

## IMPLEMENTASI WEB SERVICE RESTFUL DAN ALGORITMA RC4 DALAM PROSES VALIDASI DATA MAHASISWA DAN DOSEN PADA SAAT PROSES LOGIN KE APLIKASI BLUCAMPUS

Agung Yuliyanto<sup>1)</sup>, Muhammad Ainur Rony<sup>2)</sup>

Program Studi Teknik Informatika, Fakultas Teknologi Informasi, Universitas Budi Luhur

Jl. Raya Ciledug, Petukangan Utara, Kebayoran Lama, Jakarta Selatan 12260

E-mail : [agung.96tm@gmail.com](mailto:agung.96tm@gmail.com)<sup>1)</sup>, [ainur.rony@gmail.com](mailto:ainur.rony@gmail.com)<sup>2)</sup>

### Abstrak

Pada proses login ke aplikasi BluCampus pengguna diharuskan untuk melakukan proses registrasi terlebih dahulu sebelum dapat login ke dalam aplikasi, tentu hal ini akan merepotkan bilamana seseorang yang login tersebut adalah Mahasiswa ataupun Dosen yang berasal dari Universitas Budi Luhur, karena BluCampus adalah aplikasi yang dikhususkan untuk Mahasiswa dan Dosen Universitas Budi Luhur. Oleh karena itu penggunaan Web Service RestFul sebagai media untuk memvalidasi Mahasiswa dan Dosen digunakan dalam proses pembuatan sistem login, sehingga memudahkan Mahasiswa dan Dosen dalam upaya login ke dalam aplikasi tanpa perlu melakukan registrasi lagi. Dalam prosesnya, data Mahasiswa dan Dosen yang sudah ada seperti NIM (Mahasiswa), NIK (Dosen) serta password digunakan, untuk selanjutnya dimanfaatkan BluCampus untuk diperiksa kebenaran data tersebut dengan cara mengirimkan data tersebut ke server Universitas Budi Luhur dengan menggunakan Web Service RestFul sebagai media prantaranya serta menggunakan algoritma RC4 saat proses pengiriman tersebut agar terjaga keamanan data yang akan diperiksa. Jika data tersebut benar dan ada di server Universitas Budi Luhur, maka pengguna dibolehkan login ke dalam aplikasi dan memanfaatkan fitur yang ada didalamnya. Hasil penelitian diharapkan dapat membantu setiap orang yang ingin menggunakan Web Service RestFul sebagai media validasi serta algoritma RC4 untuk pengamanan data sehingga mengurangi proses pencurian serta penggunaan data untuk tujuan yang tidak baik.

**Kata kunci:** Web Service, RestFul, RC4, BluCampus.

### 1. PENDAHULUAN

BluCampus adalah salah satu aplikasi yang bertujuan untuk digunakan oleh semua lapisan yang ada di Universitas Budi Luhur, baik untuk Dosen dan Mahasiswa serta orang lain di luar dari Universitas Budi Luhur itupun sendiri (*guest*). BluCampus memiliki beberapa fitur-fitur menarik, dimana tiap-tiap fitur tersebut bertujuan untuk menunjang kebutuhan rutinitas harian Universitas Budi Luhur, seperti BluFeed yang berfungsi untuk memberikan informasi terbaru yang ada di kampus, BluCare yang berfungsi untuk memberikan laporan tentang penyalahgunaan fasilitas yang ada di kampus, dan sebagainya. Dimana data-data dari tiap fitur tersebut disimpan ke dalam salah satu server pusat yang dimiliki oleh BluCampus.

Sebelum seseorang dapat menggunakan aplikasi BluCampus beserta fitur yang ada di dalamnya, maka pengguna tersebut diharuskan untuk *login* terlebih dahulu. Saat proses *login* inilah yang menimbulkan permasalahan, yaitu bilamana seseorang yang *login* adalah seseorang yang berasal dari ruang lingkup Universitas Budi Luhur seperti Mahasiswa dan Dosen, apakah Mahasiswa dan Dosen harus melakukan registrasi layaknya pengguna umum (*guest*), tentu hal ini sangatlah merepotkan. Oleh karena itu, BluCampus berinisiatif membuat sebuah sistem layaknya SSO (*Single Sign On*) yaitu proses login dengan memanfaatkan data yang sudah ada seperti NIM (Mahasiswa) dan NIK

(Dosen) beserta passwordnya untuk dapat *login* ke aplikasi BluCampus.

