BAB III
ANALISIS DAN PERANCANGAN

III.1. Analisis Masalah

Google Drive adalah layanan penyimpanan daring milik *Google* yang diluncurkan pada 24 April 2012. Layanan ini merupakan ekstensi dari *Google* Docs dan akan mengganti URL docs.*google*.com dengan *drive*.*google*.com setelah diaktifkan. *Google Drive* memberikan layanan penyimpanan gratis sebesar 15 GB dan dapat ditambahkan dengan pembayaran tertentu. Analisa sistem pada yang berjalan bertujuan untuk mengidentifikasi serta melakukan evaluasi terhadap *Perancangan Aplikasi Pengamanan Data Pada Google Drive Dengan Menggunakan Algoritma TEA*. Adapun masalah yang terdapat pada sistem yang telah lama adalah tidak adanya sistem keamanan data yang aman pada penyimpanan data di *google drive* dan belum berkembangnya algoritma Kriptografi TEA (*Tiny Encryption Algorithm*) dalam sistem keamanan *google drive*.

III.1.1. Strategi Pemecahan Masalah

Sistem yang sedang berjalan saat ini masih terdapat beberapa kekurangan yang terdapat pada sistem yang telah ada sebelumnya, berikut adalah kekurangan pada sistem yang telah berjalan :

1. Merancang sebuah aplikasi *google drive* dengan memanfaatkan sistem keamanan data yang dapat menjaga kerahasiaan dan keamanan pengiriman data pada aplikasi *google drive*.

2. Merancang dan membangun sebuah aplikasi *google drive* dengan menggunakan Kriptografi TEA (*Tiny Encryption Algorithm*).

III.1.2. Analisa Kebutuhan NonFungsional

Kebutuhan NonFungsional yang dibutuhkan dalam mengakses sistem adalah sebagai berikut :

1. PC atau Notebook Core 2
2. Javascript
3. Netbeans

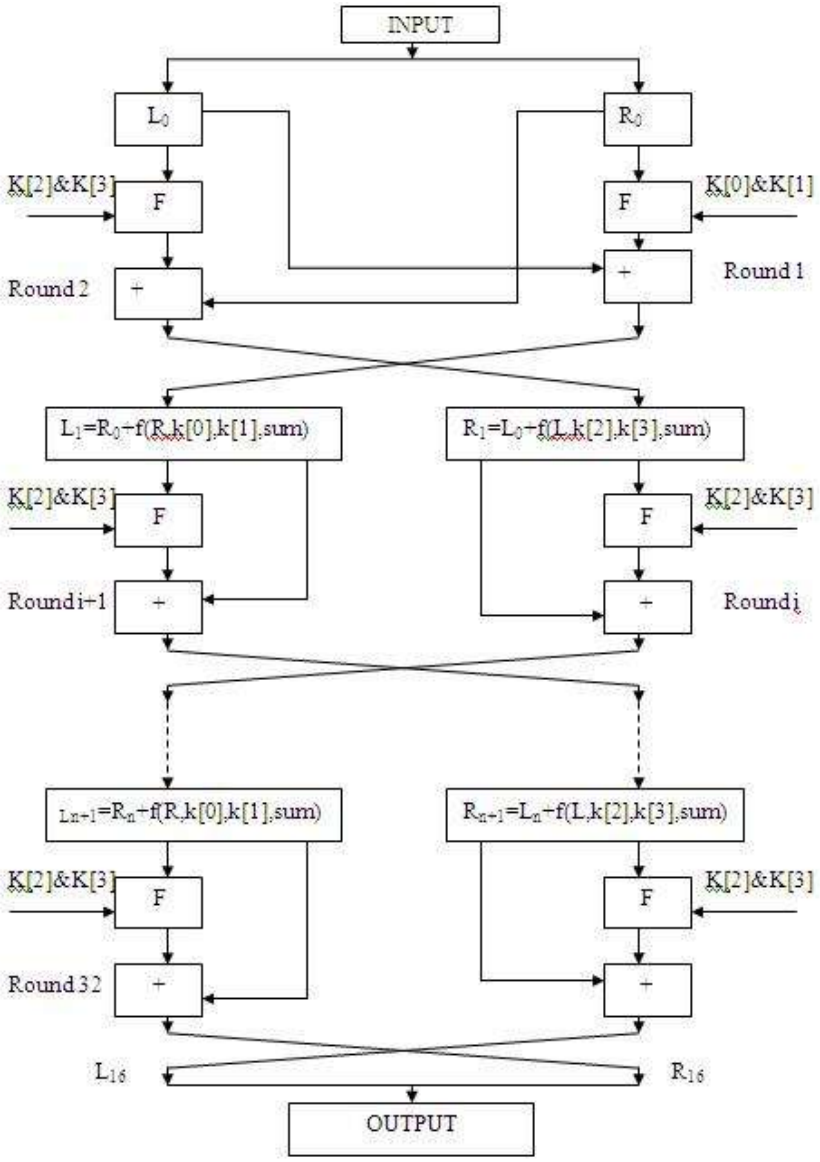
III.2. Penerapan Algoritma TEA

Tiny Encryption Algorithm (TEA) merupakan suatu algoritma sandi yang diciptakan oleh David J. Wheeler dan Roger M. Needham dari Cambridge University tahun 1994. Algoritma ini merupakan algoritma penyandian block cipher yang dirancang untuk penggunaan memori yang seminimal mungkin dengan kecepatan proses yang maksimal.

Sistem penyandian TEA menggunakan proses feistel network dengan menambahkan fungsi matematik berupa penambahan dan pengurangan sebagai operator pembalik selain XOR. Hal ini dimaksudkan untuk menciptakan sifat non-linearitas. Pergeseran dua arah (ke kiri dan ke kanan) menyebabkan semua bit kunci dan data bercampur secara berulang ulang.

TEA memproses 64-bit input sekali waktu dan menghasilkan 64- bit output. TEA menyimpan 64-bit input kedalam L0 dan R0 masing masing 32-bit, sedangkan 128-bit kunci disimpan kedalam k[0], k[1], k[2], dan k[3] yang masing masing berisi 32-bit. Diharapkan teknik ini cukup dapat mencegah penggunaan teknik exshautive search secara efektif. Hasil

outputnya akan disimpan dalam L16dan R16. Bilangan delta konstan yang digunakan adalah 9E3779B9, dimana bilangan delta berasal dari golden number, digunakan delta = (√5 -1)231 . Suatu bilangan delta ganda yang berbeda digunakan dalam setiap roundnya sehingga tidak ada bit dari perkalian yang tidak berubah secara teratur. Berbeda dengan struktur feistel yang semula hanya mengoperasikan satu sisi yaitu sisi sebelah kanan dengan sebuah fungsi F, pada algoritma TEA kedua sisi dioperasikan dengan sebuah fungsi yang sama.



Gambar III.1. Enkripsi Data
(Sumber : Mukti Qamal ; 2014 : 22)

Untuk melakukan enkripsi, Proses diawali dengan *input-bit* teks sebanyak 64-bit, kemudian 64-bit teks tersebut dibagi menjadi dua bagian, yaitu sisi kiri (L0) sebanyak 32-bit dan sisi kanan (R0) sebanyak 32-bit. Setiap bagian teks akan dioperasikan sendiri-sendiri. R0 (Z) akan digeser kekiri sebanyak empat (4) kali dan ditambahkan dengan kunci k[0], sementara itu Z ditambah dengan sum (delta) yang merupakan konstanta. Hasil penambahan ini di-XOR-kan dengan penambahan sebelumnya. Langkah selanjutnya di-XOR-kan dengan hasil penambahan antara Z yang digeser kekanan sebanyak lima (5) kali dengan kunci k[1]. Hasil tersebut kemudian ditambahkan dengan L0 (Y) yang akan menjadi R1 (Mukti Qamal ; 2014 : 22).

Teknik dekripsi, merupakan kebalikan dari enkripsi, pesan yang telah dienkripsi dikembalikan ke bentuk asalnya (*plaintext*) disebut dengan dekripsi pesan. Algoritma yang digunakan untuk dekripsi tentu berbeda dengan yang digunakan untuk enkripsi (Yoga Aprianto ; 2012 : 3).

III.2.1. Studi Kasus Algoritma TEA

Google ID Budi = 100013143816876
password = 100013143816876+100012366030312= 200025509847188

Tea.encrypt('Halo',password);

Plaintext: Halo
Long(64Bit) = K1 + K2<<8 + K3<<16 + K4<<24
Long(64Bit) = 72 + 97<<8 + 108<<16 + 111<<24 = 1869373768
Plaintext: 200025509847188
Long(64Bit) = K1 + K2<<8 + K3<<16 + K4<<24
Long(64Bit) = 50 + 48<<8 + 48<<16 + 48<<24 = 808464434
Plaintext: 200025509847188
Long(64Bit) = K1 + K2<<8 + K3<<16 + K4<<24
Long(64Bit) = 50 + 53<<8 + 53<<16 + 48<<24 = 808793394
Plaintext: 200025509847188
Long(64Bit) = K1 + K2<<8 + K3<<16 + K4<<24
Long(64Bit) = 57 + 56<<8 + 52<<16 + 55<<24 = 926169145
Plaintext: 200025509847188

Long(64Bit) = K1 + K2<<8 + K3<<16 + K4<<24
Long(64Bit) = 49 + 56<<8 + 56<<16 + NaN<<24 = 3684401
n = 2 z = 0 y = v[0] = 1869373768 delta = 2654435769
Sum = sum + delta = 2654435769
e = sum >>> 2 & 3 = 2654435769 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 0
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (0 >>> 5 ^ 0 << 2) + (0 >>> 3 ^ 0 << 4) ^ (2654435769 ^ 0) + (926169145 ^ 0)
mx= -714362382
z = v[0] + mx = 1155011386 + -714362382 = 1155011386

y = v[2 mod n] = v[0] = 1155011386
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1155011386 >>> 5 ^ 1155011386 << 2) + (1155011386 >>> 3 ^ 1155011386 << 4) ^ (2654435769 ^ 1155011386) + (3684401 ^ 1155011386)
mx= 1236216758
z = v[1] + mx = 1236216758 + 1236216758 = 1236216758

Sum = sum + delta = 5308871538
e = sum >>> 2 & 3 = 5308871538 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 1236216758
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1236216758 >>> 5 ^ 1236216758 << 2) + (1236216758 >>> 3 ^ 1236216758 << 4) ^ (5308871538 ^ 1236216758) + (808464434 ^ 1236216758)
mx= 1473888723
z = v[0] + mx = 2628900109 + 1473888723 = 2628900109

y = v[2 mod n] = v[0] = 2628900109
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (2628900109 >>> 5 ^ 2628900109 << 2) + (2628900109 >>> 3 ^ 2628900109 << 4) ^ (5308871538 ^ 2628900109) + (808793394 ^ 2628900109)
mx= 63587251
z = v[1] + mx = 1299804009 + 63587251 = 1299804009

