

MEMBANGUN LOGIN TERENKRIPSI MENGGUNAKAN BOT TELEGRAM DAN DATABASE MYSQL

Rizky Parlika¹, Lugito Michael Immanuel Prasetya², Hadiansyah Rachmawan Putra³,
Vinza Hedi Satria⁴, Faris Hirnar Pralas⁵

Jurusan Teknik Informatika, Fakultas Ilmu Komputer
Universitas Pembangunan Nasional "Veteran" Jawa Timur

rizkyparlika.if@upnjatim.ac.id¹, lugitomichael@gmail.com², hadiansyahrp.if@gmail.com³,
vinzasatria.if@gmail.com⁴, farishirnarpralas.if@gmail.com⁵

Abstrak. Login adalah sebuah fungsi yang wajib dimiliki oleh setiap sistem yang dibangun untuk memberikan batasan kepada pengguna. Kini perkembangan teknologi makin pesat. Salah satunya pada penggunaan Bot Telegram. Bot telegram kini digunakan untuk banyak hal bahkan hal yang bersifat penting. Sebuah fungsi login diperlukan untuk mengakses sebuah bot telegram. Bahkan tidak hanya login, untuk lebih mengamankan bot dari pihak lain, adanya enkripsi pun diperlukan sebuah bot. Melalui latar belakang itu lah penelitian ini dilakukan, bagaimana cara membuat sebuah bot yang memiliki fungsi login terenkripsi. Hasilnya adalah sangat memungkinkan Bot untuk menerapkan login terenkripsi, membuatnya aman untuk memproses hal penting.

Kata Kunci: Telegram, Bot, Enkripsi, MySQL

Login dari asal kata *Log* dan *In* adalah sebuah kerangka kerja yang menjadi gerbang utama bagi sebuah sistem agar dapat diakses dengan cara memasukkan identitas akun minimal memiliki *username* dan *password* (Anonim, 2013). Dikarenakan fungsinya tersebut, login menjadi sebuah fungsi yang kini berifat wajib dimiliki oleh setiap sistem dan aplikasi, termasuk aplikasi yang setiap tahunnya mulai meningkat penggunaannya yaitu IM (*Instant Messaging*) (Cokrojoyo, Andjarwirawan, & Noerjahyana, 2017). Salah satu aplikasi IM yang ada di pasaran adalah *Telegram*. Telegram memiliki banyak keunggulan daripada aplikasi lain, sebagai aplikasi pengirim pesan yang *realtime* Telegram menawarkan beberapa kemudahan seperti platform yang bermacam-macam dan juga menyediakan wadah bagi pengembang dengan cara memberikan *Open API* yang mendukung penggunaan BOT (Nufusula & Susanto, 2018). Dari sisi pengguna, bot hanya dapat berinteraksi dengan pengguna apabila pengguna sudah menyetujui bot tersebut dengan mengklik */start* namun bagaimana dengan sisi sebaliknya? Pengaplikasian bot yang bermacam-macam membuat bot tidak hanya digunakan untuk menyebarkan informasi umum saja, bot juga dapat digunakan untuk menyebarkan informasi yang bersifat rahasia seperti nilai akademis (Sastrawangsa, 2017) dan

informasi kampus (Setiaji & Paputungan, 2018), serta absensi (Chaniago & Junaidi, 2016) atau bahkan Penggunaan telegram untuk konsep keamanan rumah (Sharma & Sharma, 2019) (Kurniawan, Sunarya, & Tulloh, 2018) dan pengaplikasian Telegram untuk pekerjaan-pekerjaan yang jauh lebih kompleks lagi. Untuk itu dibutuhkan fungsi login untuk mengakses sebuah bot yang memiliki fungsi seperti di atas. Selain pengamanan dengan login dibutuhkan keamanan lain dikarenakan meningkatnya kebutuhan akan keamanan data seiring dengan meningkatnya perkembangan teknologi di era kini. Data merupakan hal penting dan kekhawatiran besar akan muncul apabila sebuah rangkaian data menghilang atau disadap (Liana, Sutardi, & Muchlis, 2018). Di era kini, transaksi data sama seperti halnya penggunaan bot, data yang berputar memiliki tingkat keamanan yang tinggi seperti data keuangan bank, identitas kependudukan ataupun data perekrutan pegawai (Parlika, Munir, & Pribadi, 2016). Dibutuhkan teknologi yang merubah bentuk atau wujud data agar data tersebut menjadi tidak mudah dipahami oleh pihak lain. (Hermawan, 2019)