Karena data tersebut berada di *server* berbeda (Universitas Budi Luhur) sedangkan BluCampus sendiri memiliki *server*-nya sendiri, maka diperlukannya sebuah teknologi untuk menjembatani (penghubung) antara *server* BluCampus dengan *server* Universitas Budi Luhur yaitu teknologi *Web Service RestFul*. *RestFul / REST (Representational State Transfer)* adalah suatu arsitektur metode komunikasi yang menggunakan protokol *HTTP* untuk pertukaran data dan metode ini sering diterapkan untuk menjembatani antara *client* dan *server* untuk dapat saling terhubung. Dimana tujuannya adalah untuk menjadikan sistem yang memiliki performa yang baik, cepat dan mudah untuk di kembangkan (*scale*) terutama dalam pertukaran dan komunikasi data.

Dengan memanfaatkan teknologi *Web Service RestFul*, BluCampus akan melakukan validasi data masukan user yang berasal dari *form input login* yang telah disediakan khusus untuk seseorang dari ruang lingkup Universitas Budi Luhur (Mahasiswa dan Dosen), dimana data yang akan dilakukan validasi adalah data pengenalan seperti NIM dan NIK beserta password yang berasal dari Universitas Budi Luhur. Selanjutnya data tersebut akan dikirim oleh BluCampus ke *server* Universitas Budi Luhur untuk diperiksa apakah data identitas itu *valid* (ada) atau tidak *valid*, dengan begitu BluCampus akan tahu apakah memang pengguna tersebut adalah seseorang

dari ruang lingkup Universitas Budi Luhur atau tidak. Tentu saja dengan memanfaatkan data identitas pengenal tersebut, *server* Universitas Budi Luhur dengan mudah memeriksanya, karena data tersebut dibuat oleh Universitas Budi Luhur. Jika data identitas itu benar, maka *server* Universitas Budi Luhur akan mengirim data identitas pengguna tersebut (*response*), seperti nama, email, dan sebagainya ke *server* BluCampus, dimana data tersebut dimanfaatkan oleh BluCampus untuk validasi pengguna bersangkutan dan mengizinkan pengguna tersebut untuk masuk serta memanfaatkan fitur yang ada di dalam aplikasi BluCampus.

Dalam proses pengirim data identitas dari satu *server* ke *server* lain dibutuhkan juga proses pengamanan data, karena data yang terlibat di dalamnya amatlah penting, yaitu data yang berasal dari *server* Universitas Budi Luhur. Oleh karena itu, untuk menghindari proses pencurian data saat proses pertukaran / pengiriman data tersebut serta menjamin bahwa seseorang yang memasukkan data dari *server* BluCampus memang benar pemilik asli dari data-data yang ada (*valid*) maka diperlukan penerapan metode kriptografi. Salah satu metode untuk proses pengamanan data tersebut adalah kriptografi *RC4*, dengan begitu data yang ada dalam proses pertukarannya melalui teknologi *Web Service RestFul* akan lebih terjamin keamanannya, karena data yang melalui proses kriptografi (*enkripsi*) akan membentuk karakter-karakter acak (*chiphertext*) yang dijamin hanya dapat dikembalikan ke data semula (*plaintext*) bila melalui proses kriptografi (*dekripsi*) lagi.

## 2. METODE PENELITIAN

### 2.1. Metode Penelitian

Berikut ini adalah rincian bagaimana proses dalam pembuatan aplikasi login untuk memudahkan Mahasiswa dan Dosen dalam upaya masuk ke dalam aplikasi BluCampus :

