PERANCANGAN APLIKASI DIARY MENGGUNAKAN ALGORITMA KRIPTOGRAFI RC6 BERBASIS ANDROID

Aedhoh Salim Assaidi [1)], **Armadyah Amborowati**[2)]

[1,2)] *Teknik Informatika STMIK AMIKOM Yogyakarta*
Jl Ringroad Utara, Condongcatur, Depok, Sleman, Yogyakarta Indonesia 55283
Email : aedhoh.a@students.amikom.ac.id[1)], armadyah.a@amikom.ac.id[2)]

Abstract *- Diary is a note about daily life. Development in information and technology changes every aspect of humans life including making a diary. Now days people not only writing their dialy life on diary books but also on their gadget or smartphone in order to maintain their privacy. Usually diary notes are saved using plaint text so everybody can see and read everything on that diary. With cryptography algorytm diary are not save in plaint text but in chiper text. chiper text are writen with shuffling and changing the plain text using encryption method. Keys are used to lock the encryption so that only the right key can open the chiper text. This problem make me want to developt a diary application program for better security and privacy for user.*

Keywords : *RC6, Diary, RC6 Diary.*

1. Pendahuluan

Latar Belakang

Menulis *Diary* atau buku harian adalah aktifitas yang dilakukan oleh banyak orang setiap harinya. Perkembangan teknologi informasi berpengaruh besar pada banyaknya perubahan aktifitas manusia, tak terkecuali menulis *Diary*. Menulis *Diary* yang dulunya ditulis pada buku tulis lalu disimpan pada tempat-tempat tertentu agar isi dari buku tersebut tidak dibaca oleh orang lain. Namun kini menulis *Diary* dapat dilakukan dimanapun dan kapanpun tanpa harus membawa buku *Diary* kita, yaitu dengan smartphone atau perangkat lain.

Kini banyak orang yang menulis *Diary*nya di smartphone, akan tetapi kebanyakan isi catatan harian yang ditulis di smartphone hanya disimpan tanpa pengamanan. Jadi orang lain dapat dengan mudah membaca catatan harian yang harusnya bersifat pribadi. Dengan Algoritma kriptografi kita dapat mengamankan tulisan dengan proses enkripsi data. Proses Enkripsi adalah proses mengubah susunan tulisan bahkan merubah karater hurufnya secara acak dengan metode tertentu sehingga tulisan tidak dapat dibaca tanpa kunci tertentu untuk merapikan kembali tulisan tersebut. Hal ini mendorong saya untuk mendesain aplikasi *Diary* atau buku harian dengan algoritma kriptografi sehingga kerahasiaan dari isi *Diary* yang ditulis tetap terjaga.

Rumusan Masalah

Berdasarkan latar bekakang yang telah dikemukakan, maka permasalahan yang dapat dirumuskan adalah bagaimana merancang aplikasi *Diary* menggunakan alogritma kriptografi RC6 berbasis android.?

Batasan Masalah

Dalam pembuatan skripsi ini agar pembahasan masalah lebih fokus maka penulis memberikan batasan masalah, adapun batasan masalahnya adalah sebagai berikut :

1. Aplikasi ini ditujukan untuk pengguna dengan sistem operasi Android minimal Versi 2.3 (Gingerbread).
2. Algoritma yang digunakan untuk mengenkripsi dan mendeskripsikan teks adalah algoritma kriptografi RC6.
3. Software yang digunakan untuk membuat aplikasi adalah Eclips.
4. Aplikasi ini hanya membahas masalah kemanan tulisan berdasarkan aspek kamanan Kriptorafi yaitu confidentiallity dan integrity.

Tujuan penelitian

Tujuan dari penelitian ini dimaksudkan untuk membuat aplikasi *Diary* berbasis android dengan kerahasian isi *Diary* lebih baik karena tulisan yang dibuat akan dienkripsi dengan algoritma kriptografi RC6.

Landasan Teori

Diary

Diary atau buku harian adalah sebuah catatan pribadi yang berisi kegiatan sehari-hari. Buku harian ini bisa berisi kegitan apa saja tentang penulis atau orang lain. Misalnya, kejadian atau peristiwa yang dialami penulis setiap hari, pikiran atau permasalahan yang sedang dihadapi penulis setiap hari, dan apa saja yang ingin dituliskan ke dalam sebuah media.

