Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

Analisis Modifikasi Pembangkitan Kunci Rivest Code 2 Dengan Metode Belum-Micali Generator (BM-G) Untuk Meningkatkan Nilai Confusion Terhadap Record Data Login Pada Database

Chaiarun Nisa Lubis

Fakultas Ilmu Komputer dan Teknologi Informasi, Prodi Teknik Informatika, Universitas Budi Darma, Medan, Indonesia
Email Penulis Korespondensi: annisalubis2295@gmail.com
Submitted: **11/09/2021**; Accepted: **25/02/2022**; Published: **28/02/2022**

Abstrak−Record data login pada database adalah tabel yang menyimpan sejumlah data yang berkaitan dengan username dan password. Informasi yang terkadung dalam data username dan password tentu tidak bisa sembarangan dapat diakses oleh orang lain, karena data tersebut bersifat rahasia. Pembahasan yang akan dilakukan pada penelitian ini adalah melakukan kegiatan analisis terhadap pembangkitan kunci Rivest Code 2 yang kemudian kunci tersebut dimodifikasi dengan memadukan metode pembangkitan kunci Belum-Micali Generator (BM-G). Hasil dari proses modifikasi pembangkitan kunci Rivest Code 2, selanjutnya dilakukan pengujian terhadap algoritma Rivest Code 2 untuk kegiatan meningkatkan nilai confusion terhadap record data pada database.

Kata Kunci: Record Data; Rivest Code 2; Blum Michali Generator; Analisis

Abstract−The login data record in the database is a table that stores a number of data related to the username and password. The information contained in the username and password data certainly cannot be accessed arbitrarily by others, because the data is confidential. The discussion that will be carried out in this research is to conduct an analysis of the Rivest Code 2 key generation which is then modified by combining the method of generating the Not-Micali Generator (BM-G) key. The results of the modification process for Rivest Code 2 key generation, then testing the Rivest Code 2 algorithm for activities to increase the value of confusion for data records in the database.

Keywords: Record Data; Rivest Code 2; Blum Michali Generator; Analysis

1. PENDAHULUAN

Nilai *confusion* adalah salah satu nilai utama yang diharapkan dari proses penerapan algoritma atau metode pengamanan data digital. Sebab degan nilai confsuion maka orang yang tidak memiliki hak terhadap data tersebut tidak dapat mengakses informasi yang terkandung didalamnya. *Confusion* memiliki arti dalam bahasa indonesia adalah Kebingungan.

Record data *login* pada *database* adalah tabel yang menyimpan sejumlah data yang berkaitan dengan *username* dan *password*. Informasi yang terkadung dalam data *username* dan *password* tentu tidak bisa sembarangan dapat diakses oleh orang lain, karena data tersebut bersifat rahasia. Maka dibutuhkan sebuah teknik pengamanan yang dapat menghasilkan nilai *confusion* terhadap bentuk data *username* dan *password* dengan tujuan untuk menghindari pihak ketiga memahami isi dari pada data *username* dan *password*. Teknik pengamanan data yang menghasilkan nilai *confusion* salah satunya adalah teknik kriptografi. Kriptografi adalah ilmu dan seni tentang teknik pengamanan data dengan cara merubah bentuk data kebentuk objek lainnya [1]. Salah satu algoritma kriptografi yang umum digunakan adalah Algoritma Rivest Code 2.

RC2 adalah cipher blok, dan ukuran blok adalah 8 byte (64 bit). Ini berarti bahwa data input pertama-tama dibagi menjadi blok-blok 8 byte dan kemudian masing-masing diproses secara terpisah. Setiap blok data diperlakukan sebagai empat kata, setiap kata memiliki 16 bit (2 byte). Array empat kata disajikan sebagai R [0] R [1] R [2] R [3]. Enkripsi dan dekripsi menggunakan array ini sebagai input dan memodifikasi empat kata. Output dikembalikan dalam array yang sama. Pada penelitian terdahulu yang dilakukan Hendrasyah menuliskan Rivest Code 2 sangat rentan terhadap serangan jenis *know-plaintext attack* dan *ciphertext-only attack* [2].