Untuk memecahkan masalah yang telah dijabarkan, akan dibuat sebuah bot yang memiliki fungsi *Log in* terenkripsi. Untuk

membuat bot dengan fungsi tersebut dibutuhkan sebuah media penyimpanan data *log in*. Untuk pemecahan masalah basis data, akan digunakan basis data *MySQL* alasannya adalah karena *MySQL* merupakan basis data yang gratis dan mudah digunakan karena memiliki tampilan GUI. Belum lagi dukungan sebuah paket lengkap *webserver* bernama *XAMPP* (Yu-Huang, 2019). Sehingga terpilihnya *MySQL* juga membuat *PHP* menjadi pilihan bahasa pemrograman yang akan digunakan karena merupakan satu *bundle* dengan *XAMPP*.

Melalui penjabaran yang telah dilakukan diatas, dapat ditarik sebuah tema penelitian mengenai pembuatan login terenkripsi melalui *Bot_telegram* menggunakan basis data *MySQL*.

I. Metodologi

Sistem login yang akan dibentuk nanti merupakan sebuah sistem login *Bot* yang menggunakan *Bot Telegram* sebagai media *output*, basis data *MySQL* sebagai media penyimpanan dan *website* berbasis *PHP*, *CSS*, *HTML* yang berfungsi sebagai media registrasi. *HTML* digunakan sebagai media registrasi untuk membuka penelitian baru kedepannya. *Bot* akan menggunakan bahasa *PHP* dengan mengambil *API telegram* melalui *bot_father*.

Alur Penelitian

Penelitian ini terdiri dari tiga fase, adapun ketiga fase itu adalah :

A. Perancangan

Pada tahap perancangan, di rancang semua kerangka yang berhubungan dan yang akan digunakan oleh sistem nanti. Perancangan adalah pembangunan pondasi dari sistem yang hendak dibuat nanti, oleh karena itu Hal-hal yang akan di rancang adalah seperti *CDM - PDM*, *Flowchart*, dan banyak lagi. Namun pada penelitian kali ini, peneliti akan menggunakan *CDM* dan *Flowchart* sebagai kerangka dari sistem login terenkripsi dengan menggunakan *bot telegram* yang akan di buat nanti.

B. Implementasi

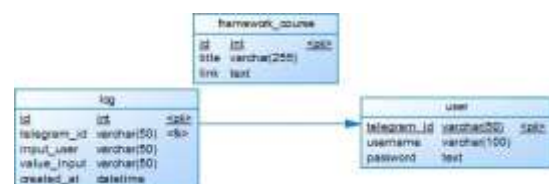
Implementasi adalah fase realisasi sistem yang sudah di rancang atau dapat dibilang tahap pengkodean. Secara keseluruhan, sistem yang dirancang menggunakan bahasa *PHP* baik untuk *bot* dan *web*, untuk *web* ditambah dengan *HTML*, *CSS*, *Javascript* untuk pemrosesan data dan tampilan, sedangkan untuk *bot PHP* akan di koneksikan melalui *webhook* agar dapat diakses oleh server *telegram*. Lalu untuk penyimpanan data, akan digunakan *DBMS (Database Management System) MySQL*.

C. Pengujian

Pengujian dilakukan dengan cara mencoba semua fungsi dari sistem untuk menjawab pertanyaan apakah sistem berjalan sesuai dengan harapan penulis atau masih terdapat kekurangan-kekurangan yang perlu di perbaiki. Uji coba akan dimulai dari titik paling awal *bot* hingga titik akhir.

Desain Basis Data

Seperti pemaparan sebelumnya, *MySQL* akan menjadi basis data yang akan digunakan. Selanjutnya adalah membangun kerangka basis data. Kerangka basis data dibangun menggunakan alat bantu *Power Designer*.



Gambar 1 – CDM Basis Data

Dari gambar 1 dapat dilihat bahwa sistem yang akan dibuat memiliki tiga buah tabel dengan satu relasi yaitu antara tabel *log* dan *user*. Tabel *log* berfungsi untuk merekam percakapan yang terjadi antara *bot* dengan pengguna, adapun isi dari tabel *log* adalah *user_id* yang berfungsi sebagai tempat *bot* mengenali siapa lawan bicaranya, *input_user* untuk mengidentifikasi input dari pengguna lalu ada *value_input* dan *created_at* untuk

mengetahui kapan aktivitas itu berlangsung. Secara keseluruhan, tabel log berperan untuk melakukan perekaman input dari pengguna, termasuk diantaranya adalah saat pengguna akan memasukkan inputan *username* dan *password*. Sedangkan tabel *user* berfungsi sebagai tabel utama fungsi *login* dimana di sinilah data *login* akan disimpan dan digunakan untuk memeriksa pengguna yang akan mengakses fitur bot, didalam tabel *user* terdapat *username* dan *password* lalu untuk menghubungkan antara tabel *log* dan *user* akan diambil *telegram_id* sebagai *foreign_key*. Tabel ke tiga adalah tabel *framework_course* yang berfungsi sebagai menu di dalam bot nanti, isi dari *framework_course* adalah *tittle* dan *link*.

