Implementasi Metode *Membership* dengan Menggunakan Konsep Kriptografi *Digital Signature*

Natanael Dias – 18220051 (*Author*)
Program Studi Sistem dan Teknologi Informasi
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jalan Ganesha 10 Bandung
E-mail: natanaeldias27.ds@gmail.com

***Abstract*—Bidang teknologi yang berkembang saat ini adalah media sosial sebagai platform untuk memenuhi kebutuhan manusia. Salah satu fitur yang muncul dalam memenuhi kebutuhan yaitu *membership*. Dengan fitur ini semua pihak baik pemilik konten dan penikmat akan mendapatkan benefit masing-masing. Namun, dibutuhkan proses otentikasi pengguna berstatus *membership* agar tidak sembarang pengguna bisa mendapatkan benefit yang ditawarkan. Dengan menggunakan ilmu kriptografi, proses otentikasi tersebut dapat terjamin lebih baik. Bentuk kriptografi yang digunakan adalah *digital signature* menggunakan kombinasi *hash* dan kriptografi kunci public. Dengan penggunaan ini, sistem dapat melakukan otentikasi pengguna dengan membandingkan *digital signature* yang tersimpan pada *database* ketika pengguna membeli *membership* dan generasi *digital signature* menggunakan *message* berupa *username* pengguna.**

***Keywords—Membership*, Kriptografi, *Digital Signature*, Otentikasi**

I. PENDAHULUAN

Dewasa ini, kebutuhan manusia semakin bertambah banyak dan tidak dapat dipungkiri dengan adanya perkembangan zaman, teknologi juga semakin berkembang. Teknologi inilah yang sedikit demi sedikit mengisi kebutuhan manusia saat ini. Salah satu teknologi yang ada pada saat ini adalah media sosial. Media sosial merupakan sebuah media untuk bersosialisasi satu sama lain dan dilakukan secara *online* yang memungkinkan manusia untuk saling berinteraksi tanpa dibatasi ruang dan waktu[1].

Meskipun fungsi utama media sosial adalah memenuhi kebutuhan sosial para penggunanya, tetapi seiring berjalan waktu, media sosial juga dijadikan sebagai media pemenuhan kebutuhan lain, salah satunya adalah kebutuhan ekonomi. Antar pengguna dapat melakukan transaksi satu dengan lainnya, menjadikan media sosial sebagai lahan mencari duit, dan lainnya. Hal ini dapat terjadi karena adanya kebebasan di media sosial. Bahkan, *platform* pengembang media sosial tersebut juga menambahkan fitur yang mendukung pemenuhan kebutuhan eknomi para penggunanya.

Beberapa contoh media sosial yang mendukung hal tersebut adalah YouTube dengan *membership* dan *adsense*, TikTok dengan *gift* dan TikTok *shop*, Twitch dengan Bits, Discord dengan Nitro, serta media sosial lainnya. Namun, yang akan menjadi fokus kali ini adalah metode *membership*. Metode *membership* merupakan model bisnis yang memberikan penawaran terhadap calon pelanggan untuk mendapatkan berbagai keuntungan, poin *loyalty*, dan informasi penting seputar *brand* secara gratis maupun berbayar[2]. Dengan metode ini, pemilik yang disebut *content creator* dapat menerima keuntungan yang lebih dari penggunanya dan pengguna juga mendapatkan benefit yang sesuai.

Salah satu metode *membership* yang ramai didengar dan diperbicangkan adalah *membership channel* YouTube. Beberapa benefit yang didapatkan oleh pengguna adalah akses ke konten-konten khusus member, seperti video, *live stream*, dan *community posts*, memakai emoji khusus di kolom komentar ataupun *live chat*, serta lambang pada akun. YouTube juga menyediakan fitur Premium yang menawarkan pengguna dapat menonton video tanpa gangguan iklan. Selain itu, ada pula Nitro pada Discord. Nitro merupakan fitur langganan per bulan yang ditawarkan pada penggunanya. Dengan fitur tersebut, pengguna Discord akan mendapatkan benefit lebih seperti lambang khusus nitro, kostumisasi profil, menambah batas ukuran berkas yang diunggah, HD *video streaming*, dan lainnya.