Sum = sum + delta = 7963307307
e = sum >>> 2 & 3 = 7963307307 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 1299804009
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1299804009 >>> 5 ^ 1299804009 << 2) + (1299804009 >>> 3 ^ 1299804009 << 4) ^ (7963307307 ^ 1299804009) + (926169145 ^ 1299804009)
mx= 132385742
z = v[0] + mx = 2761285851 + 132385742 = 2761285851

y = v[2 mod n] = v[0] = 2761285851
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)

mx = (2761285851 >>> 5 ^ 2761285851 << 2) + (2761285851 >>> 3 ^ 2761285851 << 4) ^ (7963307307 ^ 2761285851) + (3684401 ^ 2761285851)
mx= -673269553
z = v[1] + mx = 626534456 + -673269553 = 626534456

Sum = sum + delta = 10617743076
e = sum >>> 2 & 3 = 10617743076 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = 626534456
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (626534456 >>> 5 ^ 626534456 << 2) + (626534456 >>> 3 ^ 626534456 << 4) ^ (10617743076 ^ 626534456) + (808793394 ^ 626534456)
mx= -1753152562
z = v[0] + mx = 1008133289 + -1753152562 = 1008133289

y = v[2 mod n] = v[0] = 1008133289
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1008133289 >>> 5 ^ 1008133289 << 2) + (1008133289 >>> 3 ^ 1008133289 << 4) ^ (10617743076 ^ 1008133289) + (808464434 ^ 1008133289)
mx= -396989074
z = v[1] + mx = 229545382 + -396989074 = 229545382

Sum = sum + delta = 13272178845
e = sum >>> 2 & 3 = 13272178845 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 229545382
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (229545382 >>> 5 ^ 229545382 << 2) + (229545382 >>> 3 ^ 229545382 << 4) ^ (13272178845 ^ 229545382) + (3684401 ^ 229545382)
mx= 979438043
z = v[0] + mx = 1987571332 + 979438043 = 1987571332

y = v[2 mod n] = v[0] = 1987571332
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1987571332 >>> 5 ^ 1987571332 << 2) + (1987571332 >>> 3 ^ 1987571332 << 4) ^ (13272178845 ^ 1987571332) + (926169145 ^ 1987571332)
mx= -423985630
z = v[1] + mx = -194440248 + -423985630 = -194440248

Sum = sum + delta = 15926614614
e = sum >>> 2 & 3 = 15926614614 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = -194440248
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (-194440248 >>> 5 ^ -194440248 << 2) + (-194440248 >>> 3 ^ -194440248 << 4) ^ (15926614614 ^ -194440248) + (808793394 ^ -194440248)
mx= 732091951
z = v[0] + mx = 2719663283 + 732091951 = 2719663283

y = v[2 mod n] = v[0] = 2719663283
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (2719663283 >>> 5 ^ 2719663283 << 2) + (2719663283 >>> 3 ^ 2719663283 << 4) ^ (15926614614 ^ 2719663283) + (808464434 ^ 2719663283)
mx= 1793838057
z = v[1] + mx = 1599397809 + 1793838057 = 1599397809

Sum = sum + delta = 18581050383
e = sum >>> 2 & 3 = 18581050383 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 1599397809
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1599397809 >>> 5 ^ 1599397809 << 2) + (1599397809 >>> 3 ^ 1599397809 << 4) ^ (18581050383 ^ 1599397809) + (3684401 ^ 1599397809)
mx= 309086753
z = v[0] + mx = 3028750036 + 309086753 = 3028750036

y = v[2 mod n] = v[0] = 3028750036
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3028750036 >>> 5 ^ 3028750036 << 2) + (3028750036 >>> 3 ^ 3028750036 << 4) ^ (18581050383 ^ 3028750036) + (926169145 ^ 3028750036)
mx= 1546558984
z = v[1] + mx = 3145956793 + 1546558984 = 3145956793

Sum = sum + delta = 21235486152
e = sum >>> 2 & 3 = 21235486152 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 3145956793
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3145956793 >>> 5 ^ 3145956793 << 2) + (3145956793 >>> 3 ^ 3145956793 << 4) ^ (21235486152 ^ 3145956793) + (926169145 ^ 3145956793)
mx= 1307989185
z = v[0] + mx = 4336739221 + 1307989185 = 4336739221

y = v[2 mod n] = v[0] = 4336739221
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4336739221 >>> 5 ^ 4336739221 << 2) + (4336739221 >>> 3 ^ 4336739221 << 4) ^ (21235486152 ^ 4336739221) + (3684401 ^ 4336739221)
mx= -948703125
z = v[1] + mx = 2197253668 + -948703125 = 2197253668

Sum = sum + delta = 23889921921
e = sum >>> 2 & 3 = 23889921921 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 2197253668
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (2197253668 >>> 5 ^ 2197253668 << 2) + (2197253668 >>> 3 ^ 2197253668 << 4) ^ (23889921921 ^ 2197253668) + (808464434 ^ 2197253668)
mx= -247884578

z = v[0] + mx = 4088854643 + -247884578 = 4088854643

y = v[2 mod n] = v[0] = 4088854643
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4088854643 >>> 5 ^ 4088854643 << 2) + (4088854643 >>> 3 ^ 4088854643 << 4) ^ (23889921921 ^ 4088854643) + (808793394 ^ 4088854643)
mx= -778689666
z = v[1] + mx = 1418564002 + -778689666 = 1418564002

Sum = sum + delta = 26544357690
e = sum >>> 2 & 3 = 26544357690 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 1418564002
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (1418564002 >>> 5 ^ 1418564002 << 2) + (1418564002 >>> 3 ^ 1418564002 << 4) ^ (26544357690 ^ 1418564002) + (926169145 ^ 1418564002)
mx= 1287268394
z = v[0] + mx = 5376123037 + 1287268394 = 5376123037

y = v[2 mod n] = v[0] = 5376123037
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (5376123037 >>> 5 ^ 5376123037 << 2) + (5376123037 >>> 3 ^ 5376123037 << 4) ^ (26544357690 ^ 5376123037) + (3684401 ^ 5376123037)
mx= -1136147616
z = v[1] + mx = 282416386 + -1136147616 = 282416386

Sum = sum + delta = 29198793459
e = sum >>> 2 & 3 = 29198793459 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 282416386
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (282416386 >>> 5 ^ 282416386 << 2) + (282416386 >>> 3 ^ 282416386 << 4) ^ (29198793459 ^ 282416386) + (808464434 ^ 282416386)
mx= -1363203071
z = v[0] + mx = 4012919966 + -1363203071 = 4012919966

y = v[2 mod n] = v[0] = 4012919966
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4012919966 >>> 5 ^ 4012919966 << 2) + (4012919966 >>> 3 ^ 4012919966 << 4) ^ (29198793459 ^ 4012919966) + (808793394 ^ 4012919966)
mx= -1462992746
z = v[1] + mx = -1180576360 + -1462992746 = -1180576360

Sum = sum + delta = 31853229228
e = sum >>> 2 & 3 = 31853229228 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = -1180576360
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)

mx = (-1180576360 >>> 5 ^ -1180576360 << 2) + (-1180576360 >>> 3 ^ -1180576360 << 4) ^ (31853229228 ^ -1180576360) + (3684401 ^ -1180576360)
mx= -56494814
z = v[0] + mx = 3956425152 + -56494814 = 3956425152

y = v[2 mod n] = v[0] = 3956425152
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3956425152 >>> 5 ^ 3956425152 << 2) + (3956425152 >>> 3 ^ 3956425152 << 4) ^ (31853229228 ^ 3956425152) + (926169145 ^ 3956425152)
mx= 373786147
z = v[1] + mx = -806790213 + 373786147 = -806790213

Sum = sum + delta = 34507664997
e = sum >>> 2 & 3 = 34507664997 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = -806790213
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (-806790213 >>> 5 ^ -806790213 << 2) + (-806790213 >>> 3 ^ -806790213 << 4) ^ (34507664997 ^ -806790213) + (808793394 ^ -806790213)
mx= -430550881
z = v[0] + mx = 3525874271 + -430550881 = 3525874271

y = v[2 mod n] = v[0] = 3525874271
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3525874271 >>> 5 ^ 3525874271 << 2) + (3525874271 >>> 3 ^ 3525874271 << 4) ^ (34507664997 ^ 3525874271) + (808464434 ^ 3525874271)
mx= 1005394702
z = v[1] + mx = 198604489 + 1005394702 = 198604489

Sum = sum + delta = 37162100766
e = sum >>> 2 & 3 = 37162100766 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 198604489
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (198604489 >>> 5 ^ 198604489 << 2) + (198604489 >>> 3 ^ 198604489 << 4) ^ (37162100766 ^ 198604489) + (3684401 ^ 198604489)
mx= 1407723284
z = v[0] + mx = 4933597555 + 1407723284 = 4933597555

y = v[2 mod n] = v[0] = 4933597555
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4933597555 >>> 5 ^ 4933597555 << 2) + (4933597555 >>> 3 ^ 4933597555 << 4) ^ (37162100766 ^ 4933597555) + (926169145 ^ 4933597555)
mx= 1829204626
z = v[1] + mx = 2027809115 + 1829204626 = 2027809115

Sum = sum + delta = 39816536535
e = sum >>> 2 & 3 = 39816536535 >>> 2 & 3 = 1

y = v[1 mod n] = v[1] = 2027809115
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (2027809115 >>> 5 ^ 2027809115 << 2) + (2027809115 >>> 3 ^ 2027809115 << 4) ^ (39816536535 ^ 2027809115) + (808793394 ^ 2027809115)
mx= -437201356
z = v[0] + mx = 4496396199 + -437201356 = 4496396199

y = v[2 mod n] = v[0] = 4496396199
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4496396199 >>> 5 ^ 4496396199 << 2) + (4496396199 >>> 3 ^ 4496396199 << 4) ^ (39816536535 ^ 4496396199) + (808464434 ^ 4496396199)
mx= 1955434592
z = v[1] + mx = 3983243707 + 1955434592 = 3983243707

Sum = sum + delta = 42470972304
e = sum >>> 2 & 3 = 42470972304 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 3983243707
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3983243707 >>> 5 ^ 3983243707 << 2) + (3983243707 >>> 3 ^ 3983243707 << 4) ^ (42470972304 ^ 3983243707) + (808464434 ^ 3983243707)
mx= -1768318436
z = v[0] + mx = 2728077763 + -1768318436 = 2728077763

y = v[2 mod n] = v[0] = 2728077763
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (2728077763 >>> 5 ^ 2728077763 << 2) + (2728077763 >>> 3 ^ 2728077763 << 4) ^ (42470972304 ^ 2728077763) + (808793394 ^ 2728077763)
mx= 432060430
z = v[1] + mx = 4415304137 + 432060430 = 4415304137