- Pengumpulan Data  
Mengumpulkan kebutuhan dari keseluruhan elemen sistem yang akan diaplikasikan ke dalam bentuk software serta pemanfaatannya untuk proses pengiriman data menggunakan *Web Service RestFul* serta menggunakan algoritma *RC4* untuk pengaman data pada saat proses pengirimannya menggunakan *Web Service RestFul*.
- Menganalisa Kebutuhan Aplikasi  
Setelah mendapatkan kebutuhan penunjang dalam pembuatan aplikasi, maka selanjutnya adalah dilakukan analisa serta mempelajari fungsi apa saja yang akan dibutuhkan untuk di gunakan dalam aplikasi.
- Desain atau Perancangan Aplikasi  
Merancang tampilan dalam pembuatan sebuah aplikasi adalah sebuah tindakan yang diperlukan untuk mendapatkan sebuah gambaran

bagaimana sebuah aplikasi yang dibangun sesuai dengan kebutuhan sehingga memudahkan dalam melakukan proses pengkodean.

- Pengkodean  
Pengkodean dilakukan agar sebuah aplikasi dapat di rancang dengan baik menggunakan bahasa pemrograman PHP dan Javascript serta *tools* seperti Sublime Text 3 sebagai *text editor* dan XAMPP (Apache) sebagai *localhost server* dalam pembuatan aplikasi.
- Implementasi  
Rancangan aplikasi yang telah dibuat selanjutnya diimplementasikan sesuai dengan analisa masalah yang ada.
- Pengujian  
Pengujian dilakukan setelah aplikasi selesai dibuat dengan tujuan menemukan sebuah kesalahan yang kemudian akan segera di perbaiki hingga tidak ditemukan kesalahan lagi sehingga aplikasi dapat berjalan dengan lancar dan dapat digunakan sesuai dengan keinginan.

Salah satu metode yang dapat digunakan dalam pembuatan aplikasi web service selain SOAP adalah REST (Representational State Transfer). RestFul atau bisa di sebut REST menerapkan sebuah konsep dengan memanfaatkan perpindahan antar state. State disini dapat digambarkan apabila browser melakukan permintaan suatu web, maka server akan melakukan pengiriman state halaman web yang sekarang ke browser [1]. Ide awal dari metode REST adalah menggunakan mekanisme HTTP untuk menghubungkan aplikasi dibanding menggunakan mekanisme kompleks, seperti CORBA, RPC dan SOAP [2]. REST lebih befokus pada interaksi sumber daya dan perubahan state, bukan terfokus pada mengirim dan menerima pesan seperti SOAP [2].

REST didaskan pada operasi standar yang bersifat universal dan dapat digunakan untuk berbagai macam pengimanan dan sistem pencarian dana. Operasi standar ini biasa disebut dengan CRUD (Create, Read, Update, Delete) [3]

Tabel 1. Inti Metode HTTP CRUD

CRUD	REST	Penjelasan
CREATE	POST / PUT	Inisialisasi states pada resource yang telah diberi URI.
READ	GET	Mengambil state saat ini dari resource
UPDATE	PUT	Memodifikasi state dari resource
DELETE	DELETE	Menghapus resource

Selain menggunakan *Web Service RestFul* untuk media pengiriman (penjembatani) data, digunakan juga algoritma *RC4* untuk mengamankan data yang akan dikirim dari *server* BluCampus ke *server* Universitas Budi Luhur (*request*) begitu pula sebaliknya (*response*). Tujuan pengamanan ini

adalah untuk menghindari pencurian data dari orang yang tidak baik, sehingga data terjamin aman sampai tujuan.

Algoritma kriptografi Rivest Code 4 (RC4) merupakan algoritma kunci simetris yang dibuat oleh RSADSI (RSA Data Security Inc). Algoritma ini ditemukan oleh Ronald Rivest pada tahun 1987 dan menjadi simbol keamanan RSA (singkatan dari nama penemunya yaitu : Rivest, Shamir, dan Adleman). Algoritma RC4 menggunakan dua buah S-Box (Substitution Box) yaitu sebuah array dengan panjang 256 yang berisi permutasi dari bilangan 0 sampai 255, dan S-Box kedua yang berisi permutasi juga merupakan fungsi dari kunci dengan panjang variabel.