Pembahasan yang akan dilakukan pada penelitian ini adalah melakukan kegiatan analisis terhadap pembangkitan kunci Rivest Code 2 yang kemudian kunci tersebut dimodifikasi dengan memadukan metode pembangkitan kunci Belum-Micali Generator (BM-G). Hasil dari proses modifikasi pembangkitan kunci Rivest Code 2, selanjutnya dilakukan pengujian terhadap algoritma Rivest Code 2 untuk kegiatan meningkatkan nilai confusion terhadap record data pada database.

2. METODOLOGI PENELITIAN

2.1 Kriptografi

Kriptografi pada awalnya dijabarkan sebagai ilmu yang mempelajari bagaimana menyembunyikan pesan. Namun pada pengertian modern *kriptografi* adalah ilmu yang bersandarkan pada teknik matematika untuk berurusan dengan keamanan informasi seperti kerahasiaan, keutuhan data dan otentikasi entitas. Jadi pengertian*kriptografi* modern adalah tidak saja berurusan hanya dengan menyembunyikan pesan, namun lebih pada sekumpulan teknik yang menyediakan keamanan informasi [1].

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

2.2. Algoritma RC2

RC2 (Rivest Cipher 2) merupakan cipher blok yang didesain oleh Ron Rivest pada tahun 1987. Pengembangan RC2 disponsori oleh Lotus yang sedang mencari suatu cipher tertentu setelah evaluasi oleh NSA untuk diekspor sebagai bagian dari software Lotus Notes. Setelah negosiasi lanjut, cipher ini disetujui dan diekspor pada tahun 1989. Seperti diketahui pembatasan kunci untuk ekspor kriptografi oleh pemerintah AS hanya sebesar 40 bit saja. Semua ini termuat dalam export regulation for cryptography. Detail dari algoritma ini dijaga kerahasiaannya dan sebagai property dari RSA Security. Tetapi pada tanggal 29 Januari 1996, source code atau kode sumber dari RC2 telah dipublikasikan ke 29 Internet oleh orang tanpa nama melalui forum Usenet yaitu Sci.crypt [2].

RC2 adalah cipher blok yang menggunakan 64 bit sebagai ukuran per blok-nya dengan kunci yang ukurannya bervariasi (0-1024 bit). Dengan mengubah-ubah ukuran kunci ini, performansi RC2 dapat menjadi dua atau tiga kali lebih baik dibanding DES (Data Encryption Standard), algoritma yang dikembangkan oleh NSA (National Security Agent) dan telah ditetapkan sebagai algoritma enkripsi standar oleh pemerintah AS pada tahun 1976-1997 (yang kemudian digantikan oleh AES, Advanced Encryption Standard). Awalnya, kunci masukan dari pengguna di-padding (jika kunci kurang dari 128 byte) menggunakan S-Box RC2. Kemudian hasilnya dikonversi menjadi 64 buah kunci dengan masing-masing berukuran 2 byte (16 bit). Di sisi lain, pesan dibagi menjadi blok-blok 64 bit. Masing-masing blok dibagi menjadi 4 word yang masing-masing berukuran 16 bit. Word ini kemudian dikomputasikan dengan kunci 16 bit di atas dengan rumus:

$$C = CLS_s(K \oplus P)$$

Keterangan:
- C = Ciphertext
- K = Kunci
- P = Plaintext
- O = Operasi logika XOR
- CLSS = circular left shift (pergeseran bit ke kiri) sebanyak s bit.

RC2 adalah 64-bit blok cipher dengan ukuran variabel kunci. 18 Its putaran diatur sebagai sumber-berat jaringan Feistel, dengan 16 putaran dari satu jenis (pencampuran) diselingi oleh dua putaran jenis lain (dihaluskan). Sebuah putaran pencampuran terdiri dari empat aplikasi transformasi MIX

2.3. Metode Belum-Micali Generator (BM-G)

Algoritme Blum-Micali adalah metode generator yang menarik dan berguna untuk menghasilkan bilangan acak semu yang aman secara kriptografi. Mengapa mereka aman secara kriptografis? Untuk mereproduksi urutan nomor, Anda harus memiliki nilai asli yang digunakan saat melakukan seeding pada algoritme. Nilai dalam urutan tidak dapat direproduksi (jika nilai benih cukup besar) tanpa mengakui bahwa kita memang telah memecahkan masalah log diskrit (DLP) [3].

$$\mathbf{X_{i+1} = G \wedge X_i \ \%\ P}$$

Menurut rumusnya, Anda harus menyediakan root primitif dan menggunakannya sebagai generator (G) di atas modulus prima (P). Meskipun relatif mudah untuk menemukan bilangan prima melalui pengujian Rabin-Miller, menemukan akar primitif dari kelompok utama itu mungkin sulit jika bukan hanya mustahil [3].

