KALBIScientia
Jurnal Sains dan Teknologi

ISSN 2356 - 4393

Implementasi Algoritma Rivest, Shamir, Adleman Untuk File Sharing Pada PT. Sumber Makmur Pangan Sejahtera Berbasis Web

Halim Agung[1)], Irne Prasta[2)]

Teknik Informatika, Universitas Bunda Mulia
Jalan Lodan Raya No.2, Pademangan, Jakarta Utara, 14430

[1)]Email: hagung@bundamulia.ac.id,

[2)]Email: irneprasta@yahoo.com

Astract: *The process of file sharing is a data sharing activity that is often done in a company, the data to be shared sometimes there is very important information in it so it needs to do data security to keep important information. The algorithm used in this research is Rivest, Shamir, Adleman (RSA) algorithm. The result of this research is can give solution to the problem of file sharing activity done at PT. Source Prosperous Food Prosperous with the establishment of a file sharing application using RSA algorithm and container files that have been encrypted in the dropbox. it is concluded that the Rivest, Shamir, Adleman algorithm can be used for securing the data to be sent in the file sharing process with the data container in the dropbox.*

Keywords: *cryptography, decryption, encryption, RSA*

Abstrak: *Proses file sharing merupakan kegiatan berbagi data yang sering dilakukan dalam sebuah perusahaan, data yang akan dibagikan terkadang terdapat informasi yang sangat penting di dalamnya sehingga perlu dilakukan pengamanan data untuk menjaga informasi penting. Algoritma yang digunakan dalam penelitian ini adalah algoritma Rivest, Shamir, Adleman (RSA). Hasil dari penelitian ini adalah dapat memberikan solusi terhadap permasalahan kegiatan file sharing yang dilakukan pada PT. Sumber Makmur Pangan Sejahtera dengan terbentuknya pembuatan sebuah aplikasi file sharing dengan menggunakan algoritma RSA dan tempat penampung file yang sudah terenkripsi di dropbox. Disimpulkan bahwa algoritma RSA dapat digunakan untuk pengamanan data yang akan dikirimkan dalam proses file sharing dengan tempat penampung data di dropbox.*

Kata kunci: *dekripsi, enkripsi, file sharing, kriptografi, RSA*

I. PENDAHULUAN

Masalah keamanan dan kerahasiaan data merupakan hal yang penting dalam suatu organisasi. Data yang bersifat rahasia perlu dibuatkan sistem agar penyimpanan dan pengirimannya tidak terbaca dan tidak dapat diubah oleh pihak yang tidak memiliki hak atas data tersebut, baik saat data tersebut tersimpan sebagai file di dalam komputer, saat data tersebut dikirim melalui e-mail ataupun penyedia data online seperti dropbox.

Proses pengiriman data yang dilakukan PT. Sumber Makmur Pangan Sejahtera selama ini masih bersifat manual menggunakan flashdisk atau e-mail apabila data yang akan dibagi kepada orang yang ingin diberikan ini bersifat rahasia, dengan tujuan agar data tersebut dapat aman dari pihak yang tidak berhak mengetahui isi dari data yang bersifat rahasia tersebut. Proses file sharing yang dilakukan dalam perusahaan dianggap kurang efektif apabila terjadi pada saat situasi untuk dapat memperoleh data yang dibutuhkan secara cepat dan informasi dalam data dapat terjaga dengan aman. Algoritma yang akan digunakan dalam penelitian ini adalah Algoritma Rivest, Shamir, Adleman (RSA) untuk diimplementasikan dalam proses file sharing pada PT. Sumber Makmur Pangan Sejahtera.

Penelitian ini menggunakan beberapa penelitian yang dijadikan sebagai referensi. Referensi pertama adalah mengenai aplikasi e-mail client dengan manfaat memberikan sarana komunikasi tersendiri yang aman [1]. Referensi kedua adalah mengenai implementasi kriptografi kunci publik dengan Algoritma RSA-CRT pada splikasi instant messaging yang mendapatkan hasil yaitu mengelola server dan client dengan port, mengelola kunci dan isi

pesan singkat menjadi aman dengan dienkripsi [2]. Referensi ketiga adalah mengenai aplikasi web untuk e-voting menggunakan algoritma RSA, yang dapat dipergunakan oleh mahasiswa [3]. Referensi keempat adalah mengenai perancangan sistem penunjang keputusan biaya kebutuhan mahasiswa dengan waktu tercepat melalui metode Backward Chain dan Algoritma RSA yang mendapatkan hasil membantu para mahasiswa mendapatkan informasi tentang berapa banyak kebutuhan dalam jangka waku sebulan, membantu pengguna mengetahui tips mengenai kebutuhan apa saja yang seharusnya dibutuhkan, mengetahui seberapa besar pengeluaran, dari data yang dihasilkan tersebut diterapkan sistem pengaman data dengan dienkripsi [4]. Referensi kelima adalah mengenai aplikasi laporan keuangan akuntansi bulog jakarta menggunakan algoritma MD5 dan RSA [5]. Referensi keenam adalah mengenai algoritma RSA yang digunakan sebagai alat kriptografi menggunakan hybrid cryptosystem dan digital signature [6].