Alur Sistem (Bot)

Kerangka alur sistem dibuat untuk memberikan gambaran jelas bagi penulis maupun pembaca mengenai sistem yang akan dibuat. Untuk memberikan gambaran mengenai alur dari sistem, digunakan *Flowchart* yang menjelaskan alur sistem dari awal pengguna mengetikkan */start* hingga akhir fungsi bot.



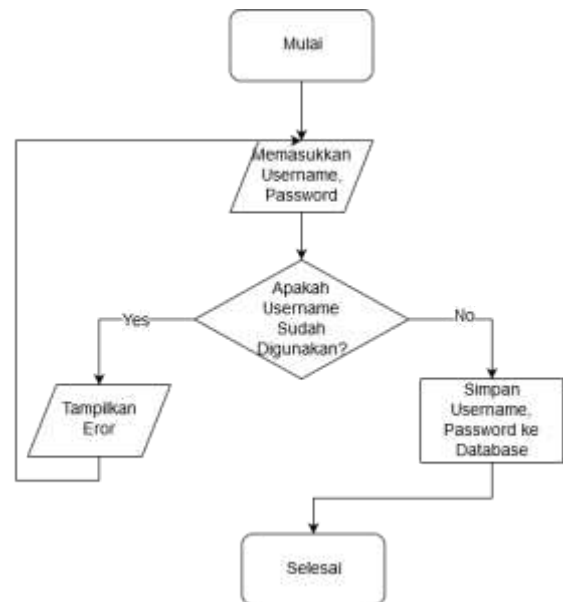
Gambar 2 – Flowchart Bot

Setelah mengetik */start*, pengguna akan diminta memasukkan *username*. Dari sana, bot akan memutuskan apakah *username* sudah terdaftar atau belum. Jika belum, maka bot akan menampilkan pesan yang

memerintahkan pengguna untuk mendaftar. Bila tidak maka pengguna akan diminta memasukkan *password*. Tidak berhenti sampai *password*, bot akan memeriksa input *password* dari pengguna. Bila *password* yang dimasukkan benar, maka bot akan menampilkan menu yang disediakan (dalam penelitian kali ini adalah menu mata kuliah). Bila bot mendeteksi pengguna gagal memasukkan *password* maka bot akan memulai penghitungan dan jika telah gagal sebanyak 3 kali maka bot akan menampilkan pesan gagal dan pengguna harus mengulang alur program dari awal.

Alur Sistem (Web)

Setelah alur dari bot sudah dijabarkan, selanjutnya adalah penjabaran alur sistem dari *website*. Alur sistem *website* lebih sedikit karena hanya berfungsi sebagai tempat registrasi saja. Adapun alur sistem dari *website* adalah seperti gambar dibawah.



Gambar 3 – Flowchart Website

Website dimulai dengan memasukkan *username* dan *password* untuk melakukan registrasi. Lalu web akan menentukan apakah *username* sudah digunakan atau belum untuk mengurangi kemungkinan yang tidak diinginkan. Jika Ya maka akan ditampilkan eror dan pengguna akan dikembalikan ke form registrasi. Jika

tidak maka *username* akan disimpan bersama dengan *password* ke dalam basis data.

Sistem Enkripsi

Sistem enkripsi dibangun dari nol dalam artian tidak menggunakan sistem enkripsi yang sudah ada. Metode enkripsi yang digunakan adalah enkripsi biner dengan rumus enkripsi sebagai berikut.