Jadi apa yang kita lakukan sekarang? Untuk menyiasati hal ini, kami secara strategis memutuskan untuk menggunakan jenis prime aman khusus. Bilangan prima aman adalah bilangan prima (P) yang juga bilangan prima di mana (P-1) / 2. Kita akan menggunakan bilangan prima aman khusus yang memiliki residu 3 atau 7 (modulo 8) yang memiliki sifat khusus bahwa akar primitif sama dengan +2 atau -2 pada kelompok modulus prima.

Dengan kata lain, kita perlu mencari bilangan bulat positif (P) yang memenuhi kondisi berikut [3]:

1. **P % 8 = 3 or 7**
2. **P is prime**
3. **(P - 1) / 2 is prime**

3. HASIL DAN PEMBAHASAN

Teknik pengamanan data yang menghasilkan nilai *confusion* salah satunya adalah teknik kriptografi. Pembahasan yang akan dilakukan pada penelitian ini adalah melakukan kegiatan analisis terhadap pembangkitan kunci Rivest Code 2 yang kemudian kunci tersebut dimodifikasi dengan memadukan metode pembangkitan kunci Belum-Micali Generator (BM-G). Hasil dari proses modifikasi pembangkitan kunci Rivest Code 2, selanjutnya dilakukan pengujian terhadap algoritma Rivest Code 2 untuk kegiatan meningkatkan nilai *confusion* terhadap *record* data pada database. Nilai *confusion* pada penelitian ini difokuskan pada perubahan karakter data login yang tersimpan dalam database.

3.1 Penerapan Algoritma RC2

Pembahasan pada sub judul ini menjelasakan cara menerapkan algoritma RC2 sekaligus cara modifikasi pembangkitan kunci enkripsi dengan menggunakan metode Blum Michali Generator.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

Berikut proses modifikasi dan penerapan algoritma RC2:
Isi *record login* yang akan di enkripsi : CHAIRUNI
Kata kunci yang digunakan : PUPUT
Setelah ditentukan sampel data yang akan di enkripsi selanjutnya membuat table array kunci, sebagai berikut:

Tabel 1. Index Array Kunci

Array	0	1	2	3	4
Kar.Kunci	P	U	P	U	T
Dec.Kunci	80	85	80	85	84

1. Pembentukan Tabel Vektor S, dengan formula S[i] = i
Tabel vektor berisi array 256 (0-255)
Berikut perhitungannya
i=0
S[0]=0…(nilai indeks ke 0)
i=1
S[1]=1…(nilai indeks ke 1)
i=2
S[2]=2…(nilai indeks ke 2), dst

Tabel 2. Tabel Vektor S

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255

2. Pembentukkan Tabel Vektor U, dengan formula
U[i] = Kunci [I mod jumlah_karakter_kunci]
Terdiri dari array 256 (0-255) yang berisi panjang kunci masukkan dari user,
i=0 i=1
U[0] = Kunci[0 mod 5] U[1] = Kunci[1 mod 5]
U[0] = Kunci[0] = 80 U[1] = Kunci[1] = 85…dst

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

Tabel 3. Tabel Vektor U

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
80	85	80	85	84	80	85	80	85	84	80	85	80	85	84	80
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
85	80	85	84	80	85	80	85	84	80	85	80	85	84	80	85
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
80	85	84	80	85	80	85	84	80	85	80	85	84	80	85	80
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
85	84	80	85	80	85	84	80	85	80	85	84	80	85	80	85
64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
84	80	85	80	85	84	80	85	80	85	84	80	85	80	85	84
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
80	85	80	85	84	80	85	80	85	84	80	85	80	85	84	80
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
85	80	85	84	80	85	80	85	84	80	85	80	85	84	80	85
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
80	85	84	80	85	80	85	84	80	85	80	85	84	80	85	80
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
85	84	80	85	80	85	84	80	85	80	85	84	80	85	80	85
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
84	80	85	80	85	84	80	85	80	85	84	80	85	80	85	84
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
80	85	80	85	84	80	85	80	85	84	80	85	80	85	84	80
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
85	80	85	84	80	85	80	85	84	80	85	80	85	84	80	85
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
80	85	84	80	85	80	85	84	80	85	80	85	84	80	85	80
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
85	84	80	85	80	85	84	80	85	80	85	84	80	85	80	85
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
84	80	85	80	85	84	80	85	80	85	84	80	85	80	85	84
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
80	85	80	85	84	80	85	80	85	84	80	85	80	85	84	80