II. METODE PENELITIAN

Prosedur penelitian ini dilakukan dengan beberapa tahapan pengembangan perangkat lunak yang menggunakan metode proses pengembangan air terjun (*waterfall model*). Alasan penggunaan metode *waterfall model* dalam pembuatan sistem ini yaitu tahapan pada model ini mengambil kegiatan dasar yang digunakan dalam hampir semua pengembangan perangkat lunak, sehingga dapat mudah untuk dipahami terlebih bila hanya digunakan dalam mengembangkan perangkat lunak yang tidak begitu besar dan kompleks [8].

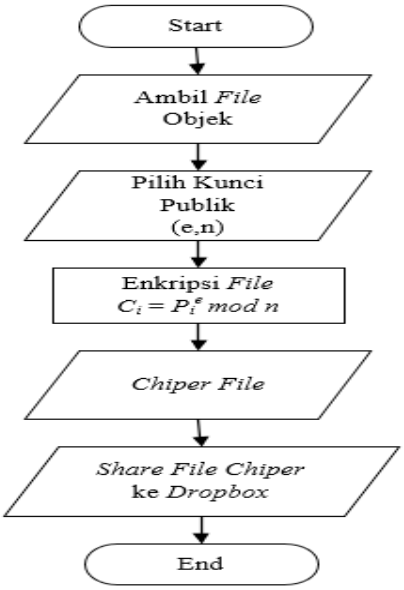
Kriptografi adalah ilmu yang mempelajari teknik matematika yang berhubungan dengan aspek keamanan informasi seperti kerahasiaan, integritas data, otentikasi entitas, dan otentikasi asal data. Kriptografi bertujuan agar informasi yang bersifat rahasia dan dikirim melalui suatu jaringan, seperti LAN atau *internet*, tidak dapat diketahui dan dimanfaatkan oleh orang lain atau pihak yang tidak berkepentingan [2].

RSA merupakan algoritma kriptografi asimetri, dimana kunci yang digunakan untuk mengenkripsi berbeda dengan yang digunakan untuk mendekripsi. Kunci yang digunakan untuk mengenkripsi disebut dengan kunci publik, dan yang digunakan untuk mendekripsi disebut dengan kunci privat. Algortima RSA dijabarkan pada tahun 1976 oleh tiga orang yaitu Ron Rivest, Adi Shamir dan Len Adleman dari Massachusetts Institute of Technology. Huruf ‘’’RSA’’’ itu sendiri berasal dari inisial nama mereka (‘R’ivest -‘S’hamir -‘A’dleman). Clifford Cocks, seorang matematikawan Inggris yang bekerja untuk GCHQ, menjabarkan tentang sistem equivalen pada dokumen internal di tahun 1973. Penemuan Clifford Cocks tidak terungkap hingga tahun 1997 karena alasan ‘’top- secret classification’’. Algoritma RSA dipatenkan oleh Massachusetts Institute of Technology pada tahun 1983 di Amerika Serikat sebagai US patent 4405829. Patent tersebut berlaku hingga 21 September 2000. Setelah bulan September tahun 2000, paten tersebut berakhir, sehingga saat ini semua orang dapat menggunakannya dengan bebas. Algoritma RSA terbagi kedalam tiga bagian utama yaitu Proses Pembuatan Kunci (Private dan Public Keys), Proses Enkripsi (Encrypt) dan Proses Dekripsi (Decrypt) [7].

Proses perhitungan algoritma Rivest Shamir Adleman (RSA) [8] yaitu menentukan dua buah bilangan prima, dimisalkan dengan p dan q kemudian menghitung modulus kunci publik dengan N = p * q dimana Nilai p dan q merupakan bilangan prima yang dipilih secara acak dengan syarat p≠q dan terpisah untuk tiap-tiap p dan q. hitung N = p*q. Nilai N merupakan hasil dari perkalian nilai p dan q. Kemudian φn = (p - 1) x (q - 1) untuk menghitung nilai m atau φn, digunakan untuk melakukan pengujian layak atau tidak layak untuk digunakan sebagai kunci. Hasil pengujian akan didapat jika nilai dari p dan q saling prima.