Manfaat buku harian bermacam-macam. Diantaranya adalah untuk mendokumentasikan peristiwa atau kegiatan sehari-hari baik sebagai kenang-kenangan. Bisa juga digunakan untuk mencurahkan isi hati (curhat), obat stress, meluapkan emosi, menyampaikan keluh kesah, atau mengekspresikan pikiran ke dalam tulisan dan masih banyak lagi manfaat lain dari buku harian.[1]

Kriptografi

Algoritma Kriptografi

Algoritma kriptografi terbagi menjadi dibagi menjadi tiga bagian berdasarkan dari kunci yang digunakan yaitu Algoritma Simetri, Algoritma Asimetri dan Fungsi Hash. Algoritma Simetri adalah algoritma klasik karena menggunakan kunci yang sama untuk proses Enkripsi dan Dekripsinya. Algortima Simetri memakai kunci simetri diantaranya Data Encryption Standart (DES), International Data Encryption Algorithm (IDEA), Advenced Encryption Standart (AES), One Time Pad (OPT) RC2, RC3, RC4, RC5,dan RC6 dan lain sebagainya. [2]

Rivest Cipher 6 (RC6)

Algoritma RC6 merupakan salah satu kandidat Advanced Encryption Standard (AES) yang diajukan oleh RSA Security Laboratories kepada NIST. Dirancang oleh Ronald L Rivest, M.J.B. Robshaw, R. Sidney dan Y.L. Yin, algoritma ini merupakan pengembangan dari algoritma sebelumnya yaitu RC5 dan telah memenuhi semua kriteria yang diajukan oleh NIST. RC6 adalah algoritma yang menggunakan ukuran blok hingga 128 bit, dengan ukuran kunci yang diguankan bervariasi antara 128, 192 dan 256 bit. Algoritma RC6 dilengkapi dengan beberapa parameter, sehingga dituliskan sebagai RC6-w/r/b. Parameter w merupakan ukuran kata dalam satuan bit, parameter merupakan bilangan bukan negatif yang menunjukan banyaknya iterasi selama proses enkripsi dan parameter b menunjukan ukuran kunci enkripsi dalam byte. Setelah algoritma ini masuk dalam kandidat AES, maka ditetapkan bahwa nilai w = 32, r=20 dan b bervariasi antara 16, 24 dan 32 byte.

RC6-w/r/b memecah blok 128 bit menajdi 4 buah blok 32-bit, dan mengikuti aturan enam operasi dasar sebagai berikut :

a + b	operasi penjumlahan bilangan integer
a - b	operasi pengurangan bilangan integer
a □ b	operasi exclusive-OR (XOR)
a x b	operasi perkalian bilangan integer
a<<<b (b)	a dirotasikan ke kiri sebanyak variabel kedua
a>>>b (b)	a dirotasikan ke kanan sebanyak variabel kedua

Karena RC6 memecah blok 128 bit menjadi 4 buah blok 32 bit, maka algoritma ini bekerja dengan 4 buah register 32-bit A, B, C, D. Byte yang pertama dari plaintext atau ciphertext ditempatkan pada byte A, sedangkan byte yang terakhirnya ditempatkan pada byte D. Dalam prosesnya akan didapatkan (A, B, C, D) = (B, C, D, A) yang diartikan bahwa nilai yang terletak pada sisi kanan berasal dari register disisi kiri.[3].

2. Pembahasan

Analisis Kebutuhan Fungsional

Kebutuhan fungsional adalah jenis kebutuhan yangn menjelaskan fungsi-fungsi yang nantinya dapat dilakukan sistem. Adapun kebutuhan fungsional aplikasi *Diary* Kriptografi RC6 ini adalah sebagai berikut :

1. Sistem dapat melayani Registrasi Akun baru.
2. Sistem dapat melayani Login User.
3. Sistem dapat membuat *Diary* berupa text.
4. Sistem dapat melakukan Enkripsi text *Diary*.
5. Sistem dapat melakukan Dekripsi text *Diary*.
6. Sistem dapat menampilkan Daftar *Diary* yang tersimpan.
7. Sistem dapat Membaca *Diary* yang ditampilkan.
8. Sistem dapat melakukan edit dan delete *Diary* yang ditampilkan.