Untuk lebih menjelaskan algoritma RC4, maka berikut langkah-langkahnya [4] :

- a. Inisialisasi array S-box pertama, S[0], S[1], S[2],S[3] ....., dan seterusnya hingga mencapai S[255], diisi dengan nilai 0 sampai 255 sehingga array S-box array S yang terbentuk adalah S[0] = 0, S[1] = 1, ..., S[255] = 255. Berikut proses inisialisasi S-box (array S):

```

1. For r = 0 to 255 then
2.   S[r] = r
3. End For
    
```

- b. Inisialisasi array kunci (S-box lain), misalnya array kunci K dengan panjang 256. Jika panjang kunci kurang dari 256 (< 256), dilakukan proses padding, yaitu menambahkan byte sehingga panjang kunci menjadi 256 byte. Misalnya, K = abcd (hanya 4 byte atau huruf), lakukan padding dengan menambahkan byte (huruf) semu, misal menjadi K = abcdabcdabcd.... sampai panjang K mencapai 256 byte sehingga S-box array kunci K berbentuk K[0], K[1], K[2], ....., K[255]. Berikut proses inisialisasi S-box (array K) :

```

1. Array Kunci // panjang kunci "length"
2. For i = 0 to 255 then
3.   K[i] = Kunci [i mod length]
4. End For
    
```

- c. Lakukan permutasi terhadap nilai di dalam array S dengan cara menukarkan isi array S[i] dengan S[j] (swap). Berikut prosesnya pertukaran nilai yang ada di dalam array S tersebut :

```

1. j = 0
2. For i = 0 to 255 then
3.   j = (j + S[i] + K[i]) mod 256
4.   swap(isi S[i] dan isi S[j])
    
```

- d. Bangkitkan *keystream* dan lakukan enkripsi. Berikut bagaimana proses untuk membangkitkan kunci enkripsi tersebut :

```

1. i = j = 0
2. i = (i + 1) mod 256
3. j = (j + S[i]) mod 256
4. isi S[i] dan S[j] ditukar
5. t = S[i]
    
```

- e. Proses Keystream K kemudian digunakan untuk mengenkripsikan plaintext ke-idx sehingga didapatkan ciphertext, sedangkan untuk mendapatkan plaintext, XOR-kan ciphertext dengan kunci yang sama dengan proses enkripsi.

**2.2. Endpoint Web Service Restful Program**

Endpoint adalah sebuah titik akhir dimana data yang akan dikirim antar *client* ke *server* maupun *server* ke *server* digunakan untuk proses ke tahap selanjutnya, baik untuk diolah dan dimasukkan ke dalam *database* maupun untuk mendapatkan data sesuai keinginan.

Berikut adalah *Endpoint* yang digunakan untuk pengguna yang ingin *login* ke dalam aplikasi BluCampus :

Tabel 2. Endpoint Server BluCampus

Endpoint	Metode	Params	Response
api/auth/signin/mahasiswa (Login Mahasiswa)	POST	nim, password	status,message,token,user(id,username,nama,email,tanggal_lahir,jenis_kelamin,alamat,foto,telepon,ni_id_status,id_otorisasi,created_at,updated_at)
api/auth/signin/dosen (Login Dosen)	POST	nik, password	status,message,token,user(id,username,nama,email,tanggal_lahir,jenis_kelamin,alamat,foto,telepon,ni_id_status,id_otorisasi,created_at,updated_at)
api/auth/signin (Login Guest)	POST	username, password	status,message,token,user(id,username,nama,email,tanggal_lahir,jenis_kelamin,alamat,foto,telepon,ni_id_status,id_otorisasi,created_at,updated_at)

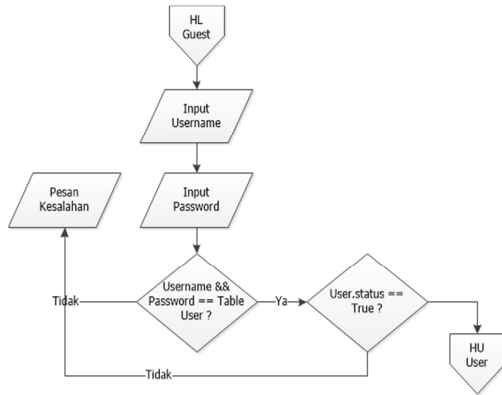
Untuk *Endpoint server* Universitas Budi Luhur yang digunakan oleh BluCampus untuk memvalidasi data Mahasiswa dan Dosen adalah sebagai berikut :