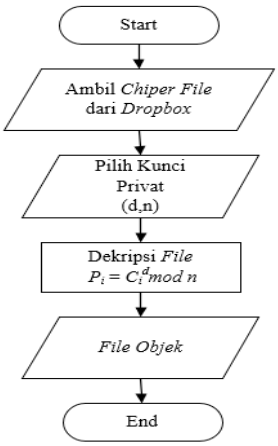
Kemudian GCD (e, φ(n)) = 1 untuk penentuan nilai e dimana untuk menghitung nilai e yang dipilih merupakan bilangan bulat atau integer, hasil koprima dari φn sama dengan 1. Kemudian proses menghitung modulus kunci privat dengan rumus (e.d) mod φn = 1yang digunakan untuk menentukan nilai d yang dapat dihasilkan dari modulus nilai e dan φn dengan hasil yang didapat sama dengan 1. Kemudian dilanjutkan proses enkripsi, dengan rumus Ci = Pie mod n dan yang terakhir adalah rumus dekripsi dengan rumus Pi = Cid mod n dimana p = bilangan prima pertama, q = bilangan prima kedua, n = hasil perkalian bilangan prima, GCD = greatest common divisor (pembentukan kunci), c_i = ciphertext, p_i = plaintext, e = kunci publik, d = kunci privat.

Flowchart adalah bagian-bagian yang menggambarkan langkah-langkah menyelesaikan suatu masalah. Flowchart merupakan suatu penyajian dari algoritma [9]. Perancangan flowchart diagram bertujuan untuk mengGambarkan aliran proses dalam sistem. Flowchart diagram dari aplikasi ini ditunjukkan pada Gambar 1 dan 2.



Gambar 1 Flowchart enkripsi

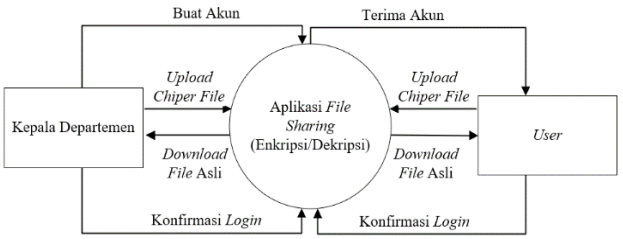
Gambar 1 menunjukkan *flowchart* atau alur proses dari enkripsi dari sistem yang dibuat dari penelitian ini dimana tahap pertama adalah aplikasi akan meminta untuk diberikan objek file yang akan dienkripsi, kemudian memilih kunci publik yang sudah digenerate sebelumnya, setelah itu aplikasi akan mengenkripsi *file* tersebut dan menghasilkan *chiper file.* Setelah itu *chiper file* tersebut dibagikan ke *dropbox.*



Gambar 2 Flowchart dekripsi

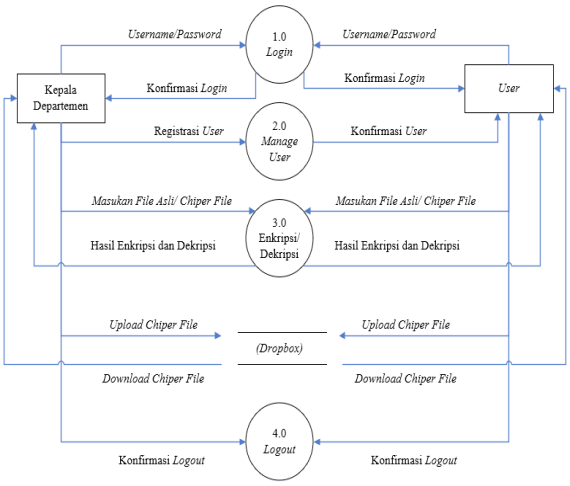
Gambar 2 adalah alur proses atau flowchart dekripsi dari aplikasi yang dihasilkan pada penelitian ini, proses dimulai dari pengambilan file chiper yang disimpan sebelumnya pada dropbox, kemudian setelah itu aplikasi akan diminta untuk memilih kunci privat yang sudah di generate sebelumnya, setelah itu aplikasi akan mendekripsi file tersebut menjadi file utuh yang sebelumnya di enkripsi.

Data Flow Diagram (DFD) adalah sebuah alat yang mengGambarkan aliran data sampai sebuah sistem selesai, dan kerja atau proses dilakukan dalam sistem tersebut [10]. Aliran data pada aplikasi file sharing ditunjukan melalui Data flow diagram. Data flow diagram pada aplikasi ini terdiri dari 3 bagian yaitu diagram konteks yang mengGambarkan aplikasi secara umum, diagram level 0 yang mengGambarkan rincian dari diagram konteks dan diagram level 1 yang mengGambarkan rincian lebih detail.