Analisis Kebutuhan Non- Fungsional

Kebutuhan Non Fungsional adalah jenis kebutuhan yang menjelaskan tentang kebutuhan diluar sistem seperti kebutuhan Operasional, Performance, Keamanan, Politik dan Budaya. Adapun kebutuhan Oprasional Aplikasi *Diary* Kriptografi RC6 ini adalah sebagai berikut :

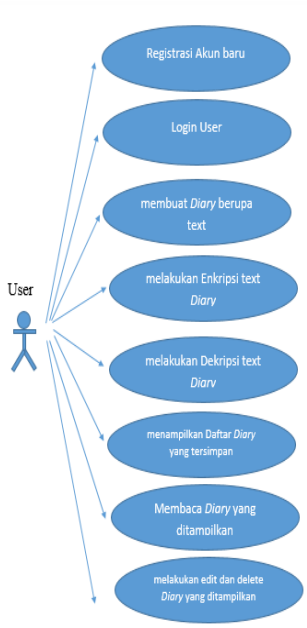
1. Kebutuhan Perangkat Keras
   Spesifikasi perangkat keras yang digunakan penulis dalam pembuatan Aplikasi *Diary* Kriptografi RC6 ini adalah sebagai berikut:
   1. Processor Intel Core i7 2.2 GHz.
   2. Memory (RAM) dengan kapasitas 4GB.
   3. Harddisk dengan kapasitas 750GB.
   4. VGA NVIDIA GeForce GT 630 M 2GB.
2. Kebutuhan minimal hardware dalam menjalankan sistem adalah:
   1. PC atau laptop dengan Processor 1GHz.
   2. Memory (RAM) dengan kapasitas 2GB.
   3. Harddisk dengan penyimpanan kosong 1GB.
   4. VGA Intel HD Graphic.
3. Kebutuhan minimal device yang digunakan untuk menjalankan aplikasi adalah:
   1. Smartphone android atau emulator dengan versi minimal 2.3 (Gingerbread).
   2. Memory (RAM) dengan kapasitas 512MB.
   3. Kebutuhan Perangkat Lunak

   Perangkat lunak yang digunakan dalam pembuatan Aplikasi *Diary* Kriptografi RC6 adalah sebagai berikut:
   1. Sistem operasi Windows 7 Ultimate 64Bit.
   2. Eclipse IDE.
   3. Android Software Development Kit (SDK).
   4. Android Development Tools (ADT).

Peracangan UML

Use Case Diagram

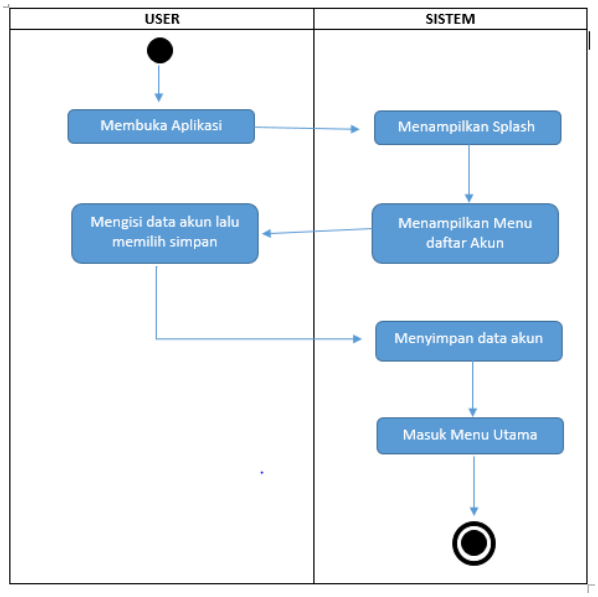
Use Case Diagram adalah metode berbasis teks untuk menggambarkan dan mendokumentasikan proses yang kompleks.



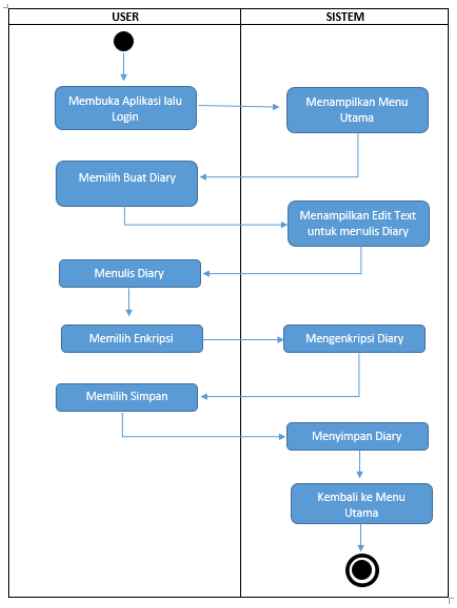
Gambar 3.1 *Use case* Diagram Aplikasi *Diary* Kriptografi RC6

Activity Diagram

Activity diagram menggambarkan rangkaian aliran dari aktifitas yang digunakan untuk mendeskripsikan aktifitas lainnya seperti use case atau interaksi.