Namun, metode ini mengharuskan sistem untuk melakukan autentikasi pengguna terlebih dahulu karena pengguna non-*membership* tidak dapat mengakses benefit tersebut. Maka dari itu, hadirlah konsep kriptografi berupa *digital signature* yang akan membantu proses autentikasi tersebut. Jika pengguna membeli *membership* maka akan secara otomatis di-*generate digital signature* dan ketika mengakses profil *channel* tertentu akan diautentikasi terlebih dahulu apakah pengguna tersebut merupakan *member*. Dengan digunakannya metode ini, *database* mengenai status *membership* pengguna menjadi tidak dibutuhkan.

II. DASAR TEORI

A. Kriptografi

Kriptografi berasal dari bahasa Yunani, yaitu *kryptos* dan *graphein*[3]. *Kryptos* berarti rahasia atau tersembunyi dan *graphein* berarti menulis. Jadi, kriptografi umumnya adalah proses menulis atau menyampaikan pesan secara rahasia dan tersembunyi. Kriptografi dapat digunakan secara klasik tanpa memerlukan bantuan komputer dan juga secara digital.

Kriptografi klasik biasanya dilakukan menggunakan alat bantu pena, batu, kertas, dan alat tradisional lain. Sedangkan untuk kriptografi digital dibutuhkan bantuan komputer dengan penggunaan enkripsi dekripsi dan *plaintext ciphertext*.

Terdapat beberapa jenis kriptografi sesuai dengan penggunaannya. Beberapa diantaranya adalah *hash function* untuk meringkas data dan mengirim penjelasan yang sudah dirangkum, *public key cryptography* yang memanfaatkan kunci publik dan privat, serta *symmetric key cryptography* yang menggunakan satu kunci untuk melakukan enkripsi dan dekripsi. Kriptografi juga berfungsi untuk memenuhi beberapa kebutuhan. Kebutuhan tersebut sering dikategorikan sebagai kebutuhan non-fungsional sebuah sistem. Kebutuhan tersebut antara lain:

1. *Authentication*

   Pengirim dan penerima dapat mengetahui identitas asli satu sama lain.

2. *Confidentiality*

   Informasi dapat terlindungi dari pihak yang tidak memiliki akses.

3. *Integrity*

   Pengirim dan penerima mendapatkan ataupun mengirim pesan tanpa perubahan data apapun.

4. *Non-Repudiation*

   Pengirim tidak dapat menyangkal bahwa ialah yang mengirim sebuah informasi.

B. RSA

RSA merupakan salah satu algoritma yang menerapkan enkripsi *public key*[4]. RSA seringkali dipakai karena tingkat kemanannya yang tinggi karena sulitnya menentukan faktor prima dari bilangan bulat yang bernilai besar. RSA sendiri merupakan singkatan dari ketiga penemunya, yaitu Rivest, Shamir, dan Adleman.

Konsep ini menggunakan kunci publik sebagai kunci yang digunakan untuk mengenkripsi sebuah pesan. Kunci publik juga disebarkan, maka dimiliki oleh semua orang. Terdapat kunci lain yaitu kunci privat. Kunci privat hanya dipegang oleh pemiliknya saja dan digunakan untuk melakukan dekripsi pesan.

