *secure massage* memiliki bentuk gambar pesan masuk atau kotak pesan berpanah ke bawah. Di sini seluruh data atau pesan yang masuk akan ditampilkan terlihat seperti pada gambar 10.



Gambar 10. gambar *inbox* (kotak masuk)

Setelah memilih pesan yang akan atau ingi dibaca atau dibuka aplikasi akan menampilkan seperti pada gambar 11 menu baca pesan.

Disini pesan yang masuk akan muncul pada kolom pesan dan nomor pengirim akan muncul pada kolom nomor pengirim *user* memasukan kunci yang sama dengan pengirim untuk dapat membuka pesan atau mengubah pesan menjadi pesan asli atau ke bentuk *plaintext*.

Menu about

Menu about berisikan penjelesan singkat tentang informasi atau profil pembuat aplikasi sehingga dengan adanya informasi singkat mengenai latar belakang pembuat aplikasi diharapkan *user* lebih mudah mengetahui informasi pembuat aplikasi, menu about pada aplikasi ini memiliki gambar lingkaran yang ditengah-tengahnya ada tulisan huruf "i" pada layout utama, terlihat seperti gambar 12.

Gambar 11. menu baca pesan

Gambar 12. menu about

Uji coba

Dalam tahapan selanjutnya yaitu tahap uji coba, uji coba ini adalah tahapan terakhir setelah melakukan perencanaan, untuk uji coba layout utama tetap terlihat seperti pada gambar 8 layout utama. Setelah *user* memilih menu kirim pesan pada layout utama akan terlihat seperti gambar 13 uji coba kirim pesan.

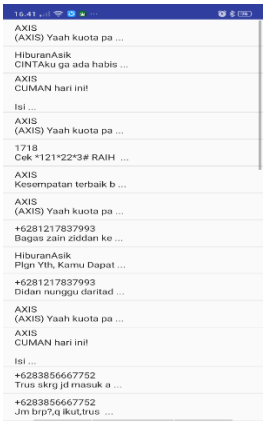
Disini *user* akan mengirimkan pesan pada nomer "083856667753" dengan memasukan kata kunci sebagai keamanan pesan "suwer" setelah nomer tujuan dan kunci sudah dimasukan kemudian dilanjutkan dengan menginputkan pesan yang mau dikirimkan dengan pesan "ujicoba123" kemudian mengklik tombol gembok untuk mengubah pesan asli menjadi pesan dekripsi, pesan yang sudah terkunci akan muncul pada kolom hasil "2545E05258BDC3F26DF2A2CE26DA778E"

kemudian mengklik tombol kirim untuk mengirimkan pesan.

Gambar 13. uji coba kirim pesan

Baca pesan

Uji coba dilanjutkan dengan menu baca pesan, pesan yang te;lah dikirim akan muncul pada menu ini, seperti terlihat pada gambar 14.



Gambar 14. inbox (kotak masuk)

Pilih atau buka pesan dengan nomer pengirim “085706243452” ini adalah nomer pengirim pesan pada uji coba kirim pesan seperti pada gambar 15 uji coba kirim pesan, setelah buka pesan akan muncul seperti pada gambar 15 uji coba baca pesan.

Pada menu ini akan muncul nomer pengirim “085706243452” dan isi pesan yang sudah terkunci atau terkode “2545E05258BDC3F26DF2A2CE26DA778E” penerima tinggal menginput kunci yang sama dengan pengirim dengan kata “suwer” kemudian dilanjutkan dengan mengklik tombol gembok untuk membuka kunci atau mengubah pesan kembali menjadi pesan dalam bentuk plaintext atau pesan asli, hasil pesan yang sudah dienkripsi akan muncul pada hasil “ujicoba123”.

Uji coba menu about

Uji coba ini mungkin masih sama dengan gambar 12 menu about dengan nama aplikasi “secure message” nama pembuat aplikasi “oktavian anggara” dengan nim”143215512” dan asal kampus “teknik informatika universitas maarif hasyim latief sidoarjo 2019”

Gambar 15. uji coba baca pesan

PENUTUP

Berdasarkan atau dari hasil pembahasan maka dapat ditarik beberapa kesimpulan aplikasi ini menggunakan kemanan dengan metode algoritma rivest code 6 dan dapat mengirim data atau sms berupa text jadi aplikasi ini menggunakan pulsa.

Saran yang diberikan pada pembahasan untuk aplikasi ini adalah selanjutnya diharapkan dapat menggunakan jaringan internet dan juga jika mengirimkan pesan ketika memasukan nomer secara otomatis dengan cara mengambil dari kontak akan muncul nama kontak tersebut.

DAFTAR PUSTAKA

[1] A. Ihyaulumiddin, “RANCANG BANGUN APLIKASI SHORT MESSAGE SERVICE (SMS) MENGGUNAKANALGORITMA RSA-RC6 BERBASIS ANDROID,” *eProceeding TIK*, vol. 1, no. 1, 2021.

[2] Y. Yusfrizal, “Rancang Bangun Aplikasi Kriptografi Pada Teks Menggunakan Metode Reverse Chiper Dan Rsa Berbasis Android,” *JTIK (Jurnal Tek. Inform. Kaputama)*, vol. 3, no. 2, pp. 29–37, 2019.

[3] N. Sianturi and K. Puspita, “keamanan source code java script menggunakan metode stream cipher dengan verifikasi md5,” *J. Mhs. Fak. Tek. dan Ilmu Komput.*, vol. 1, no. 1, pp. 583–594, 2020.

[4] D. P. Manurung, R. Puspasari, and W. Verina, “Perbandingan Metode Stream Dengan Metode Caesar Cipher Terhadap Pengiriman Pesan Pada Jaringan Wireless LAN,” *J. Mhs. Fak. Tek. dan Ilmu Komput.*, vol. 1, no. 1, pp. 332–342,

2020.
[5] K. M. A. Hakim, "RANCANG BANGUN APLIKASI ENKRIPSI-DEKRIPSI SMS PADA ANDROID DENGAN METODE RC6," *J. Tera*, vol. 1, no. 2, pp. 241–252, 2021.
[6] A. Kadir, *Algoritma & Pemrograman Menggunakan Java*. Yogyakarta: Andi, 2012.
[7] A. Nugroho, *Algoritma dan Struktur Data dalam Bahasa Java*. Yogyakarta: Andi, 2008.
[8] R. M. Yunus, H. Sujadi, and Karnia, "SISTEM KEAMANAN PESAN DENGAN ALGORITMA RIVEST CODE 6 (RC-6) MENGGUNAKAN JAVA PADA SMARTPHONE BERBASIS ANDROID," *J-ENSITEC*, vol. 2, no. 01, Nov. 2015.
[9] Wahana Komputer Semarang, *MEMAHAMI MODEL ENKRIPSI & SECURITY DATA*. Yogyakarta: Andi, 2003.
[10] W. P. Atmojo, R. R. Isnanto, and R. Kridalukmana, "Implementasi Aplikasi Kriptografi Pada Layanan Pesan Singkat (SMS) Menggunakan Algoritma RC6 Berbasis Android," *J. Teknol. dan Sist. Komput.*, vol. 4, no. 3, pp. 450–453, 2016.