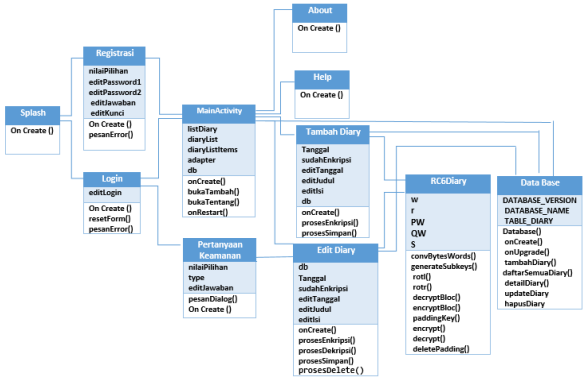
Gambar 3.2 *Activity* Diagram Menu Registrasi



Gambar 3.3 *Activity* Diagram Tulis *Diary*

Class Diagram

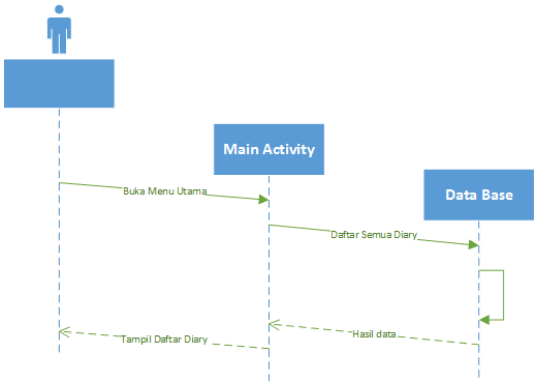
Class diagram memperlihatkan hubungan antara kelas satu dengan kelas yang lain dan penjelasan detail mengenai struktur aplikasi dan menampilkan atribut, operasi beserta method yang digunakan.



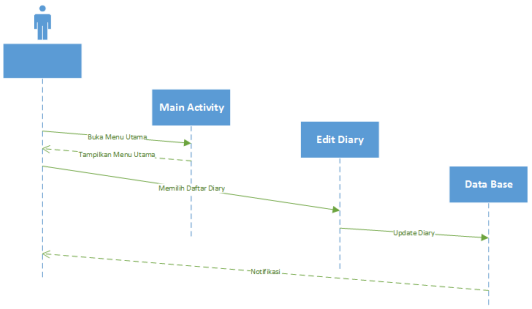
Gambar 3.4 *Class* Diagram Aplikasi *Diary* Kriptografi RC6

Sequence Diagram

Sequence diagram yaitu gambaran rangkaian dari langkah-langkah yang menggambarkan respon dari event pada aplikasi ini.



Gambar 3.5 *Sequence* Diagram Lihat Daftar Diary



Gambar 3.6 *Sequence* Diagram Edit *Diary*

Implementasi Kode Program dan Antar Muka
Kode Program Deklarasi dan Konversi RC6
Pada proses ini kode program berfungsi sebagai deklarasi dari perumpamaan perhitungan yang akan dilakukan oleh algoritma pada proses selanjutnya.

```
public class RC6 {
        // public RC6(){}
        private int w = 32, r = 20;
        private int Pw = 0xb7e15163, Qw = 0x9e3779b9;

        private int[] S;

        // konversi dari byte ke word
        private int[] convBytesWords(byte[] key, int u, int c) {
                int[] tmp = new int[c];
                for (int i = 0; i < tmp.length; i++)
                        tmp[i] = 0;

                for (int i = 0, off = 0; i < c; i++)
                        tmp[i] = ((key[off++] & 0xFF)) | ((key[off++] & 0xFF) << 8)
                                        | ((key[off++] & 0xFF) << 16) | ((key[off++] & 0xFF) << 24);

                return tmp;
        }
```

Gambar 4.1 Deklarasi dan Konversi RC6

Kode Program Penjadwalan Kunci
Penjadwalan kunci yang dilakukan pada kode program ini yaitu dengan menggabungkan antara array S dan L.

```
        private int[] generateSubkeys(byte[] key) {

                int u = w / 8;
                int c = key.length / u;
                int t = 2 * r + 4;

                int[] L = convBytesWords(key, u, c);

                int[] S = new int[t];
                S[0] = Pw;
                for (int i = 1; i < t; i++)
                        S[i] = S[i - 1] + Qw;

                int A = 0;
                int B = 0;
                int k = 0, j = 0;

                int v = 3 * Math.max(c, t);

                for (int i = 0; i < v; i++) {
                        A = S[k] = rotl((S[k] + A + B), 3);
                        B = L[j] = rotl(L[j] + A + B, A + B);
                        k = (k + 1) % t;
                        j = (j + 1) % c;
                }

                return S;
```