Gambar 3 Diagram konteks pengguna aplikasi

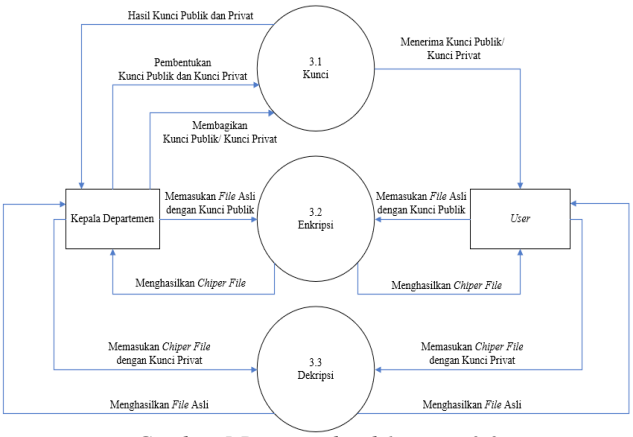
Pada Gambar 3 menunjukkan Gambaran umum proses login pengguna aplikasi saat akan menggunakan file sharing, kepala departemen dapat membuat akun dan User hanya menerima akun dari kepala departemen. Kepala departemen dan user dapat melakukan proses enkripsi dan dekripsi terhadap file yang diinginkan pada aplikasi file sharing. Kepala departemen dan user juga dapat menyimpan atau mengunduh cipher file yang telah diproses enkripsi atau yang akan diproses dekripsi dan yang disimpan pada dropbox.



Gambar 4 Diagram level 0

DFD level 0 digunakan untuk mengGambarkan proses utama yang terjadi pada sistem, pada Gambar 4 menunjukan kepala departemen dan user memiliki username dan password untuk proses login saat ingin menggunakan aplikasi file sharing, pada aplikasi terdapat proses untuk manage user yang hanya terdapat pada pengguna aplikasi berstatus kepala departemen, dimana manage user ini berfungsi untuk mengelola siapa saja yang akan menjadi pengguna aplikasi. Kepala departemen dan user dapat melakukan proses

enkripsi dan dekripsi pada aplikasi terhadap file yang diinginkan. User juga dapat membagikan file secara langsung antar pengguna melalui aplikasi, entah file yang ingin dibagikan merupakan file yang belum diamankan maupun file yang telah di proses enkrpsi atau dekripsi. Setelah selesai menggunakan aplikasi, kepala departemen dan user dapat langsung logout dari aplikasi.



Gambar 5 Diagram level 1 proses 3.0

Gambar 5 menunjukkan tahap yang menjelaskan proses enkripsi dan dekripsi yang dilakukan pada aplikasi *file sharing,* yang dilakukan diawal proses adalah pembentukan kunci yang merupakan syarat utama dalam melakukan proses enkripsi ataupun dekripsi. Hanya kepala departemen yang dapat melakukan pembentukan kunci pada aplikasi *file sharing,* kunci yang akan dibentuk terbagi dua yaitu kunci publik untuk proses enkripsi dan kunci privat untuk proses dekripsi, setelah kepala departemen melakukan pembentukan kunci maka akan dibagikan kepada *user* atau orang yang akan diberikan hak akses dalam melakukan proses enkripsi ataupun dekripsi.

III. HASIL DAN PEMBAHASAN

Algoritma yang digunakan dalam penelitian ini adalah algoritma *Rivest Shamir Adleman* (RSA). Algoritma tersebut dipilih karena algoritma ini menggunakan *public key* dan *private key* untuk melakukan proses enkripsi dan dekripsi. Kekuatan algoritma ini terletak pada proses pemfaktoran bilangan menjadi 2 bilangan prima yang hingga kini perlu waktu yang lama untuk melakukan pemfaktorannya. Tujuan utama dari algoritma ini untuk mengamankan dokumen penting sehingga data tidak mudah diakses oleh pihak-pihak tidak berkepentingan yang tidak memiliki hak akses.

Dalam mengimplementasikan algoritma RSA, terlebih dahulu dilakukan pembuatan sepasang kunci, yaitu kunci publik dan kunci privat. Sebelum membuat kunci, dilakukan penentuan bilangan prima terlebih dahulu yang dapat dilihat pada Gambar 6.

```
$KEY_SIZE =256;
    $key = array();
    $rand1=rand(($KEY_SIZE / 2), 100);
    $rand2=rand(($KEY_SIZE / 2), 100);

    // mencari bilangan prima selanjutnya dari $rand1 &rand2
    $p = gmp_nextprime($rand1);
    $q = gmp_nextprime($rand2);
```