3. Melakukan permutasi terhadap nilai Tabel Vektor S, berdasarakan pseudecode berikut

```
j=0
for i = 0 to 255
j = ( j + S[i] + U[i]) mod 256
swap (S[i], S[j])
```

Berikut ini sample perhitungannya :

j=0 i=0
j=(0 + S[0] + U[0]) mod 256
j=(0 + 0 + 80) mod 256
j=80 mod 256
j=80

j=80 i=1
j=(80 + S[1] + U[1]) mod 256
j=(80 + 1 + 85) mod 256
j= 166 mod 256
j=166…dst

Tabel 4. Tabel Permutasi Vektor S

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
80	166	248	80	168	253	88	175	12	105	195	35	127	225	67	162
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
7	104	207	54	154	4	106	214	66	171	26	133	246	103	213	73
32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47
185	47	165	24	145	6	129	252	116	242	108	236	108	233	108	235
48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63
112	245	119	255	131	13	151	30	171	52	195	82	222	112	254	146

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
38	183	78	225	122	19	169	69	221	123	25	180	85	242	149	56
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95
216	126	32	200	112	21	192	103	20	193	107	27	199	121	43	218
96	97	98	99	100	101	102	103	104	105	106	107	108	109	110	111
143	64	247	174	98	28	210	142	74	3	194	125	62	255	189	129
112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127
65	7	205	144	89	30	233	180	124	74	20	228	180	129	84	35
128	129	130	131	132	133	134	135	136	137	138	139	140	141	142	143
248	205	159	119	75	37	255	214	179	140	107	74	38	8	230	202
144	145	146	147	148	149	150	151	152	153	154	155	156	157	158	159
174	143	118	89	66	43	17	253	229	211	193	172	157	138	125	112
160	161	162	163	164	165	166	167	168	169	170	171	172	173	174	175
96	86	72	64	56	45	40	31	28	25	19	19	15	17	19	18
176	177	178	179	180	181	182	183	184	185	186	187	188	189	190	191
23	24	31	38	42	52	58	70	82	91	106	117	134	151	165	185
192	193	194	195	196	197	198	199	200	201	202	203	204	205	206	207
201	223	245	8	33	54	81	108	132	162	188	220	252	25	60	91
208	209	210	211	212	213	214	215	216	217	218	219	220	221	222	223
128	165	199	239	19	61	103	142	187	228	19	66	110	160	206	2
224	225	226	227	228	229	230	231	232	233	234	235	236	237	238	239
54	103	158	209	10	67	121	181	237	43	105	164	229	34	101	168
240	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
232	46	112	184	0	69	144	215	36	113	187	11	87	169	251	74

4. Membangkitkan Aliran Kunci

pembangkitan aliran kunci dilakukan berdasarkan pseudecode berikut :

```
i=0                    j=0
 for idx=0 to panjang_plainteks-1 do
 i=(i+1) mod 256
 j=(j+S[i]) mod 256
 swap (S[i], S[j])
 K=(S[i]+S[j]) mod 256
```