Gambar 4.2 Penjadwalan Kunci

Kode Program Pemecahan Blok Cipertext
Blok yang dipecah dalam program ini adalah blok cipertext dan pemecahannya dibagi menjadi empat register.

```
        private byte[] decryptBloc(byte[] input) {
                byte[] tmp = new byte[input.length];
                int t, u;
                int aux;
                int[] data = new int[input.length / 4];

                for (int i = 0; i < data.length; i++)
                        data[i] = 0;
                int off = 0;

                for (int i = 0; i < data.length; i++) {
                        data[i] = ((input[off++] & 0xff)) | ((input[off++] & 0xff) << 8)
                                        | ((input[off++] & 0xff) << 16)
                                        | ((input[off++] & 0xff) << 24);
                }

                int A = data[0], B = data[1], C = data[2], D = data[3];

                C = C - S[2 * r + 3];
                A = A - S[2 * r + 2];
                for (int i = r; i >= 1; i--) {
                        aux = D;
                        D = C;
                        C = B;
                        B = A;
                        A = aux;

                        u = rotl(D * (2 * D + 1), 5);
                        t = rotl(B * (2 * B + 1), 5);
                        C = rotr(C - S[2 * i + 1], t) ^ u;
                        A = rotr(A - S[2 * i], u) ^ t;
                }

                D = D - S[1];
                B = B - S[0];

                data[0] = A;
                data[1] = B;
                data[2] = C;
                data[3] = D;

                for (int i = 0; i < tmp.length; i++) {
                        tmp[i] = (byte) ((data[i / 4] >>> (i % 4) * 8) & 0xff);
                }

                return tmp;
        }
```

Gambar 4.3 Pemecahan Blok Cipertext

Kode Program Enkripsi
Seperti kita ketahui proses ini adalah proses yang menjadikan sebuah informasi tidak dapat diketahui isinya

```
public byte[] encrypt(byte[] data, byte[] key) {

        byte[] bloc = new byte[16];
        key = paddingKey(key);
        S = generateSubkeys(key);

        int lenght = 16 - data.length % 16;
        byte[] padding = new byte[lenght];
        padding[0] = (byte) 0x80;

        for (int i = 1; i < lenght; i++)
                padding[i] = 0;
        int count = 0;
        byte[] tmp = new byte[data.length + lenght];

        int i;
        for (i = 0; i < data.length + lenght; i++) {
                if (i > 0 && i % 16 == 0) {
                        bloc = encryptBloc(bloc);
                        System.arraycopy(bloc, 0, tmp, i - 16, bloc.length);
                }

                if (i < data.length)
                        bloc[i % 16] = data[i];
                else {
                        bloc[i % 16] = padding[count];
                        count++;
                        if (count > lenght - 1)
                                count = 1;
                }
        }
        bloc = encryptBloc(bloc);
        System.arraycopy(bloc, 0, tmp, i - 16, bloc.length);
        return tmp;
}
```

Gambar 4.4 Enkripsi

Kode Program Dekripsi

Dekripsi adalah pengembalian dari Fungsi Enkripsi sehingga informasi yang awalnya disembunyikan dapat dilihat kembali.

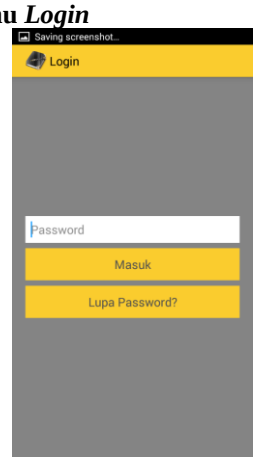
```
public byte[] decrypt(byte[] data, byte[] key) {
        byte[] tmp = new byte[data.length];
        byte[] bloc = new byte[16];
        key = paddingKey(key);
        S = generateSubkeys(key);

        int i;
        for (i = 0; i < data.length; i++) {
                if (i > 0 && i % 16 == 0) {
                        bloc = decryptBloc(bloc);
                        System.arraycopy(bloc, 0, tmp, i - 16, bloc.length);
                }

                if (i < data.length)
                        bloc[i % 16] = data[i];
        }

        bloc = decryptBloc(bloc);
        System.arraycopy(bloc, 0, tmp, i - 16, bloc.length);

        tmp = deletePadding(tmp);
        return tmp;
}
```