Sum = sum + delta = 45125408073
e = sum >>> 2 & 3 = 45125408073 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 4415304137
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4415304137 >>> 5 ^ 4415304137 << 2) + (4415304137 >>> 3 ^ 4415304137 << 4) ^ (45125408073 ^ 4415304137) + (926169145 ^ 4415304137)
mx= 942370595
z = v[0] + mx = 3670448358 + 942370595 = 3670448358

y = v[2 mod n] = v[0] = 3670448358
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3670448358 >>> 5 ^ 3670448358 << 2) + (3670448358 >>> 3 ^ 3670448358 << 4) ^ (45125408073 ^ 3670448358) + (3684401 ^ 3670448358)
mx= 311794269
z = v[1] + mx = 4727098406 + 311794269 = 4727098406

Sum = sum + delta = 47779843842
e = sum >>> 2 & 3 = 47779843842 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 4727098406
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4727098406 >>> 5 ^ 4727098406 << 2) + (4727098406 >>> 3 ^ 4727098406 << 4) ^ (47779843842 ^ 4727098406) + (808464434 ^ 4727098406)
mx= 922158565
z = v[0] + mx = 4592606923 + 922158565 = 4592606923

y = v[2 mod n] = v[0] = 4592606923
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4592606923 >>> 5 ^ 4592606923 << 2) + (4592606923 >>> 3 ^ 4592606923 << 4) ^ (47779843842 ^ 4592606923) + (808793394 ^ 4592606923)
mx= 1334463105
z = v[1] + mx = 6061561511 + 1334463105 = 6061561511

Sum = sum + delta = 50434279611
e = sum >>> 2 & 3 = 50434279611 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 6061561511
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (6061561511 >>> 5 ^ 6061561511 << 2) + (6061561511 >>> 3 ^ 6061561511 << 4) ^ (50434279611 ^ 6061561511) + (926169145 ^ 6061561511)
mx= 1974038135
z = v[0] + mx = 6566645058 + 1974038135 = 6566645058

y = v[2 mod n] = v[0] = 6566645058
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (6566645058 >>> 5 ^ 6566645058 << 2) + (6566645058 >>> 3 ^ 6566645058 << 4) ^ (50434279611 ^ 6566645058) + (3684401 ^ 6566645058)
mx= 1083931014
z = v[1] + mx = 7145492525 + 1083931014 = 7145492525

Sum = sum + delta = 53088715380
e = sum >>> 2 & 3 = 53088715380 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = 7145492525
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (7145492525 >>> 5 ^ 7145492525 << 2) + (7145492525 >>> 3 ^ 7145492525 << 4) ^ (53088715380 ^ 7145492525) + (808793394 ^ 7145492525)
mx= -1584145806
z = v[0] + mx = 4982499252 + -1584145806 = 4982499252

y = v[2 mod n] = v[0] = 4982499252
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4982499252 >>> 5 ^ 4982499252 << 2) + (4982499252 >>> 3 ^ 4982499252 << 4) ^ (53088715380 ^ 4982499252) + (808464434 ^ 4982499252)
mx= -1608037147

z = v[1] + mx = 5537455378 + -1608037147 = 5537455378

Sum = sum + delta = 55743151149
e = sum >>> 2 & 3 = 55743151149 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 5537455378
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (5537455378 >>> 5 ^ 5537455378 << 2) + (5537455378 >>> 3 ^ 5537455378 << 4) ^ (55743151149 ^ 5537455378) + (3684401 ^ 5537455378)
mx= 783158080
z = v[0] + mx = 5765657332 + 783158080 = 5765657332

y = v[2 mod n] = v[0] = 5765657332
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (5765657332 >>> 5 ^ 5765657332 << 2) + (5765657332 >>> 3 ^ 5765657332 << 4) ^ (55743151149 ^ 5765657332) + (926169145 ^ 5765657332)
mx= -1051171645
z = v[1] + mx = 4486283733 + -1051171645 = 4486283733

Sum = sum + delta = 58397586918
e = sum >>> 2 & 3 = 58397586918 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = 4486283733
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4486283733 >>> 5 ^ 4486283733 << 2) + (4486283733 >>> 3 ^ 4486283733 << 4) ^ (58397586918 ^ 4486283733) + (808793394 ^ 4486283733)
mx= 707435486
z = v[0] + mx = 6473092818 + 707435486 = 6473092818

y = v[2 mod n] = v[0] = 6473092818
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (6473092818 >>> 5 ^ 6473092818 << 2) + (6473092818 >>> 3 ^ 6473092818 << 4) ^ (58397586918 ^ 6473092818) + (808464434 ^ 6473092818)
mx= -625410516
z = v[1] + mx = 3860873217 + -625410516 = 3860873217

Sum = sum + delta = 61052022687
e = sum >>> 2 & 3 = 61052022687 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 3860873217
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3860873217 >>> 5 ^ 3860873217 << 2) + (3860873217 >>> 3 ^ 3860873217 << 4) ^ (61052022687 ^ 3860873217) + (3684401 ^ 3860873217)
mx= -1467732518
z = v[0] + mx = 5005360300 + -1467732518 = 5005360300

y = v[2 mod n] = v[0] = 5005360300
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)

mx = (5005360300 >>> 5 ^ 5005360300 << 2) + (5005360300 >>> 3 ^ 5005360300 << 4) ^ (61052022687 ^ 5005360300) + (926169145 ^ 5005360300)
mx= 1915902818
z = v[1] + mx = 5776776035 + 1915902818 = 5776776035

Sum = sum + delta = 63706458456
e = sum >>> 2 & 3 = 63706458456 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 5776776035
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (5776776035 >>> 5 ^ 5776776035 << 2) + (5776776035 >>> 3 ^ 5776776035 << 4) ^ (63706458456 ^ 5776776035) + (926169145 ^ 5776776035)
mx= 224676646
z = v[0] + mx = 5230036946 + 224676646 = 5230036946

y = v[2 mod n] = v[0] = 5230036946
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (5230036946 >>> 5 ^ 5230036946 << 2) + (5230036946 >>> 3 ^ 5230036946 << 4) ^ (63706458456 ^ 5230036946) + (3684401 ^ 5230036946)
mx= 1184256829
z = v[1] + mx = 6961032864 + 1184256829 = 6961032864

Sum = sum + delta = 66360894225
e = sum >>> 2 & 3 = 66360894225 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 6961032864
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (6961032864 >>> 5 ^ 6961032864 << 2) + (6961032864 >>> 3 ^ 6961032864 << 4) ^ (66360894225 ^ 6961032864) + (808464434 ^ 6961032864)
mx= -521004662
z = v[0] + mx = 4709032284 + -521004662 = 4709032284

y = v[2 mod n] = v[0] = 4709032284
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4709032284 >>> 5 ^ 4709032284 << 2) + (4709032284 >>> 3 ^ 4709032284 << 4) ^ (66360894225 ^ 4709032284) + (808793394 ^ 4709032284)
mx= 2016022750
z = v[1] + mx = 8977055614 + 2016022750 = 8977055614

Sum = sum + delta = 69015329994
e = sum >>> 2 & 3 = 69015329994 >>> 2 & 3 = 2
y = v[1 mod n] = v[1] = 8977055614
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (8977055614 >>> 5 ^ 8977055614 << 2) + (8977055614 >>> 3 ^ 8977055614 << 4) ^ (69015329994 ^ 8977055614) + (926169145 ^ 8977055614)
mx= -166299415
z = v[0] + mx = 4542732869 + -166299415 = 4542732869

y = v[2 mod n] = v[0] = 4542732869
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (4542732869 >>> 5 ^ 4542732869 << 2) + (4542732869 >>> 3 ^ 4542732869 << 4) ^ (69015329994 ^ 4542732869) + (3684401 ^ 4542732869)
mx= 110554589
z = v[1] + mx = 9087610203 + 110554589 = 9087610203

Sum = sum + delta = 71669765763
e = sum >>> 2 & 3 = 71669765763 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 9087610203
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (9087610203 >>> 5 ^ 9087610203 << 2) + (9087610203 >>> 3 ^ 9087610203 << 4) ^ (71669765763 ^ 9087610203) + (808464434 ^ 9087610203)
mx= -1350763680
z = v[0] + mx = 3191969189 + -1350763680 = 3191969189

y = v[2 mod n] = v[0] = 3191969189
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3191969189 >>> 5 ^ 3191969189 << 2) + (3191969189 >>> 3 ^ 3191969189 << 4) ^ (71669765763 ^ 3191969189) + (808793394 ^ 3191969189)
mx= 1355358368
z = v[1] + mx = 10442968571 + 1355358368 = 10442968571

Sum = sum + delta = 74324201532
e = sum >>> 2 & 3 = 74324201532 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 10442968571
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (10442968571 >>> 5 ^ 10442968571 << 2) + (10442968571 >>> 3 ^ 10442968571 << 4) ^ (74324201532 ^ 10442968571) + (3684401 ^ 10442968571)
mx= 722730499
z = v[0] + mx = 3914699688 + 722730499 = 3914699688

y = v[2 mod n] = v[0] = 3914699688
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3914699688 >>> 5 ^ 3914699688 << 2) + (3914699688 >>> 3 ^ 3914699688 << 4) ^ (74324201532 ^ 3914699688) + (926169145 ^ 3914699688)
mx= -1355414505
z = v[1] + mx = 9087554066 + -1355414505 = 9087554066

Sum = sum + delta = 76978637301
e = sum >>> 2 & 3 = 76978637301 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = 9087554066
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (9087554066 >>> 5 ^ 9087554066 << 2) + (9087554066 >>> 3 ^ 9087554066 << 4) ^ (76978637301 ^ 9087554066) + (808793394 ^ 9087554066)
mx= 1343424317

z = v[0] + mx = 5258124005 + 1343424317 = 5258124005

y = v[2 mod n] = v[0] = 5258124005
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (5258124005 >>> 5 ^ 5258124005 << 2) + (5258124005 >>> 3 ^ 5258124005 << 4) ^ (76978637301 ^ 5258124005) + (808464434 ^ 5258124005)
mx= -1468722776
z = v[1] + mx = 7618831290 + -1468722776 = 7618831290

Sum = sum + delta = 79633073070
e = sum >>> 2 & 3 = 79633073070 >>> 2 & 3 = 3
y = v[1 mod n] = v[1] = 7618831290
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (7618831290 >>> 5 ^ 7618831290 << 2) + (7618831290 >>> 3 ^ 7618831290 << 4) ^ (79633073070 ^ 7618831290) + (3684401 ^ 7618831290)
mx= -2052193197
z = v[0] + mx = 3205930808 + -2052193197 = 3205930808

y = v[2 mod n] = v[0] = 3205930808
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3205930808 >>> 5 ^ 3205930808 << 2) + (3205930808 >>> 3 ^ 3205930808 << 4) ^ (79633073070 ^ 3205930808) + (926169145 ^ 3205930808)
mx= 1570007831
z = v[1] + mx = 9188839121 + 1570007831 = 9188839121