Tabel 3. Endpoint Server Universitas Budi Luhur

Endpoint	Metode	Params	Response
api/signin/mahasiswa (Validasi Mahasiswa)	POST	nim, password	status,message,mahasiswa(nama,email,tanggal_lahir,jenis_kelamin,nomor_telepon)
api/signin/dosen (Validasi Dosen)	POST	nik, password	status,message,dosen(nama,email,tanggal_lahir,jenis_kelamin,nomor_telepon)

**2.3. Flowchart dan Algoritma Program**

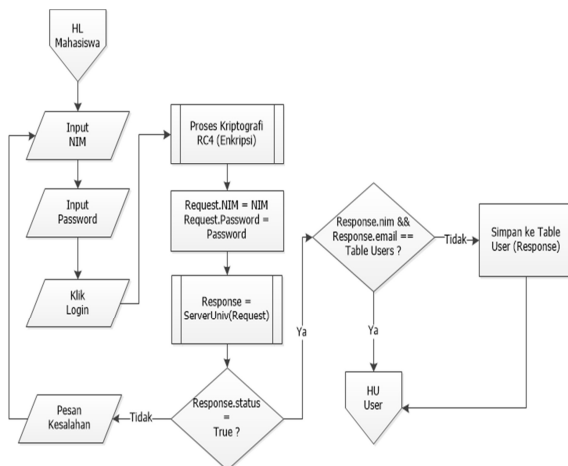
Berikut adalah *flowchart* dan algoritma bagaimana seseorang dapat *login* ke dalam aplikasi BluCampus. Dan difokuskan pada tahap *login* saja ke dalam aplikasi, baik untuk *guest*, Mahasiswa, dan Dosen :



Gambar 1. Flowchart Proses Login Guest

**Algoritma proses Login Guest :**

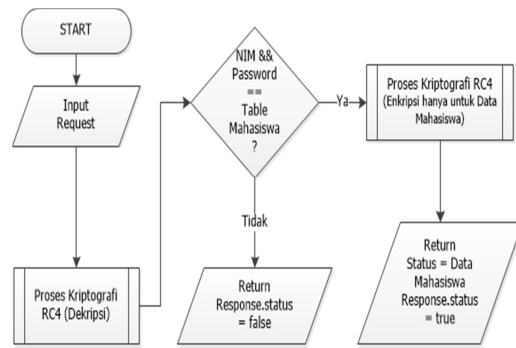
1. Tampil Halaman Login Guest
2. Input Username
3. Input Password
4. IF Login then
5. IF Username & Password == Table User then
6. IF User.status == true then
7. Tampil Halaman Utama User
8. ELSE
9. Tampil Error “belum melakukan validasi”
10. ELSE IF
11. Tampil Error “user tidak ditemukan”
12. End IF



Gambar 2. Flowchart Proses Login Mahasiswa

**Algoritma proses Login Mahasiswa :**

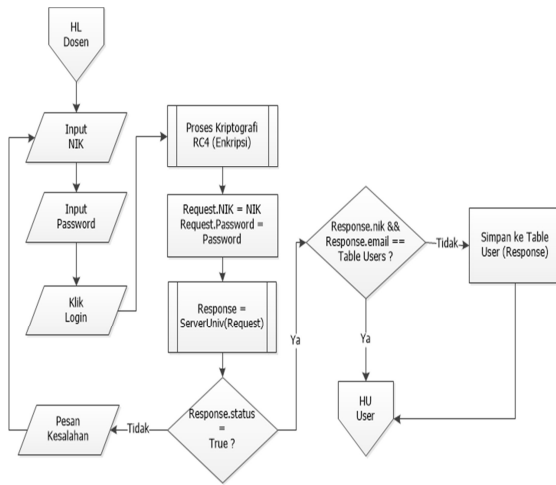
1. Tampil Halaman Login Mahasiswa
2. Input NIM
3. Input Password
4. IF Login then
5. Request.NIM = encrypt(NIM)
6. Request.password = encrypt>Password)
7. Result = SendToServerUBL(Request)
8. IF Result.status == TRUE then
9. Result = decrypt(Result.mahasiswa)
10. IF Result.NIM && Result.email != Table User then
11. saveToTableUser(Result)
12. Tampil Halaman Utama User
13. ELSE
14. Tampil Halaman Utama User
15. ELSE
16. Tampil Error
17. END IF