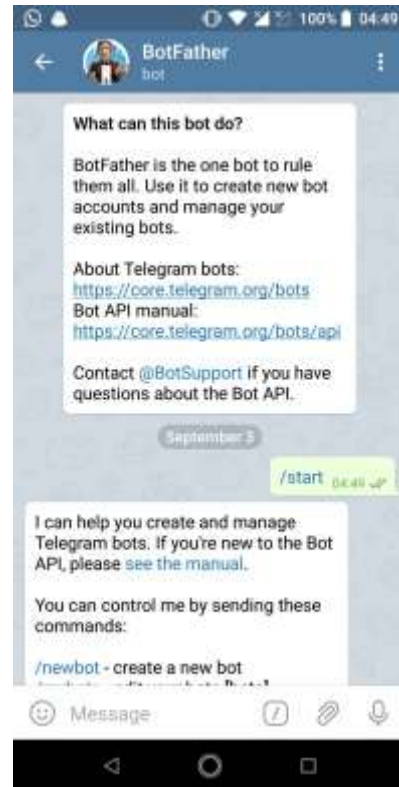
```
protected function
encryptPass($password){
$encryptedPass = '';
$arPass = str_split($password);
foreach ($arPass as $key =>
$char) {
$asciiChar = ord($char);//char to

ascii int
$binaryChar =
decbin($asciiChar);//int to binary
0101
$encryptedPass .= $binaryChar;
}
return $encryptedPass;
}
```

Untuk menambah kecepatan respon sistem terhadap input pengguna, maka enkripsi hanya dilakukan pada parameter *password* saja. Peneliti menilai *password* lah yang memiliki fungsi krusial dimana isinya sangat rahasia, berbeda dengan *username* yang menjadi tampilan umum ataupun *telegram_id* yang dapat di temukan oleh bot manapun.

API Telegram

Agar dapat menggunakan API dari Telegram, tentu saja langkah pertama yang harus dilakukan adalah melakukan pendaftaran API ke BotFather



Gambar 4 – Bot Father

Setelah terdaftar, pengguna akan diberikan sebuah token API, token itu lah yang akan digunakan untuk melakukan aktivitas dengan bot.

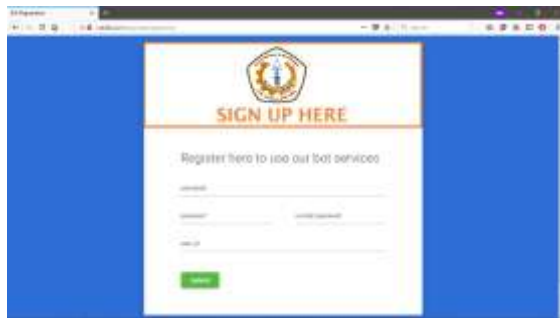
II. Hasil dan Pembahasan

Masuk kepada fase ke tiga dari penelitian kali ini, yaitu fase pengujian. Sistem yang sudah dibuat dicoba untuk melihat apakah sudah berjalan sesuai keinginan peneliti. Pengujian dimulai dari awal sejak di klik */start*.



Gambar 5 – username ditolak

Dapat dilihat pada gambar 4 bahwa setelah memulai penggunaan bot. Bot akan meminta sebuah *username* lalu dimasukkan sebuah *username* yang belum terdaftar. Pada kesempatan kali ini sebut saja *asiimov* maka bot akan meminta pengguna untuk masuk ke sebuah link *website* untuk mendaftar.



Gambar 6 – Tampilan utama

Di dalam *website* terdapat form registrasi seperti pada gambar. Pengguna melakukan pengisian ke dalam form yang ada di gambar. Setelah semua form sudah diisi, klik *submit*. Maka pengguna sudah terdaftar.

Setelah mendaftarkan diri melalui *website* selanjutnya kembali ke bot untuk mencoba lagi. Pertama-tama klik */start* lagi untuk memulai dari awal lalu ketikkan *username* yang sudah terdaftar.



Gambar 7 – username diterima

Setelah itu masukkan *password* dari *username* yang sudah di registrasikan sebelumnya. Untuk percobaan kali ini, peneliti mencoba menyalahkan *password* untuk melihat respon dari bot.



Gambar 8 – Password Salah

Selanjutnya, di masukkan *password* yang benar maka bot akan merespon seperti gambar berikut ini.



Gambar 9 – Password Berhasil

Maka bot baru membuka fiturnya kepada pengguna yang telah melakukan login terlebih dahulu, seperti pada gambar.

Setelah semua pengujian tampak depan sistem sudah berjalan dengan baik, selanjutnya adalah melakukan pengujian bagian belakang sistem yaitu pada enkripsi, seperti yang sudah diketahui bahwa *password* dari sistem akan di enkripsi sehingga susah untuk ditembus. Untuk menguji apakah enkripsi sudah terjadi atau belum adalah

dengan melakukan pemeriksaan terhadap data yang sudah tersimpan di dalam basis data. Untuk memeriksa data tersebut digunakan alat bantu *phpmyadmin*.