Sum = sum + delta = 82287508839
e = sum >>> 2 & 3 = 82287508839 >>> 2 & 3 = 1
y = v[1 mod n] = v[1] = 9188839121
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (9188839121 >>> 5 ^ 9188839121 << 2) + (9188839121 >>> 3 ^ 9188839121 << 4) ^ (82287508839 ^ 9188839121) + (808793394 ^ 9188839121)
mx= -776305563
z = v[0] + mx = 2429625245 + -776305563 = 2429625245

y = v[2 mod n] = v[0] = 2429625245
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (2429625245 >>> 5 ^ 2429625245 << 2) + (2429625245 >>> 3 ^ 2429625245 << 4) ^ (82287508839 ^ 2429625245) + (808464434 ^ 2429625245)
mx= 1066244002
z = v[1] + mx = 10255083123 + 1066244002 = 10255083123

Sum = sum + delta = 84941944608
e = sum >>> 2 & 3 = 84941944608 >>> 2 & 3 = 0
y = v[1 mod n] = v[1] = 10255083123
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)

mx = (10255083123 >>> 5 ^ 10255083123 << 2) + (10255083123 >>> 3 ^ 10255083123 << 4) ^ (84941944608 ^ 10255083123) + (808464434 ^ 10255083123)
mx= 1067178857
z = v[0] + mx = 3496804102 + 1067178857 = 3496804102

y = v[2 mod n] = v[0] = 3496804102
mx = (z >>> 5 ^ y << 2) + (y >>> 3 ^ z << 4) ^ (sum ^ y) + (k[p & 3 ^ e] ^ z)
mx = (3496804102 >>> 5 ^ 3496804102 << 2) + (3496804102 >>> 3 ^ 3496804102 << 4) ^ (84941944608 ^ 3496804102) + (808793394 ^ 3496804102)
mx= -1792108230
z = v[1] + mx = 8462974893 + -1792108230 = 8462974893

Encoding TEA v = [3496804102, 8462974893]

Output akhir untuk kata "Halo" diencode kedalam Base64 = "Bv9s0K2/bvg="

Di google drive: !@#$%TEA***Bv9s0K2/bvg=

III.3. Perancangan

Desain sistem pada penelitian ini dibagi menjadi dua desain, yaitu desain sistem secara global untuk penggambaran model sistem secara garis besar dan desain sistem secara detail untuk membantu dalam pembuatan sistem.

III.3.1. Desain Sistem

Perancangan Aplikasi Pengamanan Data Pada Google Drive Dengan Menggunakan Algoritma TEA menyajikan informasi data yang dapat digunakan oleh pengguna google drive.

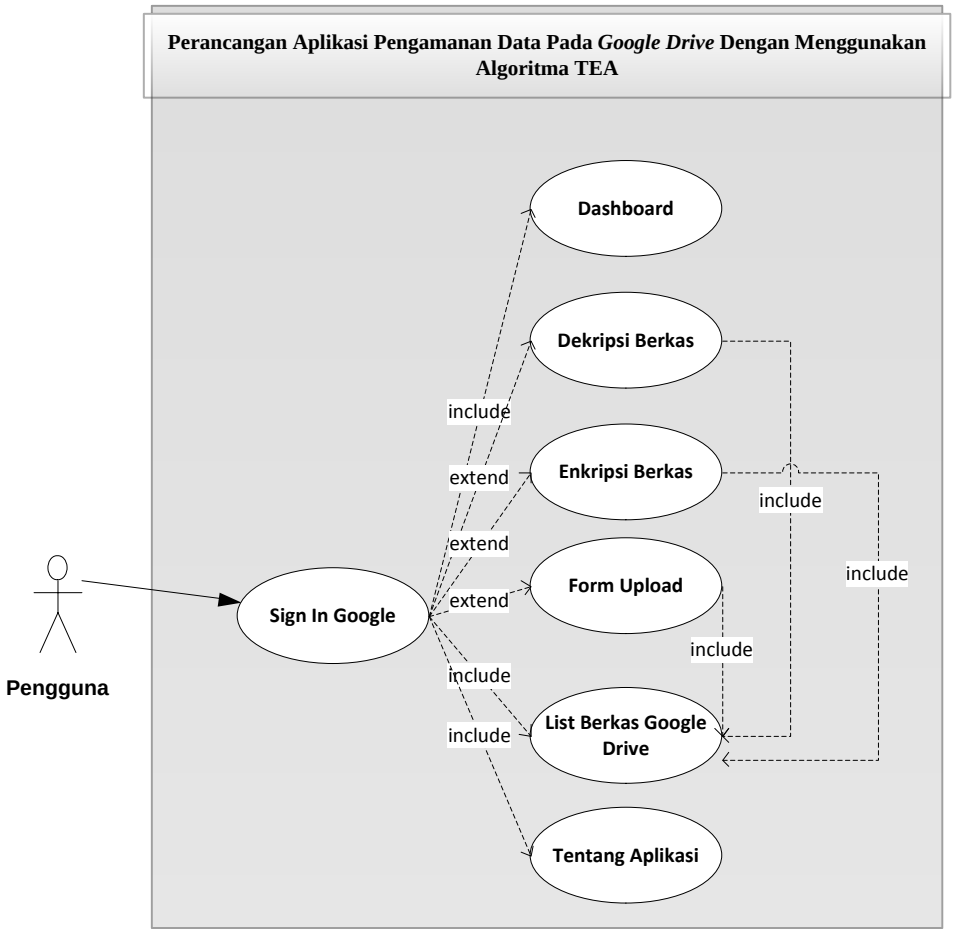
III.3.2. Desain Sistem Secara Global

Desain sistem secara global menggunakan bahasa pemodelan UML yang terdiri dari *Usecase Diagram*, *Activity Diagram* dan *Sequence Diagram*.

III.3.2.1. *Usecase* Diagram

Dalam penyusunan suatu program diperlukan suatu model data yang berbentuk diagram yang dapat menjelaskan suatu alur proses sistem yang akan di bangun. Dalam penulisan skripsi

ini penulis menggunakan metode UML yang dalam metode itu penulis menerapkan diagram *Use Case*. Maka digambarlah suatu bentuk diagram *Use Case* yang dapat dilihat pada gambar III.2 berikut :



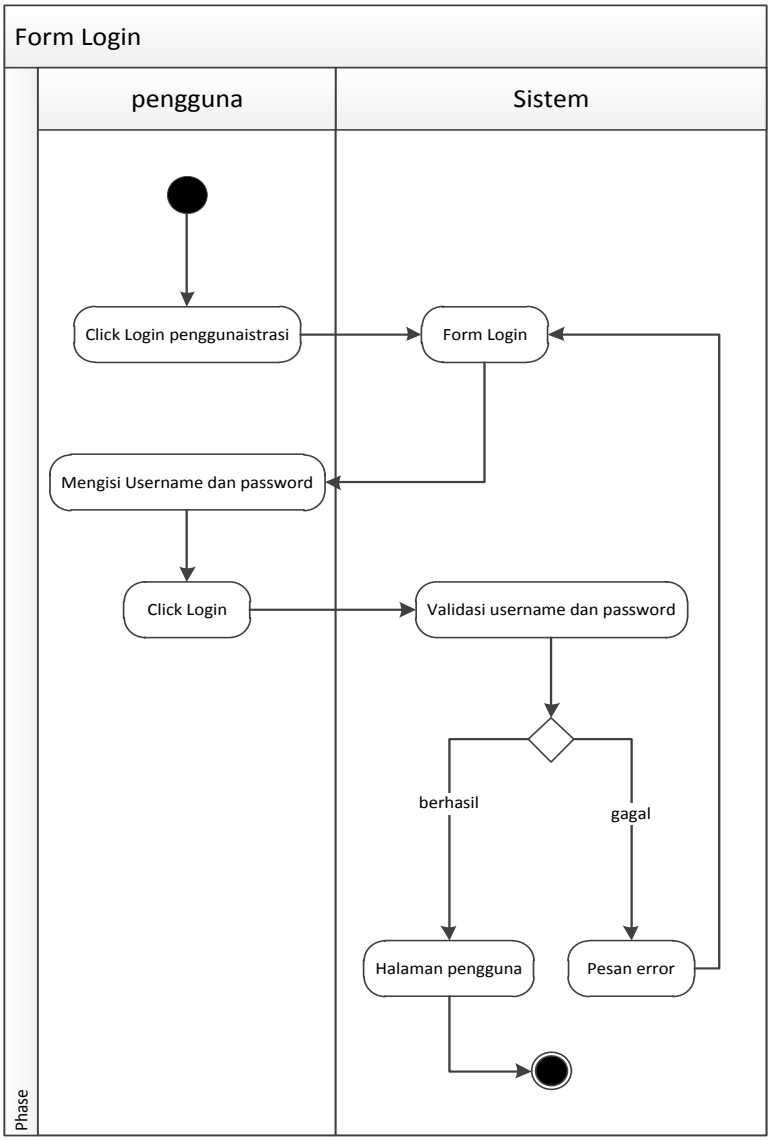
Gambar III.2. *Use Case* Diagram Aplikasi *Google Drive*

III.2.2.2. *Activity* Diagram

Bisnis proses yang telah digambarkan pada *use case diagram* dijabarkan dengan *Activity diagram* :