Gambar 6 Source code penentuan bilangan prima

Pada penentuan bilangan prima dilakukan secara random, dengan *source code* terdapat pada Gambar 6, perhitungan key size ditentukan dengan perkalian 256 *byte*. Angka secara acak oleh sistem komputer yang nantinya ditampung pada perhitungan untuk mendapat nilai p dan q.

```
//menghitung n=p*q
    $n=gmp_mul($p,$q);
    //menghitung totient/phi=(p-1)(q-1)
    $totient=gmp_mul(gmp_sub($p,1),gmp_sub($q,1));
    //echo "phi = ".rand((count(array_map('intval', str_split(decbin($totient))))), 1000)."<br/>";
    //mencari e, dimana e merupakan coprime dari totient
    //e dikatakan coprime dari totient jika gcd/fpb dari e dan totient/phi = 1
    for($e=2;$e<100;$e++){  //mencoba perulangan max 100 kali,
        $gcd = gmp_gcd($e, $totient);
        //echo $e."<br/>";
        if(gmp_strval($gcd)=='1')
            break;
    }
    //cari d
    // d.e mod totient =1
    // d.e = totient*x + 1
    // d.e = totient*1 + 1
    // d = (totient *1 + 1)/e
    //menghitung&menampilkan d
    $i=1;
    do{
        $res = gmp_div_qr(gmp_add(gmp_mul($totient,$i),1), $e);
        //echo '((totient*'.$i.') + 1) / e='.gmp_strval($res[0])." ; sisa= ".gmp_strval($res[1])."\n";
        //echo "<pre>";
        //print_r(gmp_strval($res[0],16))."<br/>";
        //echo "</pre>";
        $i++;
        if($i==10000) //maksimal percobaan 10000
            break;
    }
    while(gmp_strval($res[1])!='0');
    $d=$res[0];
    //echo $d."<br/>";
    $key['n'] = gmp_strval($n);
    $key['e'] = gmp_strval($e);
    $key['d'] = gmp_strval($d);
    //$key['n'] = gmp_strval($n,16);
    //$key['e'] = gmp_strval($e,16);
    //$key['d'] = gmp_strval($d,16);
    return $key;
}
```

Gambar 7 Source pembentukan kunci

Pembentukan kunci dapat dilihat pada Gambar 7, setelah nilai bilangan prima p dan q dihasilkan, untuk pembentukan kunci dimulai dengan perhitungan variabel "n", "e", dan "d". dimana variabel "n" ditampung dengan hasil perkalian antara "p" dan "q". Kemudian pencarian nilai dari variabel "e" sebagai *public key* dan nilai variabel d sebagai *private key*

Proses enkripsi dapat dilihat pada Gambar 8, proses enkripsi dapat dilakukan apabila *file* yang ingin dienkripsi sudah dimasukan dan kunci publik sudah dipilih. Proses enkripsi dijalankan dengan rumus M^e mod n, dimana M adalah representasi naskah asli (bilangan bulat) ketika dijalankannya

proses enkripsi. Setelah proses enkripsi selesai maka akan menghasilkan *cipher file*.

```
function encrypt($key, $file){
        $result = "";
        //echo strlen($file); die;
        for($i=0;$i<strlen($file);++$i){
            //rumus enkripsi <enkripsi>=<pesan>^<e>mod<n>
            $result.=gmp_strval(gmp_mod(gmp_pow(ord($file[$i]), $key['e']), $key['n']));
            //echo ord($file[$i])."-".gmp_strval(gmp_mod(gmp_pow(ord($file[$i]), $key['e']), $key['n'])."<br/>";
            //$r = gmp_strval(gmp_mod(gmp_pow(ord($file[$i]), $key['e']), $key['n']));
            //$rr = gmp_strval($r,16);
            //$d= chr(gmp_strval(gmp_mod(gmp_pow($r, $key['d']), $key['n'])));
            //echo "encrypt = ".$r."<br/>";
            //echo $key['e']."<br/>";
            //echo "decrypt = ".$d."<br/>";
            //antar tiap karakter dipisahkan dengan "."
            if($i!=strlen($file)-1){
               $result.=".";
            }
        }
        //die;
        return $result;
    }
```

Gambar 8 Source code enkripsi

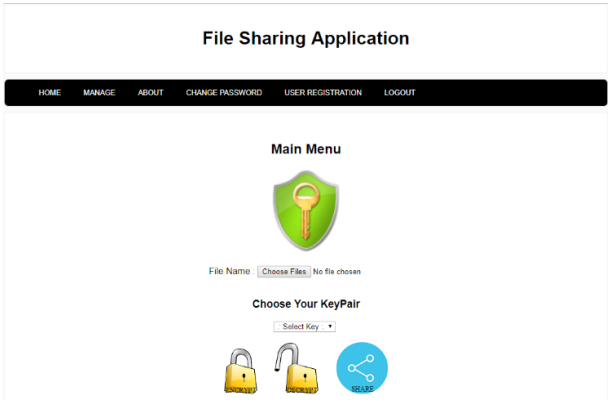
```
function decrypt($key, $file){
        $result = "";
        $file=explode(".",$file);
        //echo count($file); die;
        foreach($file as $nilai){
            //rumus dekripsi <pesan>=<enkripsi>^<d>mod<n>
            $result.=chr(gmp_strval(gmp_mod(gmp_pow($nilai, $key['d']), $key['n'])));
            //echo $result;
        }
        // die;
        return $result;
    }
```

Gambar 9 Source code dekripsi

Proses Dekripsi dapat dilihat pada Gambar 9, proses dekripsi dapat dilakukan apabila *cipher file* dari hasil enkripsi sebelumnya sudah dimasukkan dan kunci privat sudah dipilih. Proses dekripsi dijalankan dengan rumus M^d mod n, dimana M adalah representasi naskah asli (bilangan bulat) ketika dijalankannya proses dekripsi. Setelah proses dekripsi selesai maka *file* yang digunakan sebagai objek enkripsi akan kembali seperti *file* aslinya saat *file* tersebut belum dilakukan proses enkripsi.