Gambar 3. Flowchart Proses Validasi Data Mahasiswa di Server Univeristas Budi Luhur

**Algoritma proses Validasi Data Mahasiswa di Server Univeristas Budi Luhur:**

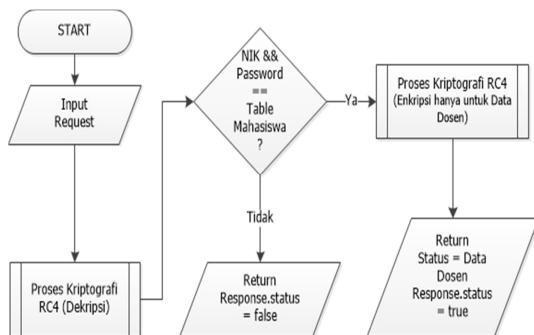
1. Input Request
2. NIM = decrypt(Request.NIM)
3. Password = decrypt(Request.password)
4. IF NIM & Password == Table Mahasiswa then
5. Result.mahasiswa = encrypt(data Mahasiswa)
6. Result.status = true
7. ELSE
8. Result.status = false
9. Result.message = tidak ditemukan.
10. SendResponseToBluCampus(Result)
11. END IF



Gambar 4. Flowchart Proses Login Dosen

Algoritma proses Login Dosen :

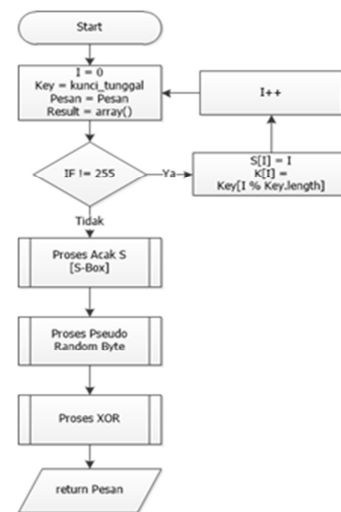
1. Tampil Halaman Login Dosen
2. Input NIK
3. Input Password
4. IF Login then
5. Request.NIK = encrypt(NIK)
6. Request.password = encrypt>Password)
7. Result = SendToServerUBL(Request)
8. IF Result.status == TRUE then
9. Result = decrypt(Result.dosen)
10. IF Result.NIM && Result.email != Table User then
11. saveToTableUser(Result)
12. Tampil Halaman Utama User
13. ELSE
14. Tampil Halaman Utama User
15. ELSE
16. Tampil Error
17. END IF



Gambar 5. Flowchart Proses Validasi Data Dosen di Server Univeristas Budi Luhur

Algoritma proses Validasi Data Dosen di Server Univeristas Budi Luhur:

1. Input Request
2. NIK = decrypt(Request.NIK)
3. Password = decrypt(Request.password)
4. IF NIM & Password == Table Dosen then
5. Result.dosen = encrypt(data Dosen)
6. Result.status = true
7. ELSE
8. Result.status = false
9. Result.message = tidak ditemukan.
10. SendResponseToBluCampus(Result)
11. END IF



Gambar 6. Flowchart Proses Algoritma RC4

Proses Enkripsi dan Dekripsi Algoritma RC4

1. Input Pesan
2. Key = Key
3. S = array(0,254)
4. IF ENKRIPSI/DEKRIPSI then
5. AcakSBox()
6. Result = PseudoRandomCode()
7. Pesan = chr(Pesan ^ Result)
8. return Pesan
9. END IF

Proses Acak S-Box Algoritma RC4

1. j = 0
2. n = Key.length
- 3.
4. FOR (i = 0 ; i < 255 ; i++) then
5. char = ord(Key[I % n])
6. j = (j + S[i] + char) % 255
7. swap(s[i],s[j])
8. END FOR