Berikut Proses Pembangkitan Aliran Kunci

```
i=0                    j=0
for idx=0 to panjang_plainteks-1 do
for idx=0 to 8-1 = 7 do
i=(i+1) mod 256
i=(0+1) mod 256
i= 1 mod 256
i=1
j=(j+S[i]) mod 256
j=(0+S[1]) mod 256
j=(0+166) mod 256
j=166
swap (S[i], S[j])
swap (S[1], S[166])
swap(166,40)
S[1] = 40, S[166]=166
K=(S[i]+S[j]) mod 256
K=(S[1]+S[166]) mod 256
K= (40+166) mod 256
K=206          Biner =    11001110
```

```
i=1            j=166
i=(i+1) mod 256
i=(1+1) mod 256
i= 2 mod 256
i=2
j=(j+S[i]) mod 256
j=(166 +S[2]) mod 256
j=(166+248) mod 256
j= 414 mod 256
j= 158
swap (S[i], S[j])
swap (S[2], S[158])
swap(248,125)
S[2] = 125, S[158]=248
K=(S[i]+S[j]) mod 256
K=(S[2]+S[158]) mod 256
K= (125+248) mod 256
K=373 mod 256
K = 117        Biner =       01110101
```

```
i=2            j=158
i= (i+1) mod 256
i= (2+1) mod 256
i= 3
j= (j+ S[i]) mod 256
j= (158 + S[3]) mod 256
j= (158 + 80) mod 256
j= 238 mod 256
j= 238
```

```
i=3            j=238
i=(i+1) mod 256
i=(3+1) mod 256
i=4
j=(j+S[i]) mod 256
J=(238+S[4]) mod 256
j=(238+168) mod 256
j=406 mod 256
j=150
```

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

swap (S[i], S[j])
swap (S[3], S[238])
swap (80,101)
S[3]= 101, S[238]= 80
K= (S[i] + S[j]) mod 256
K= (S[3] + S[238]) mod 256
K= (101 + 80) mod 256
K= 181 mod 256
K= 181 Biner = 10110101

i=4 j=150
i=(i+1) mod 256
i=(4+1) mod 256
i=5
j=(j+S[i]) mod 256
j=(150 + S[5]) mod 256
j=(150 + 253) mod 256
j=403 mod 256
j=147
swap (S[i], S[j])
swap (S[5], S[147])
swap (253, 89)
S[5]= 89, S[147]= 253
K= (S[i] + S[j]) mod 256
K= (S[5] + S[147]) mod 256
K= (89 + 253) mod 256
K= 342 mod 256
K= 86 Biner= 01010110

i=6 j=235
i=(i+1) mod 256
i=(6+1) mod 256
i=7
j=(j+S[i]) mod 256
j=(235+S[7]) mod 256
j=(235 + 175) mod 256
j= 410 mod 256
j= 154
swap (S[i], S[j])
swap (S[7], S[154])
swap (175, 193)
S[7]= 193, S[154]= 175
K= (S[i] + S[j]) mod 256
K= (S[7] + S[154]) mod 256
K= (193 + 175) mod 256
K= 368 mod 256
K= 112 Biner= 01110000

swap (S[i], S[j])
swap (S[4], S[150])
swap (168, 17)
S[4] = 17, S[150]= 168
K= (S[i] + S[j]) mod 256
K= (S[4] + S[150]) mod 256
K= (17 + 168) mod 256
K= 185 mod 256
K= 185 Biner= 10111001

i=5 j=147
i=(i+1) mod 256
i=(5+1) mod 256
i=6
j=(j+S[i]) mod 256
j=(147+S[6]) mod 256
j=(147 + 88) mod 256
j= 235 mod 256
j= 235
swap (S[i], S[j])
swap (S[6], S[235])
swap (88, 164)
S[6]= 164, S[235]= 88
K= (S[i] + S[j]) mod 256
K= (S[6] + S[235]) mod 256
K= (164 + 88) mod 256
K= 252 mod 256
K= 252 Biner= 11111100

i=7 j=154
i=(i+1) mod 256
i=(7+1) mod 256
i=8
j=(j+S[i]) mod 256
j=(154 + S[8]) mod 256
j=(154 + 12) mod 256
j= 166 mod 256
j= 166
swap (S[i], S[j])
swap (S[8], S[166])
swap (12, 40)
S[8]= 40, S[166]= 12
K= (S[i] + S[j]) mod 256
K= (S[8] + S[166]) mod 256
K= (40 + 12) mod 256
K= 52 mod 256
K= 52 Biner= 00110100

Tabel 5. Nilai Kunci

Kunci	Desimal	Biner
K1	206	11001110
K2	117	01110101
K3	181	10110101
K4	185	10111001
K5	86	01010110
K6	252	11111100
K7	112	01110000
K8	52	00110100

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

3.2. Penerapan Metode BM-G

Setelah proses pembangkitan kunci RC2 dilakukan, selanjutnya proses pembangkitan nilai dilakukan terhadap masing-masing nilai K dengan menggunakan metode Blum Michali Generator, dimana nilai modulus adalah 256, dan nilai bilangan awal yang diacak sama dengan K. berikut proses perhitungan.