Gambar 4.5 Dekripsi

Tampilan *Splash Screen*

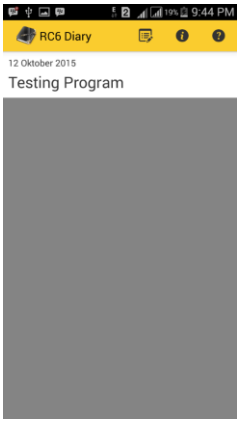
Gambar 4.6 Antar Muka *Splash Screen*

Tampilan Menu *Login*



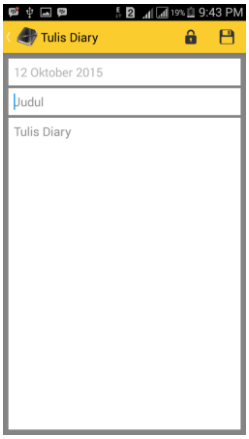
Gambar 4.7 Tampilan Menu *Login*

Tampilan Manu Utama



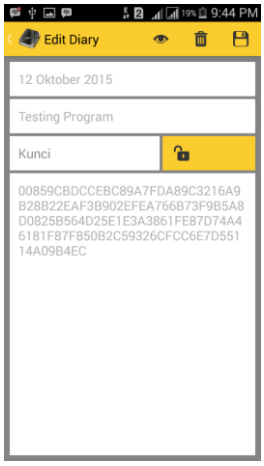
Gambar 4.8 Tampilan Menu Utama

Tampilan Tambah *Diary*



Gambar 4.9 Tampilan Tambah *Diary*

Tampilan Menu *Edit*



Gambar 4.10 Tampilan Menu *Edit*

3. Kesimpulan

Setelah melalui tahapan tahapan yang telah dijelaskan pada pembahasan sebelumnya makan dapat ditarik kesimpulan tentang Aplikasi *Diary* Kriptografi RC6 ini adalah:

Aplikasi *Diary* Kriptografi RC6 ini memiliki fitur pengamanan informasi ganda yaitu Login Password dan Enkripsi isi *Diary* jadi penulis dapat merahasiakan isi dairy lebih baik. Aplikasi *Diary* Kriptografi RC6 ini mengunakan algoritma kriptografi RC 6 dengan ukuran blok 128 bit dan ukuran kunci 128. Setelah melakukan pengujian pada program didapatkan hasil yang memuaskan dimana program berfungsi secara maksimal sesuai rancangan dan kapasitasya.

Berikut beberapa saran yang dapat dipergunakan sebagai pertimbangan untuk pengembangan aplikasi pada penelitian selanjutnya.

1. Aplikasi membutuhkan pengembangan lebih lanjut agar aplikasi lebih sempurna dan terhindar dari berbagai macam bug dan Error.
2. Pada pengembangan selanjutnya diharapkan dapat menambahkan fitur-fitur yang dapat memudahkan pengguan terutama saat mebuat dan mengenkripsi *Diary* yang ditulis serta fitur backup data yang terhubung pada e-mail pengguna.

Daftar Pustaka

[1] Cipir, Jurang.”Pengertian Buku Harian dan Manfaatnya”. juragancipir.com/pengertian-buku-harian-dan-manfaatnya/ (Diakses 17 Maret 2015

[2] Ariyus, Dony. 2006. Kriptografi : Keamanan Data dan Komunikasi. Yogyakarta: Penerbit Andi

[3] Abdurohman, Maman. 2002, “ Analisis Performansi Algoritma Kriptografi RC6 ”, https://ml.scribd.com/doc/75063444/Enkripsi. (Diakses 25 Maret 2015

Biodata Penulis

Aedhoh Salim Assaidi, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Teknik Informatika STMIK AMIKOM Yogyakarta, lulus tahun 2015.

Armadyah Amborowati, memperoleh gelar Sarjana Komputer (S.Kom), Jurusan Sistem Informasi, STMIK AMIKOM Yogyakarta. Memperoleh gelar Magister Enginnering (M.Eng), Program Pasca Sarjana Magister Teknologi Informasi Fakultas Teknik Elektro Universitas Gajah Mada Yogyakarta. Saat ini menjadi Dosen di STMIK AMIKOM Yogyakarta.