Proses Pseudo Random Code Algoritma RC4

```

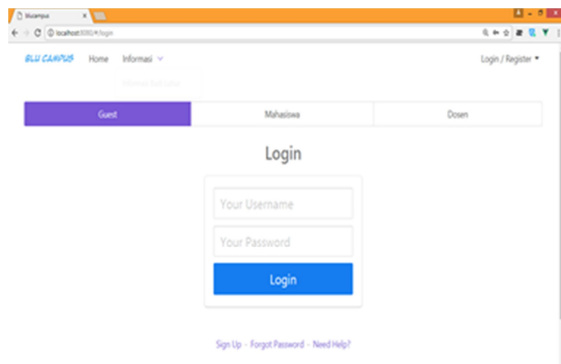
1. j = 0
2. t = array()
3. n = Pesan.length
4. data = str_split(Pesan, 1)
5. FOR (m = 0 ; m < 255 ; m++) then
6.     i = (i + 1) % 255
7.     j = (j + S[i]) % 255
8.     swap(S[i],S[j])
9.     char = ord(data[m])
10.    t[m] = (S[i] + S[j]) % 255
11. END FOR
12.
13. return t
    
```

3. HASIL DAN PEMBAHASAN

3.1. Tampilan Layar

Pada bagian ini akan dijelaskan mengenai tampilan layar bagaimana *login Guest*, Mahasiswa, dan Dosen dari sisi *client* sehingga memudahkan dalam memahami aplikasi yang sedang dibuat:

3.1.1 Tampilan Layar Halaman *Login Guest*

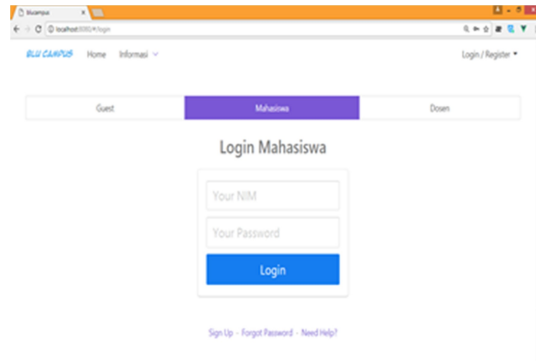


Gambar 7. Tampilan Layar Halaman *Login Guest*

Pada Gambar 7 adalah tampilan login khusus untuk *guest* yang telah mendaftar melewati menu registrasi. Di halaman ini pula pengguna hanya perlu memasukkan username dan passwordnya agar dapat masuk ke halaman utama user.

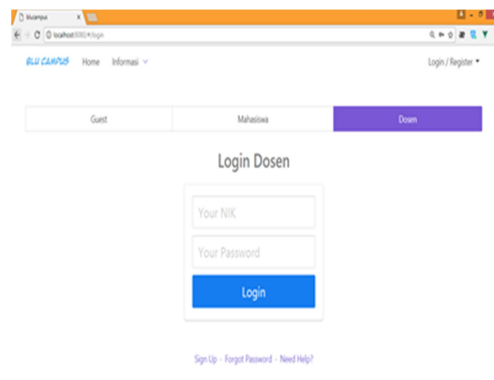
3.1.2 Tampilan Layar Halaman *Login Mahasiswa*

Pada Gambar 8 adalah tampilan *login* untuk pengguna yang berasal dari ruang lingkup Universitas Budi Luhur yaitu Mahasiswa. Mahasiswa bersangkutan (pengguna) hanya perlu memasukkan NIM dan Password yang telah dimilikinya dari Universitas Budi Luhur. Berikut adalah Gambar 8 tersebut :



Gambar 8. Tampilan Layar Halaman *Login Mahasiswa*

3.1.3 Tampilan Layar Halaman *Login Dosen*



Gambar 9. Tampilan Layar Halaman *Login Dosen*

Pada Gambar 9 adalah tampilan login yang diperuntukkan untuk Dosen dari Universitas Budi Luhur. Cukup memasukkan NIK dan Password yang telah dimilikinya, yang tersimpan di server Universitas Budi Luhur.

3.2 Uji Coba Program

Pada bagian ini, akan dilakukan pengujian enkripsi dan dekripsi algoritma *RC4* untuk membuktikan bahwa algoritma *RC4* dapat berjalan dengan baik.