1. Generate BM-G terhadap kunci 206
   $X_1 = K^2$ Mod 256
   $X_1 = 206^2$ mod 256
   $X_1 = 42436$ mod 256 = 196
2. Generate BM-G terhadap kunci 117
   $X_2 = K^2$ Mod 256
   $X_2 = 117^2$ mod 256
   $X_2 = 13689$ mod 256 = 121
3. Generate BM-G terhadap kunci 181
   $X_3 = K^2$ Mod 256
   $X_3 = 181^2$ mod 256
   $X_3 = 32761$ mod 256 = 249
4. Generate BM-G terhadap kunci 185
   $X_4 = K^2$ Mod 256
   $X_4 = 185^2$ mod 256
   $X_4 = 34225$ mod 256 = 177
5. Generate BM-G terhadap kunci 86
   $X_5 = K^2$ Mod 256
   $X_5 = 86^2$ mod 256
   $X_5 = 7396$ mod 256 = 228
6. Generate BM-G terhadap kunci 252
   $X_6 = K^2$ Mod 256
   $X_6 = 252^2$ mod 256
   $X_6 = 63504$ mod 256 = 16
7. Generate BM-G terhadap kunci 112
   $X_7 = K^2$ Mod 256
   $X_7 = 112^2$ mod 256
   $X_7 = 12544$ mod 256 = 0
8. Generate BM-G terhadap kunci 52
   $X_8 = K^2$ Mod 256
   $X_8 = 52^2$ mod 256
   $X_8 = 2704$ mod 256 = 144

Berikut adalah proses Enkripsi Algoritma RC2 dengan cara dilakukan proses XOR antara biner plaintext dengan biner kunci

1. Konversi plainteks dan Kunci Hasil pembangkitan kunci kedalam biner

Tabel 6. Konversi Bilangan Plainteks dan Kunci

Plainteks	Desimal	Biner	Kunci	Desimal	Biner
C	67	01000011	K1	196	11000100
H	72	01001000	K2	121	01111001
A	65	01000001	K3	249	11111001
I	73	01001001	K4	177	10110001
R	82	01010010	K5	228	11100100
U	85	01010101	K6	16	00010000
N	78	01001110	K7	0	00000000
I	73	01001001	K8	144	10010000

2. Lakukan operasi XOR antara Biner Kunci hasil pembangkitan kunci dengan Biner Plainteks

C	01000011	
K1	11000100	XOR
C1	10000111	
H	01001000	
K2	01111101	XOR
C2	00110101	
A	01000001	

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

K3	11111001	XOR
C3	10111000	
I	01001001	
K4	10111001	XOR
C4	11110000	
R	01010010	
K5	11100100	XOR
C5	10110110	
U	01010101	
K6	00010000	XOR
C6	01000101	
N	01001110	
K7	00000000	XOR
C7	01001110	
I	01001001	
K8	10010000	XOR
C8	11011001	

Setelah proses XOR bilangan biner plainteks terhadap biner kunci selesai dilakukan, berikut tampilan nilai ciphertext.

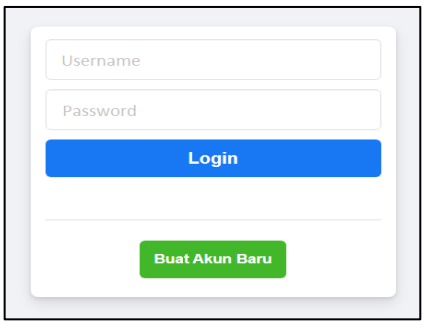
Tabel 7. Tabel Ciphertext

Biner Hasil XOR	Desimal	Karakter
C1= 10000111	135	ç
C2 = 00110101	53	5
C3 = 10111000	184	©
C4 = 11110000	240	≡
C5 = 10110110	182	Â
C6 = 01000101	69	E
C7 = 01001110	78	N
C8 = 11011001	217	┘

Dari proses perhitungan diatas maka disimpulkan proses modifikasi kunci terhadap algoritma RC2 dapat dilakukan dan dapat menghasilkan nilai ciphertext yang dapat memberikan *effect confusion* terhadap karakter *record login* yang telah di enkripsi.