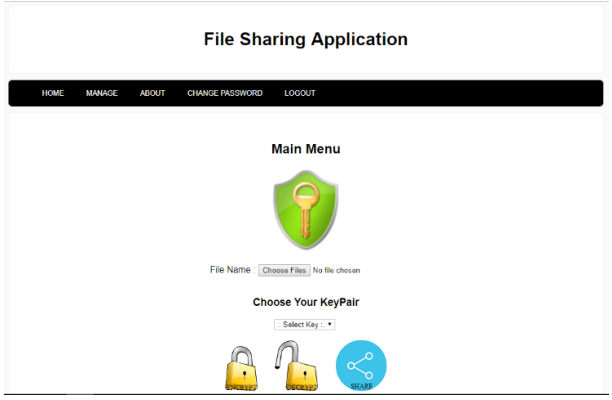
Gambar 10 Tampilan awal program

Gambar 10 merupakan Gambar halaman awal pengguna aplikasi saat akan masuk ke aplikasi file sharing. Pengguna aplikasi terdiri dari dua yaitu admin atau kepala departemen dan user, dimana dari kedua pengguna ini harus memiliki username dan memiliki password untuk dapat mengakses aplikasi file sharing ini.



Gambar 11 Tampilan home pada admin

Gambar 11 merupakan Gambar halaman home pada admin atau kepala departemen, pada halaman ini admin atau kepala departemen dapat melalukan proses sharing file dengan teknik pengamanan data terlebih dahulu, dengan cara admin atau kepala departemen memasukan file yang ingin dienkripsi pada menu file name dengan klik choose file yang akan langsung terhubung dengan tempat penyimpanan dokumen yang diinginkan di dalam PC, kemudian admin atau kepala departemen harus memilih kunci publik yang harus dibuat terlebih dahulu pada menu manage untuk proses enkripsi, setelah proses enkripsi selesai maka akan menghasilkan cipher file yang dapat di share kepada orang yang ingin diberikan hak akses, cipher file tersebut akan dibagikan melalui tempat penyimpanan dropbox. Jika admin atau kepala departemen ingin mengambil file yang dikirimkan oleh orang lain, admin atau kepala departemen harus mengambil cipher file yang terdapat pada dropbox lalu memilih kunci privat yang sudah diberikan oleh orang yang ingin mengirim dokumen tersebut.



Gambar 12 Tampilan home pada user

Gambar 12 merupakan Gambar halaman home pada user, pada halaman ini user dapat melalukan proses sharing file dengan teknik pengamanan data terlebih dahulu, dengan cara user memasukan file yang ingin dienkripsi pada menu file name dengan klik choose file yang akan langsung terhubung dengan tempat penyimpanan dokumen yang diinginkan

di dalam PC, kemudian user harus memilih kunci publik yang sudah diberikan oleh admin atau kepala departemen terlebih dahulu karena yang dapat manage key hanyalah pengguna aplikasi yang berstatus sebagai admin/ kepala departemen, setelah proses enkripsi selesai maka akan menghasilkan cipher file yang dapat dibagikan kepada orang yang ingin diberikan hak akses, cipher file tersebut akan dibagikan melalui tempat penyimpanan dropbox. Jika user ingin mengambil file yang dikirimkan oleh orang lain, user harus mengambil cipher file yang terdapat pada dropbox lalu memilih kunci privat yang sudah diberikan oleh orang yang ingin mengirim dokumen tersebut, kunci privat yang dimiliki oleh user untuk proses dekripsi hanya dapat satu kali pemakaian saja, setelah kunci privat terpakai maka kunci akan secara otomatis akan hilang dari daftar kunci yang dimiliki oleh user.