Terdapat beberapa komponen yang menunjang proses algoritma RSA, antara lain:

1. p dan q sebagai bilangan prima, bersifat rahasia
2. n sebagai hasil perkalian p dan q, bersifat tidak rahasia
3. $\phi(n)$ sebagai hasil perkalian (p – 1) dan (q – 1), bersifat rahasia
4. e sebagai kunci enkripsi (kunci publik), bersifat tidak rahasia
5. d sebagai kunci dekripsi (kunci privat), bersifat rahasia
6. m sebagai *plain text*, bersifat rahasia
7. c sebagai *cipher text*, bersifat tidak rahasia

Algoritma RSA terdiri dari 3 tahap utama, yaitu:

1. Pembangkitan Sepasang Kunci

   Melakukan perhitungan dengan menggunakan p dan q. Hasil perhitungan akan didapatkan n dan $\phi(n)$. Hasil algoritmanya adalah kunci publik (pasangan e dan n) dan kunci privat (pasangan d dan n)

2. Enkripsi

   Membuat pesan menjadi blok *plain text* yang lebih kecil (m_1, m_2, m_3, …), lalu menghitung *cipher text* dengan persamaan $c_i = m_i^e \bmod n$.

3. Dekripsi

   Didapatkan *ciphertext* pada blok-blok tertentu (c_1, c_2, c_3, …). Untuk setiap blok hitung kembali *plain text* dengan persamaan $m_i = c_i^d \bmod n$.

RSA pada *digital signature* diterapkan secara terbalik. Kunci publik yang tadinya menjadi kunci enkripsi, akan dijadikan sebagai kunci dekripsi. Begitupun dengan kunci privat yang tadinya menjadi kunci dekripsi, akan dijadikan kunci enkripsi.

C. SHA3 (*Hash*)

Fungsi *hash* merupakan sebuah fungsi yang akan mengompresi sebuah pesan yang memiliki ukuran sembarang dan mengubahnya menjadi sebuah *string* dengan ukuran yang ditetapkan[5]. *String* hasil *hash* tersebut dinamakan *message-digest* atau *hash value*. Fungsi *hash* dapat dinyatakan dengan h = H(M), dengan h sebagai *hash value*, H sebagai *hash function*, dan M sebagai pesan.

Fungsi *hash* juga digunakan pada konsep SHA-3 (Keccak)[6]. Dengan menggunakan SHA-3, *message digest* yang dihasilkan akan berukuran sembarang, maka dari itu algoritma ini memiliki tingkat keamanan yang terbilang tinggi. Keccak menggunakan konstruksi spons dalam penerapannya. Terdapat 3 proses pada konstruksi spons:

1. Praproses

   Proses menambahkan *padding* pada pesan agar dapat habis dibagi bilangan tertentu (r) dan memotong hasil tersebut menjadi blok-blok berukuran r-bit.

2. Penyerapan (*absorsing*)

   Melakukan pehirungan XOR untuk setiap blok masukan (P_i), lalu dimasukkan ke fungsi permutasi f untuk menghasilkan *state* yang baru. Lalu dilanjutkan ke fase pemerasan ketika semua blok masukan selesai diproses.

3. Pemerasan (*squeezing*)

   Hasil *hash value* disimpan pada Z yang berawal dengan inisialisasi Z dengan *string* kosong. Jika panjang Z belum = d, maka r-bit pertama *state* S akan disambungkan ke Z. Jika panjang Z masih belum sama = d, maka gunakan fungsi permutasi f untuk

mengahasilkan *state* baru S. c bit terakhir tidak akan terpengaruh secara langsung oleh blok masukan dan tidak akan mengeluarkan *output* selama fase pemerasan berlangsung untuk mencegah terjadinya kolisi.

Berikut gambaran konstruksi spons pada SHA-3 (Keccak).

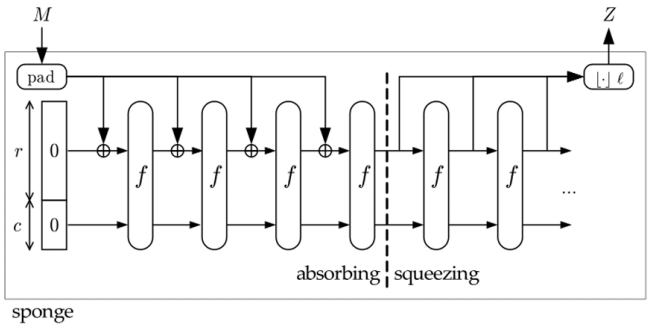

Fig. 1. Konstruksi Spons

(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/15-SHA-3-2023)