3.3 Implementasi

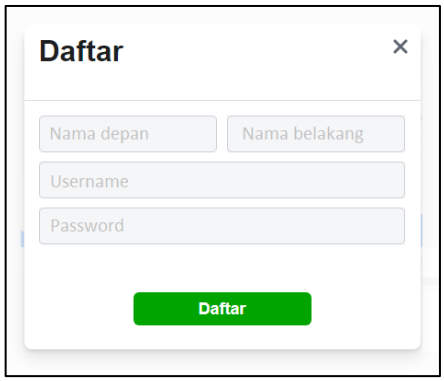
Berikut tampilan tiga form yang dibangun penulis untuk menguji algoritma Rivest Code 2 yang telah di modifikasi untuk meningkatkan nilai *confusion* terhadap record data login pada database.



Gambar 1. Tampilan Form Login

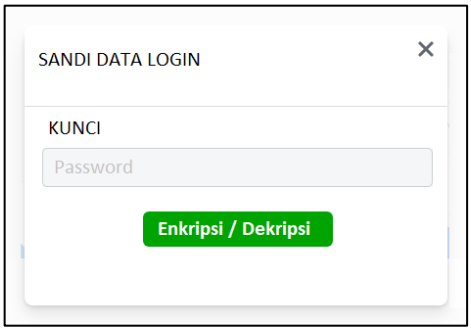
Tampilan login ini berfungsi untuk memasukan data *username* dan *password* di mana data *username* dan *password* sebelumnya telah di simpan dalam table login yang ada dalam database.

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953



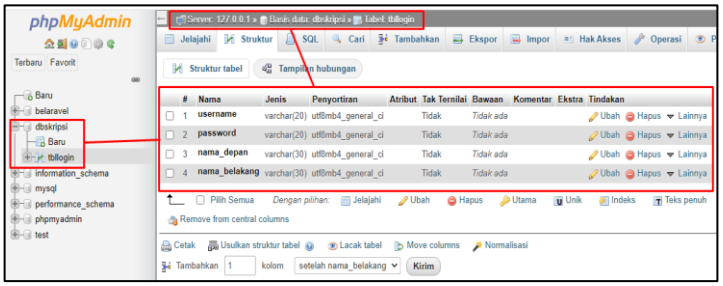
Gambar 2. Tampilan Form Create Account Login

Tampilan form ini berfungsi untuk memasukan data user dimana data yang dimasukan terdiri dari nama depan, nama belakang, *username* dan *password*, kemudian data tersebut tersimpan dalam table login yang ada dalam database.



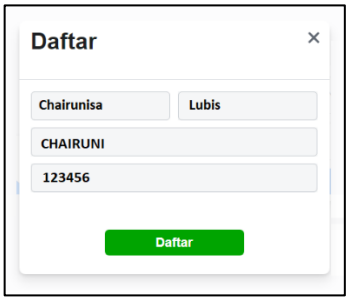
Gambar 3. Form Input Key Enkripsi/Dekripsi

Form input kunci enkripsi atau dekripsi berfungsi untuk memasukan kata kunci yang digunakan untuk menyandikan karakter pada saat melakukan proses enkripsi dan dekripsi data *username* dan *password* yang ada di dalam database.



Gambar 4. Tampilan table login dalam database

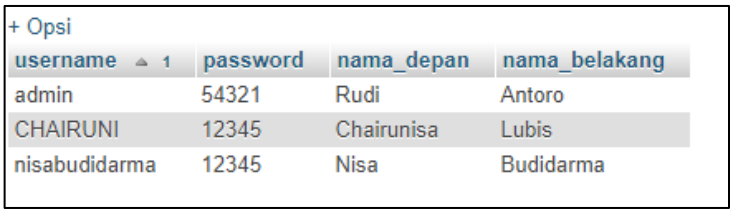
Tampilan phpmyadmin ini berfungsi untuk melihat struktur table dan data yang ada pada masing-masing field pada database. Selain itu juga fungsi dari phpmyadmin ini penulis menggunakan nya untuk melihat perubahan karkater hasil dari proses enkripsi dan dekripsi Algoritama RC2. Pada sub bab berikut ini penulis menjelaskan bentuk tampilan luaran pada saat melakukan pengujian terhadap penerapan algoritma RC2 yang pembangkitan kunci nya telah di modifikasi dengan menggunakan michali blum generator.