1. *Activity Diagram Login*

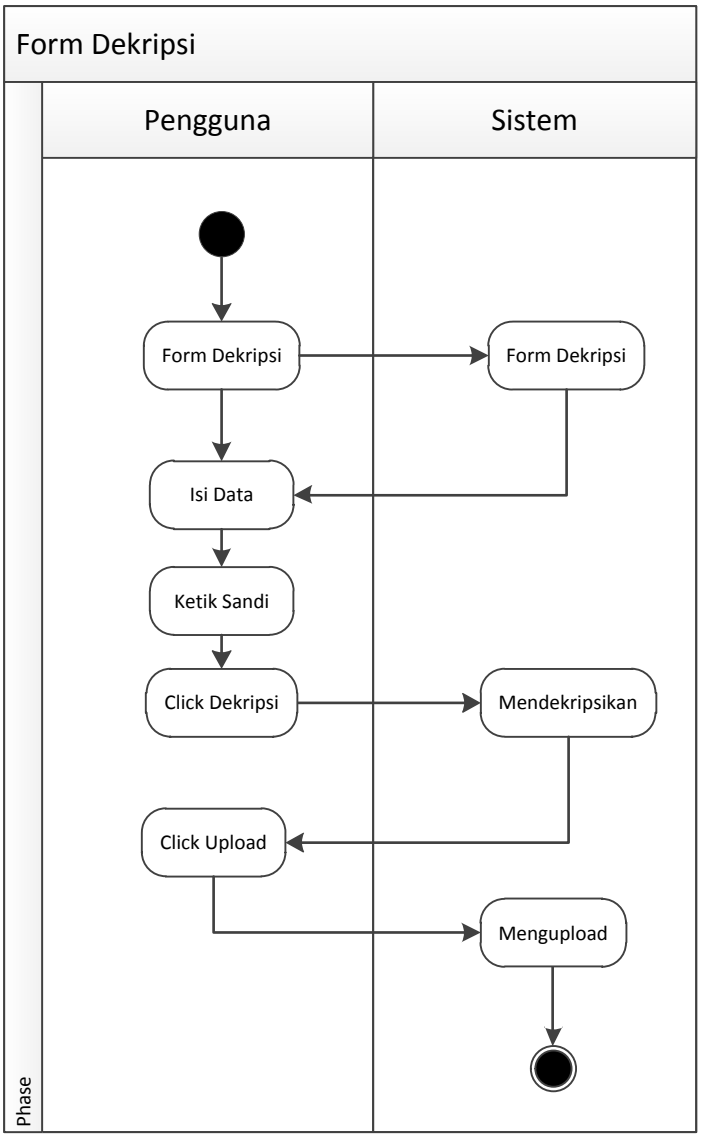
Aktivitas *login* yang dilakukan oleh user dapat diterangkan dengan langkah-langkah *state* berikut :



Gambar III.3. ***Activity Diagram Login***

2. *Activity Diagram* Dekripsi

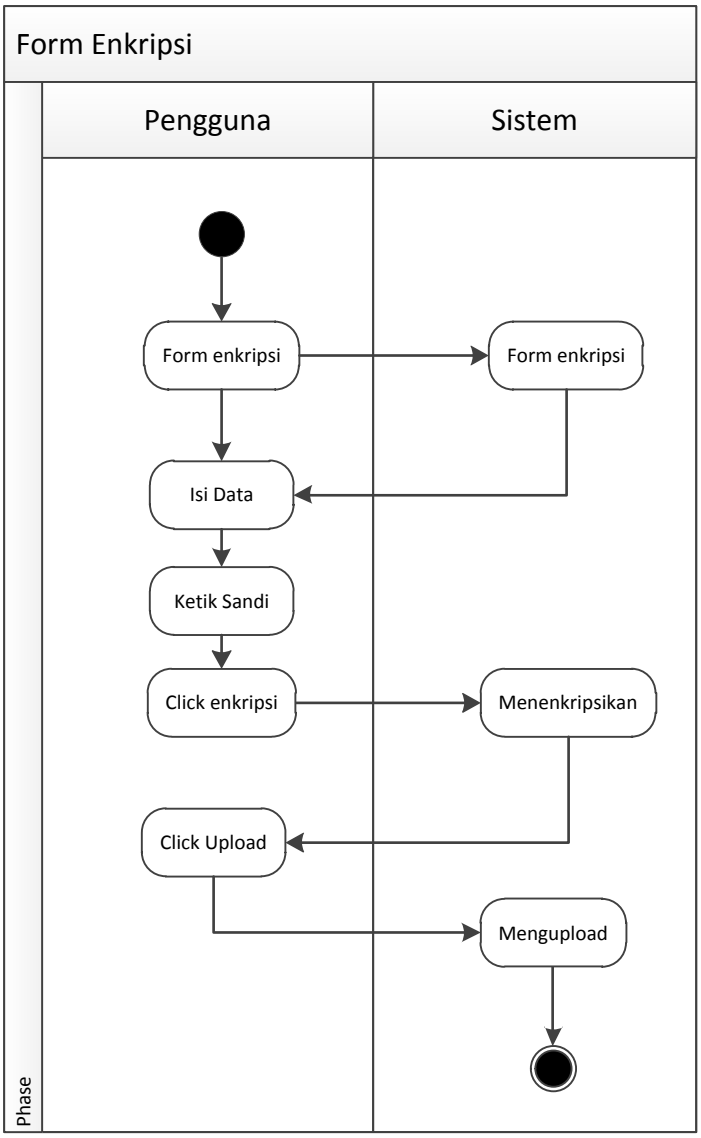
Aktivitas yang dilakukan oleh Admin pada pengolahan dekripsi dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.4 berikut :



Gambar III.4. *Activity Diagram* Dekripsi

3. *Activity Diagram* Enkripsi

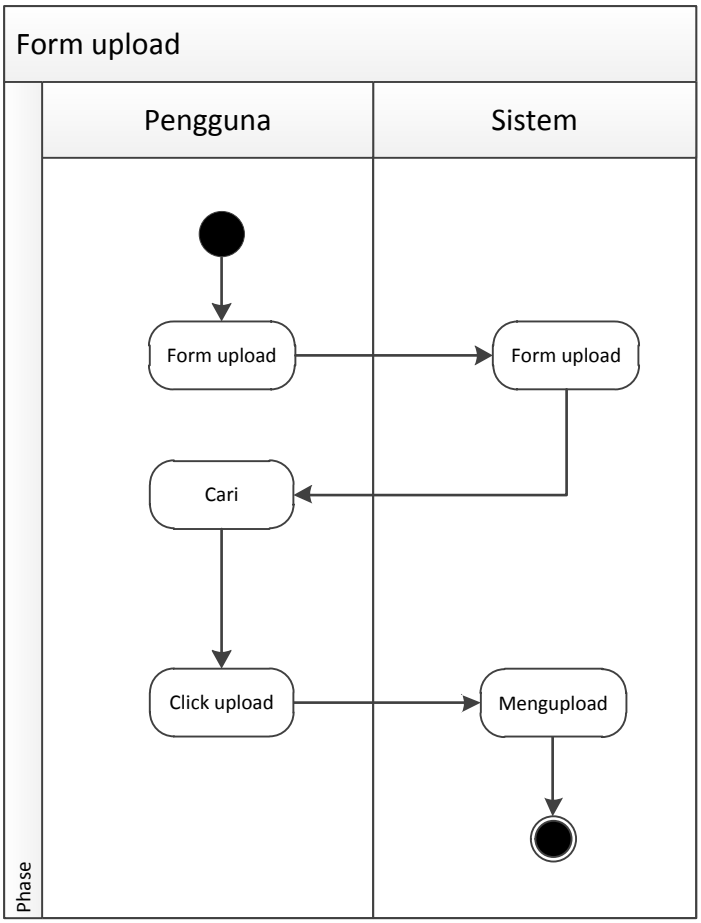
Aktivitas yang dilakukan oleh Admin pada pengolahan enkripsi dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.5 berikut :



Gambar III.5. *Activity Diagram* Enkripsi

4. *Activity Diagram* Upload

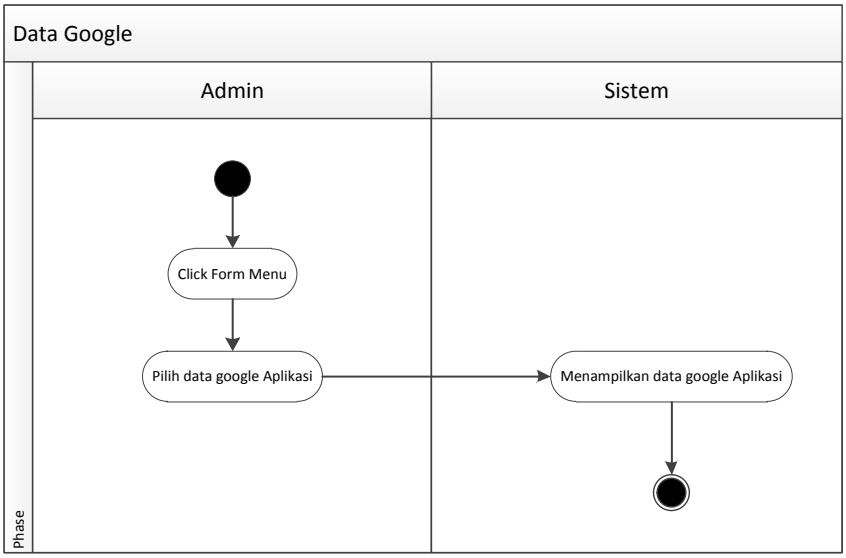
Aktivitas yang dilakukan oleh Admin pada pengolahan upload dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.6 berikut :



Gambar III.6. *Activity Diagram* Upload

3. *Activity Diagram* Data Google

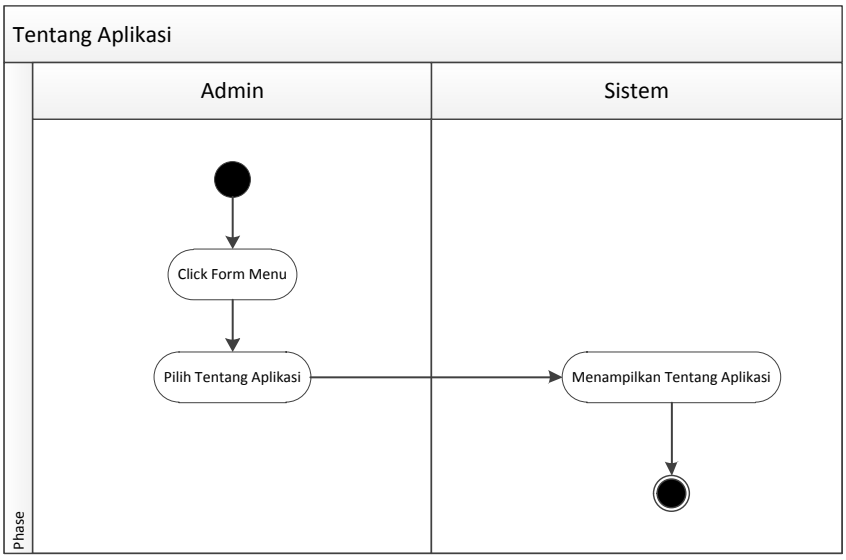
Aktivitas yang dilakukan oleh Admin pada pengolahan data google dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.7 berikut :



Gambar III.7. *Activity Diagram* Data Google

4. *Activity Diagram* Tentang

Aktivitas yang dilakukan oleh Admin pada pengolahan tentang dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.8 berikut :