Pengujian algoritma melalui perhitungan manual yaitu dengan menggenerate kunci publik dan kunci privat, pertama dengan menentukan 2 bilangan prima, misal p = 5 dan 7, menentukan pemilihan angka bilangan prima secara acak, misal angka yang dipilih untuk bilangan prima pertama p adalah 5 dan untuk bilangan prima kedua adalah 7, kedua dengan menghitung modulus kunci publik dimana n = p*q; n = 5*7 = 35 kemudian menghitung nilai n dengan mengkalikan 2 bilangan prima yang sudah ditentukan, bilangan prima pertama yaitu p = 5 dan bilangan prima kedua yaitu q = 7 maka nilai n adalah 5 x 7 = 35. Berikutnya φn = (p - 1) x (q - 1); φn = (5 - 1) x (7 - 1) = 24 dimana menghitung φn dengan menggunakan rumus (p - 1) x (q - 1), diketahui sebelumnya p = 5 dan q = 7, maka nilai φn = (5 - 1) x (7 - 1) = 4 x 6 = 24. e ditentukan random nilai integer dengan syarat e > 1 sehingga tidak bisa membagi rata d dan GCD (e,24) = 1, misal e = 5. Kemudian GCD (e, φ(n)) = 1; GCD (5,24) = 1. Berikutnya pembuktian apakah nilai GCD (5,24) = 1 ? 24 mod 5 = 4, 5 mod 4 = 1, 4 mod 1 = 0 dan ternyata benar GCD (5,24) = 1 (1 didapat dari angka yang diberi warna biru). Berarti diperbolehkan nilai dari e = 5. Kemudian menghitung modulus kunci privat, (e.d) mod φn = 1; 5(d) mod 24 = 29 dan pembuktian apakah (e.d) mod φn = 1 ? (5*29) mod 24 = 1 ? 145 mod 24 = 1 dan ternyata benar (5*29) mod 24 = 1. Berarti persyaratan sudah terpenuhi maka diperbolehkan nilai d = 29.

Sehingga hasil dari perhitungan di atas memperoleh kunci publik (n,e) = (35, 5) dan kunci Privat (n,d) = (35,29) setelah itu proses enkripsi dengan $C_i = P_i^e \bmod n$; Jika plaintext bernilai 23, $C_{i=} 23^5 \bmod 35 = 18$. Kemudian proses dekripsi dengan $P_i = C_i^d \bmod n$; Jika ciphertext bernilai 18, $P_{i=} 18^{29} \bmod 35 = 23$.

Tabel 1 Pengujian enkripsi aplikasi

No.	Nama *File*	Ukuran *File* (Kilo *Byte*)	Waktu Enkripsi (Menit-Detik-Milidetik)
1	logo.png	8	00:02.58
2	daftar kontak customer.xlsx	10	00:01.42
3	surat keterangan.docx	13	00:02.01
4	data penjualan harian 2 april 17.xlsx	14	00:02.48
5	surat tugas.docx	27	00:02.12
6	laporan keuangan.xlsx	49	00:02.32
7	penilaian customer.pdf	67	00:03.08
8	struktur organisasi.png	123	00:03.21
9	Stock Opname.xlsx	140	00:01.10
10	design ine.png	149	00:01.12
11	denah lokasi.jpg	207	00:01.32
12	barang pesanan.jpg	232	00:01.44
13	peningkatan kualias kerja.pdf	330	00:01.59
14	perjanjian kontrak.pdf	337	00:02.05
15	bakmi special.pdf	532	00:02.14
16	stock opname pabrik.pdf	620	00:02.31
17	stock opname perusahaan.pdf	652	00:02.45
18	Laporan tahunan divisi finance.docx	654	00:03.25
19	pangsit super.pdf	727	00:03.43
20	list suplier.pdf	823	00:03.58
21	acara akhir tahun.pdf	954	00:04.13
22	design produk.pdf	1391	00:06.43
23	alur pengiriman orderan.docx	2259	00:09.54
24	susunan acara gathering.doc	3525	00:12.42
25	list gaji.docx	4219	00:14.21
26	list bonus dan thr.docx	4306	00:14.53
27	sejarah perusahaan.pdf	4467	00:19.21
28	ringkasan meeting 022020.doc	6706	00:23.43
29	data karyawan.docx	7900	00:27.02
30	lap thnan divisi marketing.docx	7984	00:27.41

Tabel 2 Pengujian dekripsi aplikasi

No.	Nama *File*	Ukuran *File* (Kilo *Byte*)	Waktu Enkripsi (Menit-Detik-Milidetik)
1	logo.edr	38	00:08:04
2	daftar kontak customer.edr	43	00:09.39
3	surat keterangan.edr	56	00:11.45
4	data penjualan harian 2 april 17.ed	61	00:15.59
5	surat tugas.edr	129	00:26.19
6	laporan keuangan.edr	238	00:45.44
7	penilaian customer.edr	333	01:03.45
8	design ine.png	537	01:28.15
9	struktur organisasi.edr	614	01:47.44
10	Stock Opname.xlsx	716	02:08.25
11	denah lokasi.jpg	1065	02:46.03
12	barang pesanan.edr	1164	03:36.50
13	perjanjian kontrak.edr	1653	03:54.41
14	peningkatan kualias kerja.pdf	1710	04:28.49
15	bakmi special.pdf	2741	04:40.43
16	stock opname pabrik.pdf	3209	05:41.19
17	Laporan tahunan divisi finance.do	3223	05:45.06
18	stock opname perusahaan.pdf	3374	05:58.10
19	pangsit super.pdf	3773	05:57.32
20	list suplier.pdf	4249	07:37.51
21	acara akhir tahun.pdf	4943	08:35.06
22	design produk.pdf	7195	09:52.03
23	alur pengiriman orderan.docx	11653	10:53.28
24	susunan acara gathering.doc	17933	15:32.54
25	list gaji.docx	21492	28:10.12
26	list bonus dan thr.docx	21943	29:23.42
27	sejarah perusahaan.pdf	22649	32:54.12
28	ringkasan meeting 022020.doc	33828	52:34.21
29	data karyawan.docx	36275	61:56.09
30	lap thnan divisi marketing.docx	36659	62:34.35