D. *Digital Signature*

Dalam kehidupan sehari-hari, tanda tangan digunakan sebagai bukti otentik suatu dokumen cetak[7]. Tanda tangan memilik beberapa karekteristik, antara lain:

1. Bukti yang otentik
2. Tidak dapat dilupakan
3. Tidak dapat dipindah untuk di-*reuse*
4. Isi dokumen yang sudah ditandatanganin tidak dapat diubah
5. Mengikuti prinsip *non-repudiation*, yaitu tidak dapat disangkal

Saat ini, banyak dokumen yang berbentuk digital. Maka dari itu, tanda tangan yang digunakan untuk data digital dinamakan *digital signature*. Fungsi *digital signature* sama seperti tanda tangan biasa, yaitu untuk otontikasi, tetapi untuk data yang bersifat digital. *Digital signature* menerapkan prinsip kriptografi yang memiliki ketergantungan dengan isi pesan dan kunci. Maka dari itu, *digital signature* tidak sama dengan *digitized signature* (tanda tangan yang didigitisasi) dengan cara difoto ataupun dipindai.

Jika tanda tangan fisik seseorang pada tiap dokumen adalah sama, maka tanda tangan digital akan selalu berbeda pada masing-masing pesan atau pada masing-masing kunci. Terdapat dua proses dalam tanda tangan digital, yaitu:

1. *Signing*

   Memberikan tanda tangan digital pada pesan.

2. *Verification*

   Memastikan tanda tangan adalah asli atau tidak dengan memeriksa keabsahannya.

Dalam menandatangani pesan digital, terdapat 2 cara yang dapat dilakukan tergantung tingkat kerahasiaan sebuah pesan. Pertama, dengan cara enkripsi pesan, cara ini dilakukan secara khusus untuk pesan yang dinilai rahasia. Kedua, dengan menggunakan kombinasi *hash* dan kriptografi kunci publik untuk pesan yang tidak dinilai rahasia.

Cara penandatanganan digital menggunakan enkripsi pesan dibagi lagi menjadi 2 metode. Pertama yaitu menggunakan algoritma kriptografi kunci simetri. Metode ini memerlukan pihak ketiga yang terpercaya untuk mengatasi masalah penyangkalan yang ada. Fungsi pihak ketiga ini, atau biasa disebut penengah, adalah memberikan kunci rahasia kepada pihak pengirim dan penerima. Kunci rahasia tersebut hanya diketahui oleh masing-masing penerima kunci dan penengah. Berikut gambaran cara penandatanganan digital menggunakan enkripsi pesan dengan metode algoritma kriptografi kunci simetri.

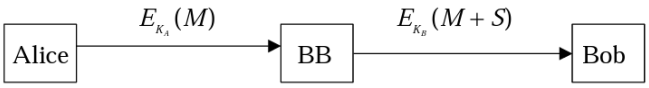

Fig. 2. Proses *Digital Signature* Algoritma Kriptografi Kunci Simetri

(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/16-Tanda-tangan-digital-2023.pdf)

Metode kedua yaitu menggunakan algoritma kriptografi kunci publik. Namun, hal ini agak berbeda karena cara yang umum tidak dapat melakukan proses otentikasi pengirim pesan. Maka dari itu pesan akan dienkripsi menggunakan kunci privat pengirim dan didekripsi menggunakan kunci publik pengirim. Dengan demikian, kerahasiaan pesan tetap terjaga dan proses otentikasi dapat dilakukan. Berikut gambaran cara penandatangan digital menggunakan enkripsi pesan dengan metode algoritma kriptografi kunci publik.