Gambar III.8. *Activity Diagram* Tentang

III.3.2.3. *Sequence* Diagram

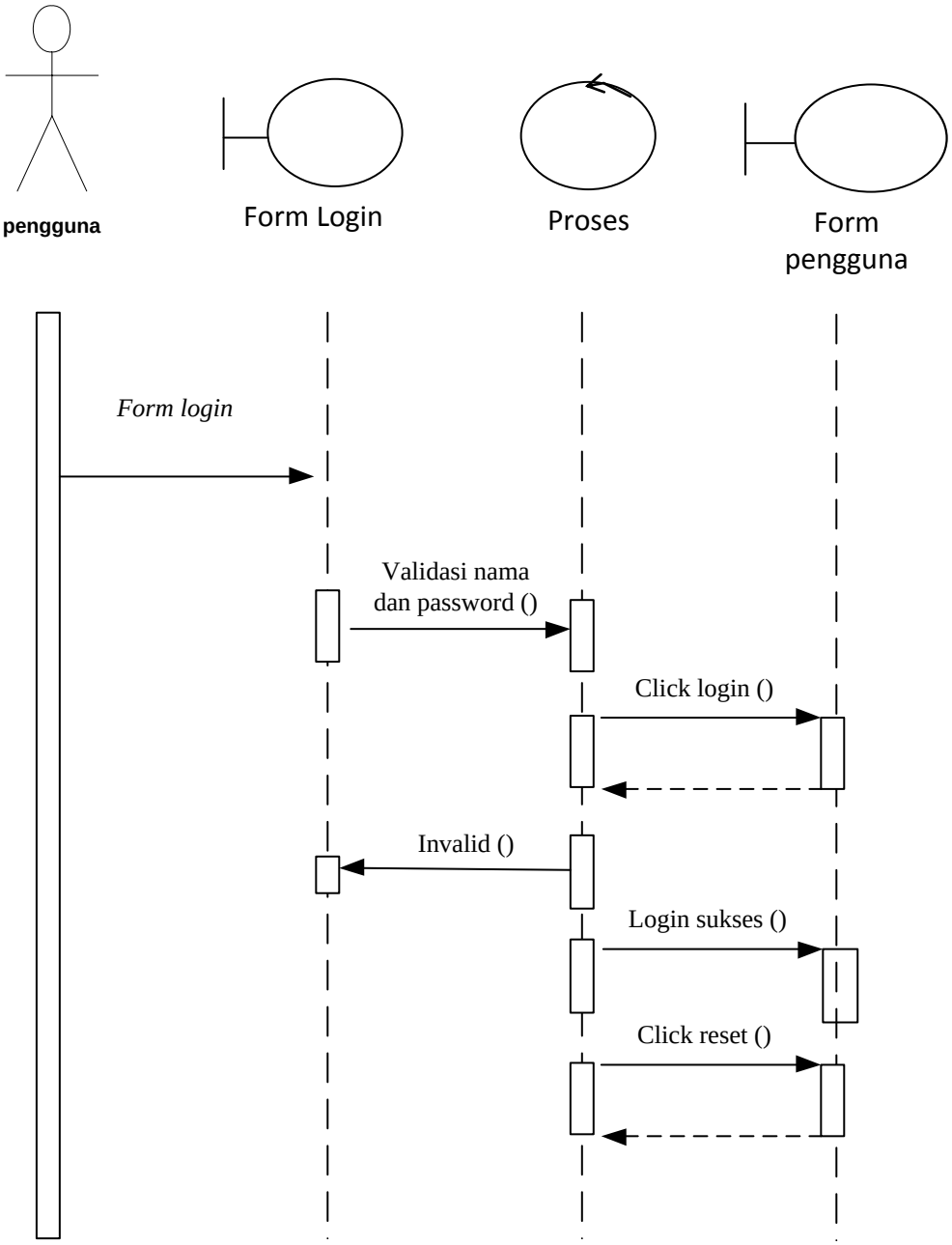
Sequence Diagram (diagram urutan) adalah suatu diagram yang memperlihatkan atau menampilkan interaksi-interaksi antar objek di dalam sistem yang disusun pada sebuah urutan atau rangkaian waktu. Interaksi antar objek tersebut termasuk pengguna, *display*, dan sebagainya berupa pesan/*message*.

Sequence Diagram digunakan untuk menggambarkan skenario atau rangkaian langkah – langkah yang dilakukan sebagai sebuah respon dari suatu kejadian/event untuk menghasilkan output tertentu. *Sequence* Diagram diawali dari apa yang me-trigger aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Diagram ini secara khusus berasosiasi dengan use case diagram. *Sequence* diagram juga memperlihatkan tahap demi tahap apa yang seharusnya terjadi untuk menghasilkan sesuatu di dalam use case. *Sequence* Diagram juga dapat merubah atribut atau method pada class yang telah dibentuk oleh class diagram, bahkan menciptakan sebuah class baru. *Sequence* Diagram memodelkan aliran logika dalam sebuah sistem dalam cara yang visual.

1. *Sequence Diagram Login*

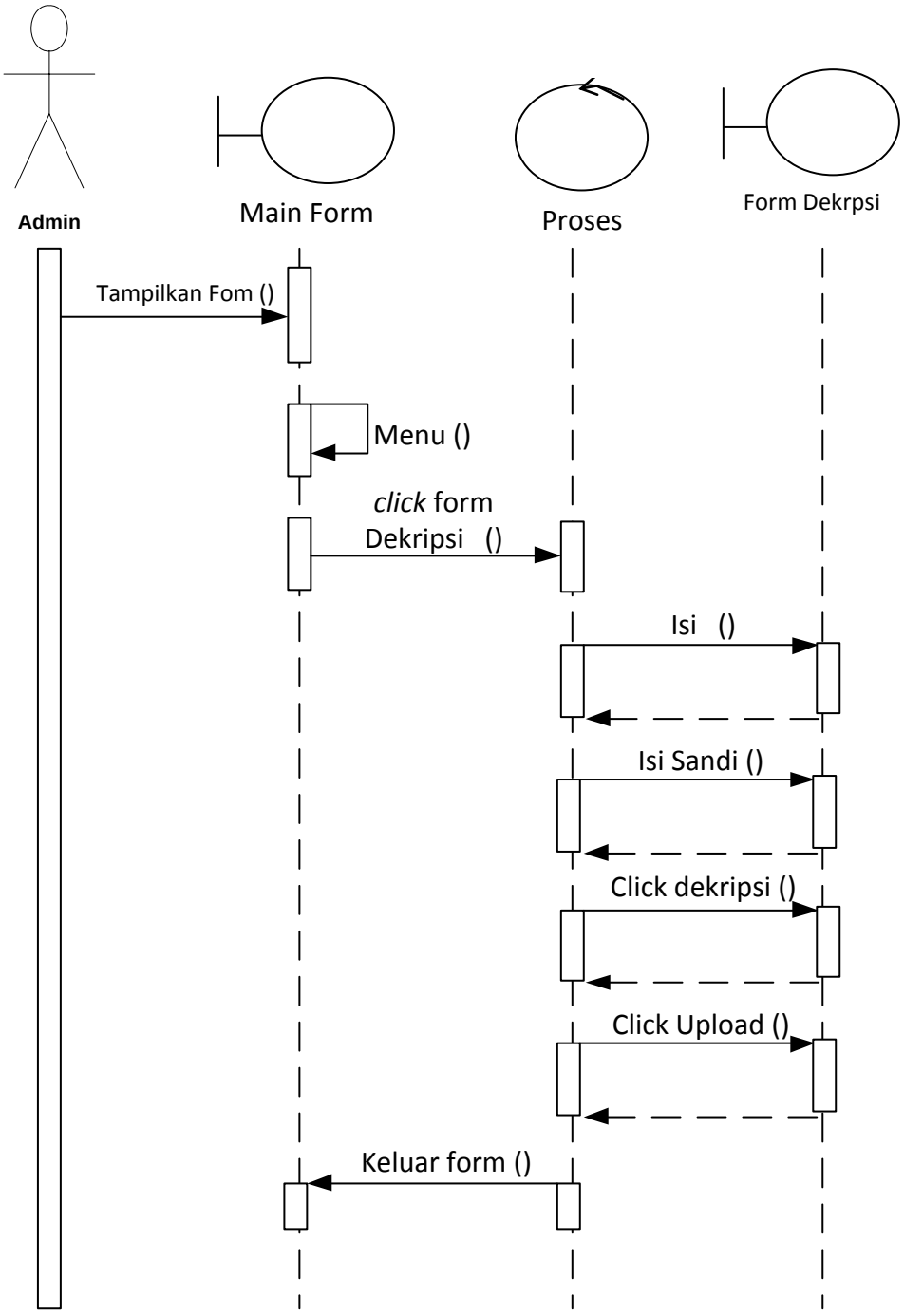
Serangkaian kinerja sistem *login* yang dilakukan oleh user dapat diterangkan dengan langkah-langkah *state* berikut :



Gambar III.9. *Sequence Diagram Login*

2. *Sequence Diagram* Dekripsi

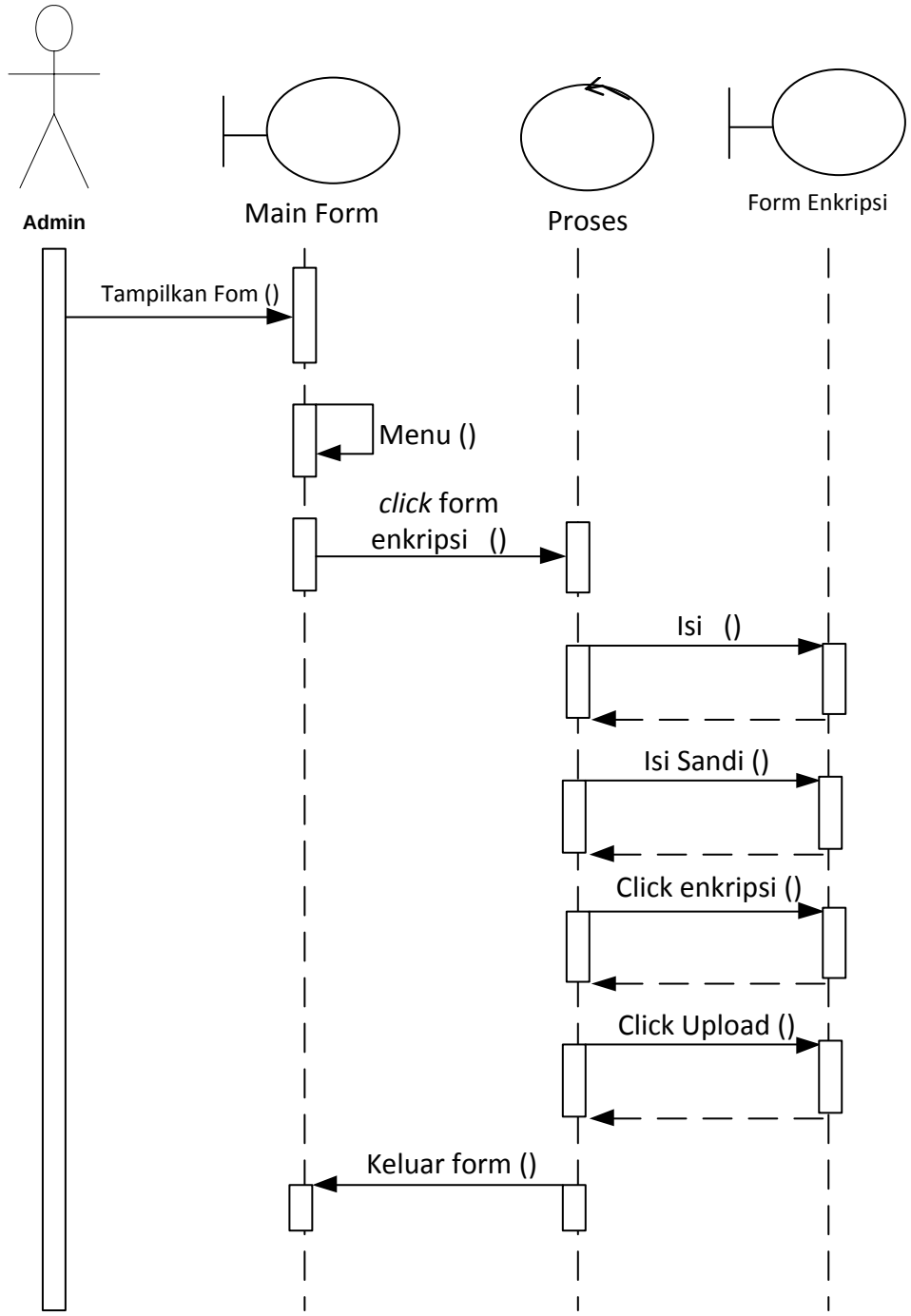
Serangkaian kinerja sistem yang dilakukan oleh Admin pada pengolahan dekripsi dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.10 berikut :