Dari hasil tabel 1 dan 2 dapat di simpulkan bahwa lama waktu enkripsi juga dipengaruhi oleh besar kecilnya sebuah *file*, apabila *file* semakin besar maka proses enkripsi akan memakan waktu yang lebih lama begitupun dekripsi.

IV. SIMPULAN

Berdasarkan hasil uji coba aplikasi dan uraian bab-bab sebelumnya, maka diambil kesimpulan dari penelitian ini sebagai berikut: (1) Aplikasi file sharing berhasil mengimplementasikan algoritma RSA dalam pengamanan data, semua file yang diperlukan dapat diproses enkripsi dan dekripsi tanpa mengubah informasi yang terdapat dalam file tersebut; (2) Aplikasi file sharing mampu berbagi file terenkripsi maupun terdekripsi melalui dropbox; (3) Berdasarkan hasil pengujian, ukuran file yang semakin besar mempengaruhi lama waktu dalam proses enkripsi dan dekripsi; dan (4) Terbentuknya aplikasi file sharing dengan teknik pengamanan file dapat membantu proses berjalannya bisnis pada PT. Sumber Makmur Pangan Sejahtera dalam kegiatan file sharing yang dilakukan.

V. DAFTAR RUJUKAN

[1] W. P. Anang, *Implementasi Algoritma Kriptografi RSA Pada surat Elektronik (E-Mail)*. Jurnal Transient, Vol. 3 No. 4, hlm. 442-450. ISSN: 2302-9927,443. Semarang. 2014.

[2] A. Arief., et al. Implementasi Kriptografi Kunci Publik Dengan Algoritma RSA-CRT Pada Aplikasi Instant Messaging. Scientific Journal of Informatics, Vol.3, No. 1, hlm. 46-54. ISSN: 2407-7658. Semarang. 2016

[3] M. Ridwan., et al. Rancang Bangun E-Voting Dengan Menggunakan Keamanan Algoritma Rivest Shamir Adleman (RSA) Berbasis Web (Studi Kasus: Pemilihan Ketua BEM FMIPA). Jurnal Informatika Mulawarman, Vol. 11, No. 2, hlm. 22-28. ISSN: 1858-4853. Samarinda. 2014.

[4] A. S. D. Maulani, *Perancangan Sistem Penunjang Keputusan Biaya Kebutuhan Mahasiswa Dengan Waktu Tercepat Melalui Metode Backward Chain Dan Algoritma RSA*. Seminar Nasional Teknologi Informasi dan Multimedia. hlm 31-35. ISSN: 2302-3805. Yogyakarta. 2017.

[5] Linda & H. Agung. Aplikasi Laporan Keuangan Akuntansi Bulog Jakarta Menggunakan Algoritma MD5 dan RSA. Proceeding of Innovative and Creative Information Technology Conference, Vol. 1. hlm. 140-152. ISSN: 2548-1665. 2016.

[6] H. Agung., et al. Kriptografi Menggunakan Hybrid Cryptosystem dan Digital Signature. Jurnal Teknik Informatika dan Sistem Informasi, Vol.3, No.1, hlm. 34-45. ISSN: 2407-4322. 2016.

[7] Busran, et al. Rekayasa Perangkat Lunak Kriptografi Menggunakan Algoritma RSA Pada Sistem Keamanan File Berbasis Java. Jurnal TEKNOIF. Vol. 2. No. 1, hlm. 7-17. ISSN: 2338-2724. Padang. 2014.

[8] Sukamto, (2015) Modul Pembelajaran Perangkat Lunak, Bandung. 2015. hlm. 57.

[9] I. Kurniawan. Analisa Dan Perancangan Perangkat Lunak Keamanan Data Dengan Menggunakan Algoritma RSA. Tanjung Mulia. 2012. hlm. 69.

[10] F. Azmi. Analisis Keamanan Data Pada Block Chiper Algoritma Kriptografi RSA. CESS (Journal of Computer Engineering, System and Science). Vol. 2. No. 1, hlm. 27-29. ISSN: 2502-7131. Medan. 2017.