Gambar 10 – Isi Basis Data

Pada tabel *password* terlihat bahwa seluruh data berbentuk deretan panjang biner (terdiri dari 1 dan 0) yang membuktikan bahwa enkripsi yang dipasang oleh peneliti berjalan dengan baik dan sesuai dengan keinginan atau tujuan awal penelitian.

III. Simpulan

Setelah dilakukan penelitian mengenai *login* terenkripsi menggunakan Bot Telegram dengan basis data MySQL. Dapat diambil kesimpulan bahwa hal tersebut sangat mungkin dilakukan. Telegram selaku *Instant Messenger* yang membuka API nya agar dapat di akses oleh pengembang memberikan banyak sekali kemungkinan pengembangan yang hampir tidak terbatas, salah satunya adalah pembuatan login menggunakan bot.

Kedepan sangat memungkinkan untuk tidak hanya login biasa, namun terdapat login- login jenis lain seperti menggunakan wajah (scan muka) yang di kombinasikan dengan bot telegram.

Selain itu,. Bot dapat menyimpan id setiap orang yang mengaksesnya dan dapat mengunci diri apabila id tersebut belum terdaftar, maka sangat memungkinkan kedepannya bot telegram menyelesaikan pekerjaan yang sifatnya jauh lebih rahasia dan krusial.

Saran kedepan untuk penelitian-penelitian selanjutnya adalah menggunakan desain basis data yang lebih kompleks lagi.

IV. Daftar Pustaka

1. Anonim. (Juni 2013). *Pengertian Login*. Hämtat från Temukan pengertian: <https://www.temukanpengertian.com/2013/06/pengertian-login.html>
2. Chaniago, M. B., & Junaidi, A. (2016). Student Presence Using RFID and Telegram Messenger Application. *WIDYATAMA INTERNATIONAL SEMINAR* (ss. 1-5). Bandung: Universitas Widyatama.
3. Cokrojoyo, A., Andjarwirawan, J., & Noerjahyana, A. (2017). Pembuatan Bot telegram Untuk Mengambil Informasi dan Jadwal Film Menggunakan PHP. *INFRA*, 1-4.
4. Hermawan. (den 16 Mei 2019). *Pengertian enkripsi Beserta Cara Kerja dan Jenis-Jenis Enkripsi*. Hämtat från Nesabamedia: <https://www.nesabamedia.com/pengertian-enkripsi/>
5. Kurniawan, M. I., Sunarya, U., & Tulloh, R. (2018). Internet of Things : Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger. *ELKOMIKA*, 1-15.
6. Liana, Sutardi, & Muchlis, N. F. (2018). APLIKASI ENKRIPSI DAN DEKRIPSI DATA MENGGUNAKAN TINY ENCRYPTION ALGORITHM (TEA) BERBASIS JAVA. *SemanTIK*, 39-48.
7. Nufusula, R., & Susanto, A. (2018). Rancang Bangun Chat Bot Pada Server Pulsa Menggunakan Telegram Bot API. *Journal of Information System (JOINS)*, 80-87.
8. Parluka, R., Munir, M. S., & Pribadi, D. U. (2016). Rancang Bangun Aplikasi Rekrutmen Online Menggunakan Verifikasi Otomatis Di Lingkungan UPN Veteran Jawa Timur. *SCAN*, 61-69.
9. Sastrawangsa, G. (2017). Pemanfaatan Telegram Bot

- Untuk automatisasi Layanan Dan Informasi Mahasiswa Dalam Konsep Smart Campus. *Konferensi Nasional Sistem & Informatika 2017* (ss. 772-776). Bali: STMIK STIKOM.
10. Setiaji, H., & Paputungan, V. I. (2018). Design of Telegram Bots for Campus Information Sharing. *IOP Conference Series: Materials Science and Engineering* (ss. 1-7). IOP Publishing.
 11. Sharma, K. H., & Sharma, M. (2019). IoT Based Home Security System. *International Conference on Sustainable Computing in Science, Technology & Management* (ss. 583-588). Jaipur: Elsevier SSRN.
 12. Yu-Huang, C. (2019). Learning Database through Developing Database Web. *International Journal of Information and Education Technology*, 241-249.
 13. Source Code Form Login:
<https://colorlib.com/wp/free-bootstrap-registration-forms/>
 14. Source Code Animasi Welcome :
https://www.w3schools.com/w3css/w3css_animate.asp