Gambar III.10. *Sequence Diagram* Dekripsi

3. *Sequence Diagram* Enkripsi

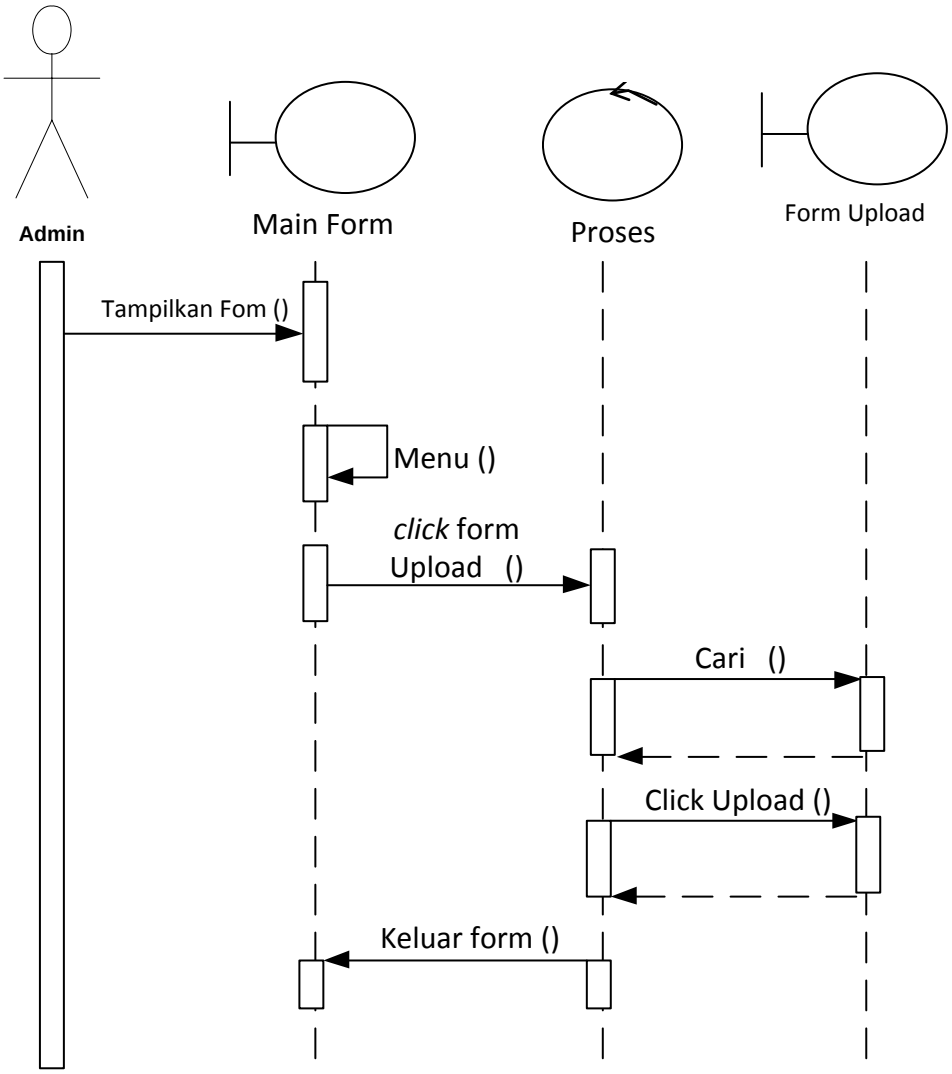
Serangkaian kinerja sistem yang dilakukan oleh Admin pada pengolahan enkripsi dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.11 berikut :



Gambar III.11. *Sequence Diagram* Enkripsi

4. *Sequence Diagram* Upload

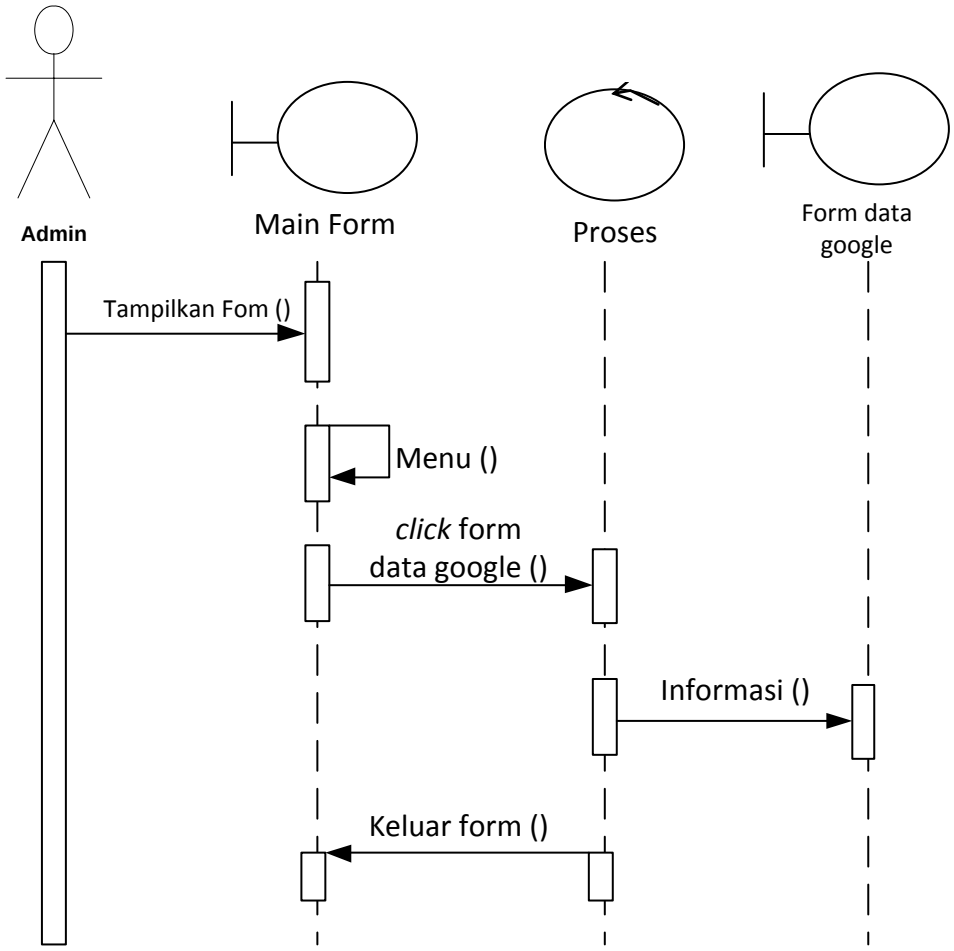
Serangkaian kinerja sistem yang dilakukan oleh Admin pada pengolahan upload dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.12 berikut :



Gambar III.12. *Sequence Diagram* Upload

5. *Sequence Diagram* Data Google

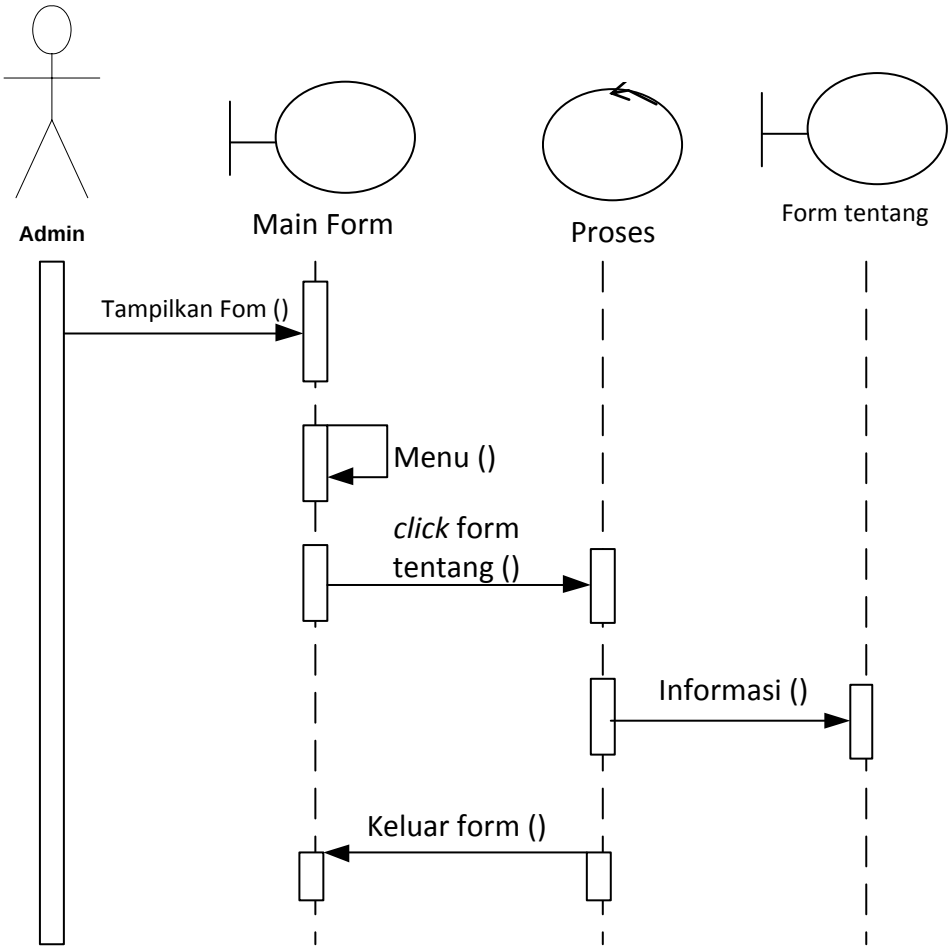
Serangkaian kinerja sistem yang dilakukan oleh Admin pada pengolahan data google dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.13 berikut :