Gambar 5. Form Input Data User Baru

Journal of Computer System and Informatics (JoSYC)
ISSN 2714-8912 (media online), ISSN 2714-7150 (media cetak)
Volume 3, No. 2, Februari 2022, Page 73−82
https://ejurnal.seminar-id.com/index.php/josyc
DOI 10.47065/josyc.v3i2.953

Berikut tampilan form penginputan data akun user yang baru kemudian di klik tombol daftar untuk menyimpan data yang dimasukan pada masing-masing *textbox* pada form. Berikut tampilan data dalam table login pada database sebelum di enkripsi

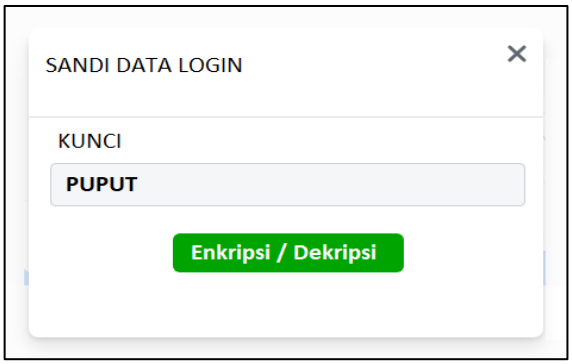


+ Opsi

username	password	nama_depan	nama_belakang
admin	54321	Rudi	Antoro
CHAIRUNI	12345	Chairunisa	Lubis
nisabudidarma	12345	Nisa	Budidarma

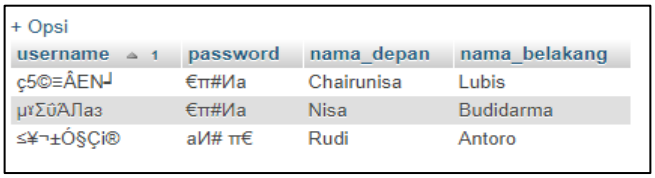
Gambar 6. Bentuk data pada tabel sebelum di enkripsi

Selanjutnya setelah data *username* dan *password* masuk dalam tabel login, langkah selanjutnya adalah mengakses form *input key* Enkripsi/Dekripsi untuk memasukan nilai kunci proses enkripsi atau dekripsi.



Gambar 7. Form Input Data Kunci Proses Enkripsi

Setelah data kunci proses enkripsi dimasukan, kemudian tekan tombol Enkripsi/Dekripsi untuk melakukan proses penyandian terhadap data yang ada pada *record* tabel *login*. Sehingga tampilan teks yang ada pada tabel login berubah ke dalam bentuk karakter lain seperti pada gambar berikut ini



+ Opsi

username	password	nama_depan	nama_belakang
ç5©≡ÂEN┘	€π#Иa	Chairunisa	Lubis
µ¤ΣûÄЛaз	€π#Иa	Nisa	Budidarma
≤¥¬±Ó§Çi®	aИ# π€	Rudi	Antoro

Gambar 8. Bentuk Tampilan Karakter Setelah Proses Enkripsi

4. KESIMPULAN

Setelah penelitian dilakukan dan hasil pengujian diperoleh, maka penulis dapat menyimpulkan garis besar dari keseluruhan rangkuman Algoritma *Rivest Code 2* yang diterapkan pada proses pengamanan record login bisa mengamankan data yang ada dalam *record* tersebut. Pembangkit kunci Belum Micali Generator (BM-G) mempunyai tahap pembangkitan kunci dan proses yang sangat cepat karena menggunakan perhitungan yang sederhana. Berdasarkan pengujian yang dilakukan, maka ciphertext yang dihasilkan menggunakan kunci yang dibangkitkan menggunakan metode belum micali generator menghasilkan text yang memberikan nilai confusion.

REFERENCES

[1] N. P. Smart, *Cryptography: An Introduction*. 2003.
[2] R. Munir, *Kriptografi*. Bandung: Informatika, 2006.
[3] T. S. Waruwu and K. Telaumbanua, "Kombinasi Algoritma OTP Cipher dan Algoritma BBS dalam Pengamanan File," *JSM STMIK Mikroskil*, vol. 17, no. 1, pp. 119–126, 2016.