Tabel 4. Hasil Enkripsi Data dengan Algoritma *RC4*

Key	Plaintext	Chipertext
Test	96timah	aMSS <sup>L</sup> ~h
testing123	Bersatukita	,!!-[5*+@Ce
UnivBudi123	indonesia123	P◀Up↑5m( r v
Un1v13ud123	Indonesiajakarta	U-? *FW#Of Vr/Lk

Tabel 5. Hasil Dekripsi Data dengan Algoritma *RC4*

Key	Chipertext	Plaintext
Test	aMSS <sup>L</sup> ~h	96timah

testing1 23	!!-[5+!@Ce	Bersatukita
UnivBu di123	P◀◀Up↑5m( r v	indonesia123
Un1v13 ud123	U-? *FW#Of Vr/L k	Indonesiajakarta

Hasil yang didapat dari proses enkripsi dan dekripsi dengan menggunakan algoritma RC4 berjalan dengan baik di dalam aplikasi, sehingga tidak perlu ragu atau takut data tidak dapat di acak (enkripsi) maupun di kembalikan (dekripsi).

### 3.3 Evaluasi Program

Kelebihan Program :

- Menanggulangi permasalahan registrasi, sehingga seseorang dari ruang lingkup Universitas Budi Luhur tidak perlu registrasi lagi.
- Memudahkan pengguna khususnya seseorang dari ruang lingkup Universitas Budi Luhur dapat *login* ke aplikasi BluCampus.
- Program yang diimplementasikan disini adalah *web*, tapi sebenarnya dapat digunakan untuk semua jenis *platform* dan bahasa pemrograman apapun.
- Algoritma RC4 cukup handal dalam proses enkripsi dan dekripsi data selama data tersebut tidak berhubungan dengan transaksi atau jenis data yang tingkat kemanannya harus diperkuat.

Kekurangan Program

- Jika *server* Universitas Budi Luhur *down* atau *maintenance*, kemungkinan seseorang dari ruang lingkup Universitas Budi Luhur tidak bisa *login*.
- Pengguna dari ruang lingkup Universitas Budi Luhur tidak dapat membuat data username karena data username tersebut sudah di atur akan sama dengan data NIM (Mahasiswa) atau NIK (Dosen) yang telah terdaftar dalam *server* Universitas Budi Luhur.
- Jika terjadi perubahan data dari sisi *server* Universitas Budi Luhur, maka kemungkinan data pengguna akan dibuat baru lagi (*fresh*, tidak ada *history* yang akan tersimpan di *server* BluCampus).

### 4. KESIMPULAN

setelah membahas semua yang ada dalam proses aplikasi, maka dapat ditarik kesimpulan mengenai pemanfaatan teknologi Web Service RestFul sebagai alat untuk pengiriman atau pertukaran data serta metode kriptografi RC4, antara lain :

- Memudahkan user yang berasal dari Universitas Budi Luhur seperti Mahasiswa dan Dosen agar dapat login masuk ke aplikasi BluCampus, tanpa perlu melakukan registrasi lagi.

- Dengan cara seperti ini user tidak perlu melakukan aktivasi lagi melalui email, karena sudah pasti email dan data user bersangkutan adalah data asli yang dapat dipertanggung jawabkan (berasal dari Universitas Budi Luhur).
- Memudahkan user yang berasal dari ruang Lingkup Universitas Budi Luhur dalam masalah mengingat password, karena cukup menggunakan data pengenalan, NIM (Mahasiswa) dan NIK (Dosen) beserta password yang berasal dari database server Universitas Budi Luhur tanpa harus mengingat password yang berasal dari BluCampus.

### 5. DAFTAR PUSTAKA

- [1] Fauizah, Y. (2013) 'Aplikasi Iklan Baris Online menggunakan Arsitektur REST Web Service', vol.9, pp. 75-80.
- [2] Ramanathan, R., Korte, T. (2014) 'Software service architecture to access weather data using RESTful web services', pp. 1-8.
- [3] Philip, M., Clync, G. (2013) 'A Performance Analysis of WS-\* (SOAP) & RESTful Web Services for implementing Service and Resource Orientated Architectures', pp. 93-100.
- [4] Putra, D. et al. (2017) 'Implementasi Algoritma RC4 dan Playfair Cipher', 16, pp. 328-334.