Gambar III.13. *Sequence Diagram* Data Google

6. *Sequence Diagram* Tentang

Serangkaian kinerja sistem yang dilakukan oleh Admin pada pengolahan tentang dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.14 berikut :



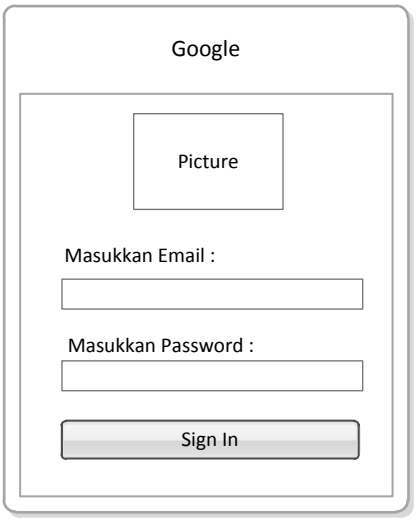
Gambar III.14. *Sequence Diagram* Tentang

III.2.3. Desain Sistem Secara Detail

Tahap perancangan berikutnya yaitu desain sistem secara detail yang meliputi desain sistem.

1. Tampilan *Form Login*

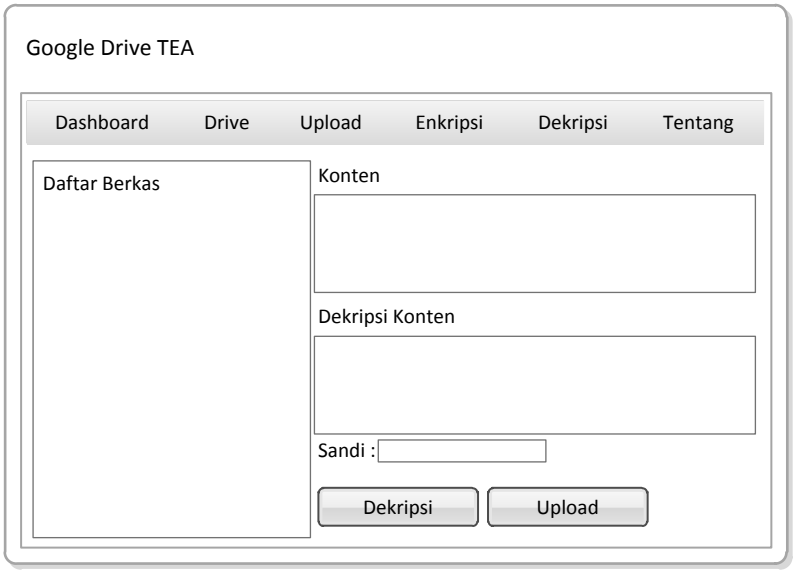
Tampilan sistem *login* yang dilakukan oleh user dapat diterangkan dengan langkah-langkah *state* berikut :



Gambar III.15. Tampilan *Form Login*

2. Tampilan *Form* Dekripsi

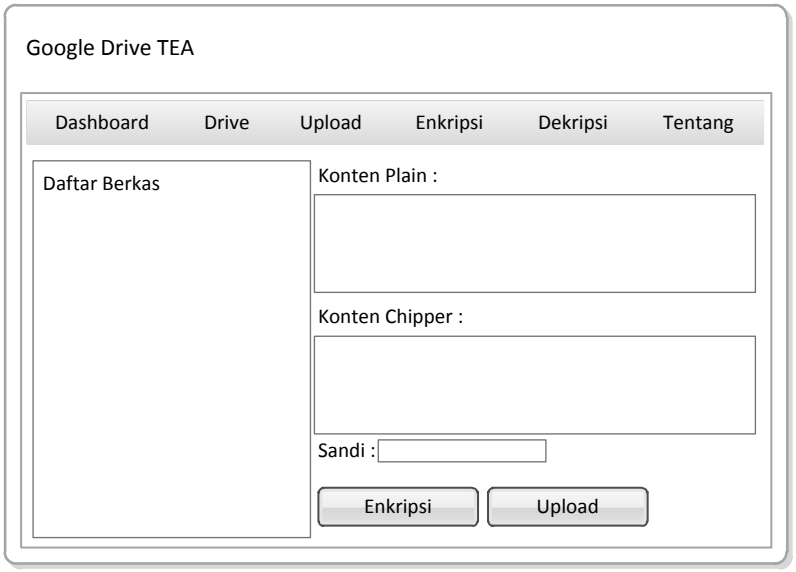
Tampilan sistem yang dilakukan oleh Admin pada pengolahan dekripsi dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.16 berikut :



Gambar III.16. Tampilan *Form* Dekripsi

3. Tampilan *Form* Enkripsi

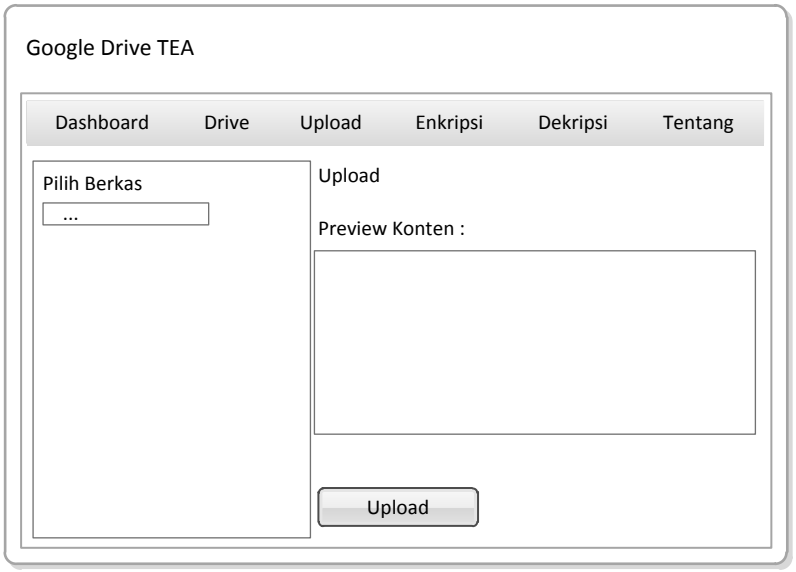
Tampilan sistem yang dilakukan oleh Admin pada pengolahan enkripsi dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.17 berikut :



Gambar III.17. Tampilan *Form* Enkripsi

4. Tampilan *Form* Upload

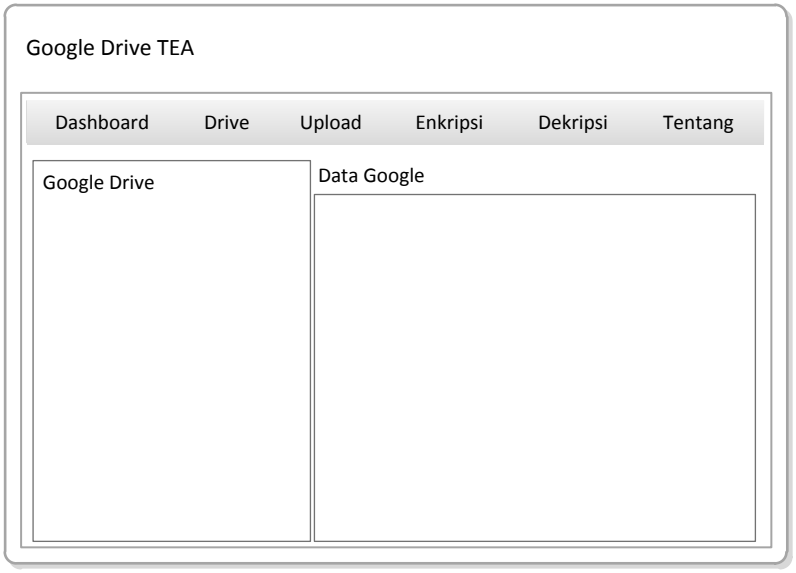
Tampilan sistem yang dilakukan oleh Admin pada pengolahan upload dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.18 berikut :



Gambar III.18. Tampilan *Form* Upload

5. Tampilan *Form* Data Google

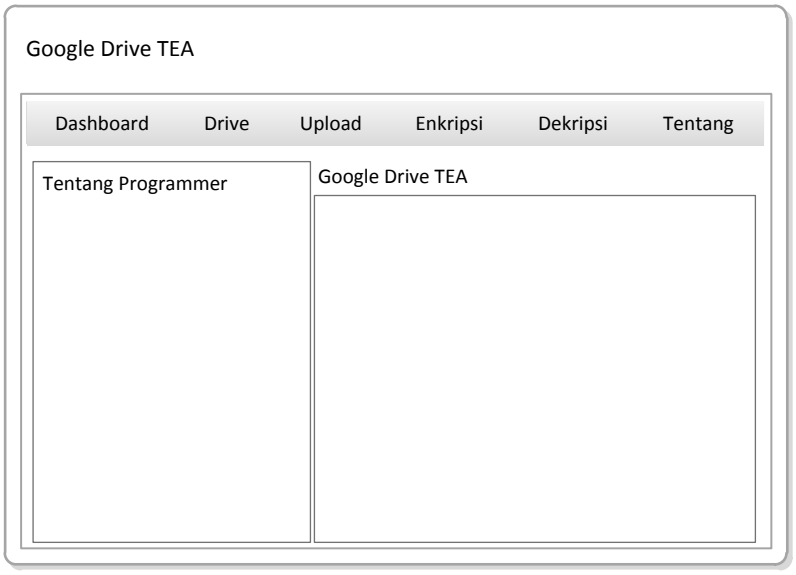
Tampilan sistem yang dilakukan oleh Admin pada pengolahan data google dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.19 berikut :



Gambar III.19. Tampilan *Form* Data Google

6. Tampilan *Form* Tentang

Tampilan sistem yang dilakukan oleh Admin pada pengolahan tentang dapat diterangkan dengan langkah-langkah *state* berikut, yang ditunjukkan pada gambar III.20 berikut :



Gambar III.20. Tampilan *Form* Tentang