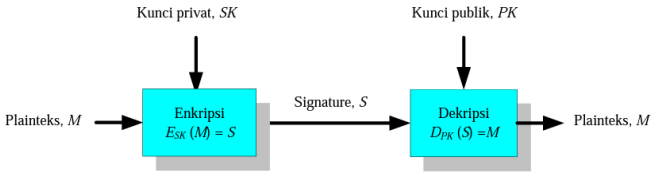

Fig. 3. Proses *Digital Signature* Algoritma Kriptografi Kunci Publik

(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/16-Tanda-tangan-digital-2023.pdf)

Cara penandatanganan menggunakan kombinasi *hash* dan kriptografi kunci publik menerapkan algoritma RSA dan ElGamal Signature. Proses penandatangan diawali dengan menghitung nilai *hash* pesan, lalu dilanjut dengan mengenkripsi hasil *hash* dengan kunci privat pengirim menggunakan RSA, hasil yang didapat yaitu S akan dikirimkan bersamaan dengan pesan. Dalam hal

memverifikasi, proses dimulai dengan menghitung nilai *hash* pesan yang diterima, lalu melakukan dekripsi terhadap S menggunakan kunci publik pengirim menggunakan RSA, setelah itu hasil *hash* yang didapat akan dibandingkan dengan hasil dekripsi S. Berikut gambaran cara penandatanganan menggunakan kombinasi *hash* dan kriptografi kunci publik.

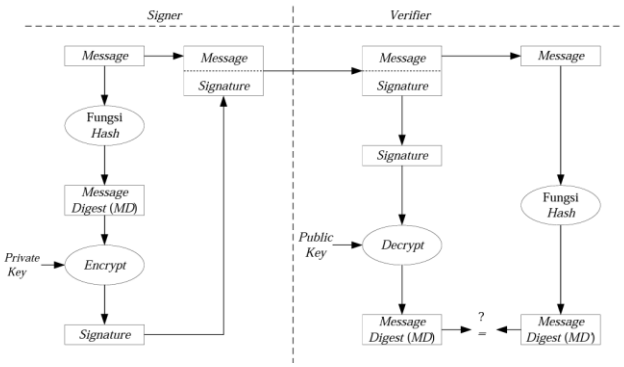

Fig. 4. Proses *Digital Signature* Kombinasi *Hash* & Kriptografi Kunci Publik

(sumber: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/16-Tanda-tangan-digital-2023.pdf)

III. RANCANGAN SOLUSI DAN IMPLEMENTASI

Dari masalah yang diangkat sebelumnya, yaitu proses otentikasi pengguna yang membeli fitur *membership*, diberikan sebuah solusi yaitu *digital signature* dengan menggunakan cara kombinasi *hash* dan kriptografi kunci publik. Solusi ini menerapkan konsep kriptografi, penggunaan RSA, penggunaan SHA3 (*hash*), dan *digital signature* yang tertera pada dasar teori sebelumnya. Solusi pengecekan *membership* ini dirancang dengan berlandaskan kepada metode *membership* berbagai *channel content creator* pada YouTube. Dalam rancangan kali ini, pembelian, pembatalan, ataupun masa *membership* sudah habis belum ikut diimplementasikan karena implementasi hanya berfokus pada otentikasinya saja, digunakan pula asumsi bahwa *username* tiap pengguna bersifat unik.

Pada solusi sistem yang dirancang pengguna akan melakukan *login* terlebih dahulu dan sistem akan mengecek apakah *username* dan *password* yang dimasukkan sesuai dengan yang tersimpan pada *database*. Jika tidak sesuai, maka pengguna akan diperintah untuk melakukan *login* kembali. Sedangkan jika sesuai maka pengguna akan masuk ke halaman utama. Halaman utama akan memperbolehkan pengguna untuk memilih *channel* yang akan diakses ataupun *logout*. Jika memilih *channel* maka pengguna akan dicek terlebih dahulu apakah dia memiliki *membership* atau tidak. Pengguna non-*membership* dapat mengakses fitur normal ataupun biasa, sedangkan pengguna *membership* akan mendapatkan akases fitur tambahan sebagai benefit yang didapat. Sistem akan berhenti kalau pengguna memilih opsi *logout*. Berikut merupakan diagram alir solusi sistem yang akan dirancang dari sisi pengguna.

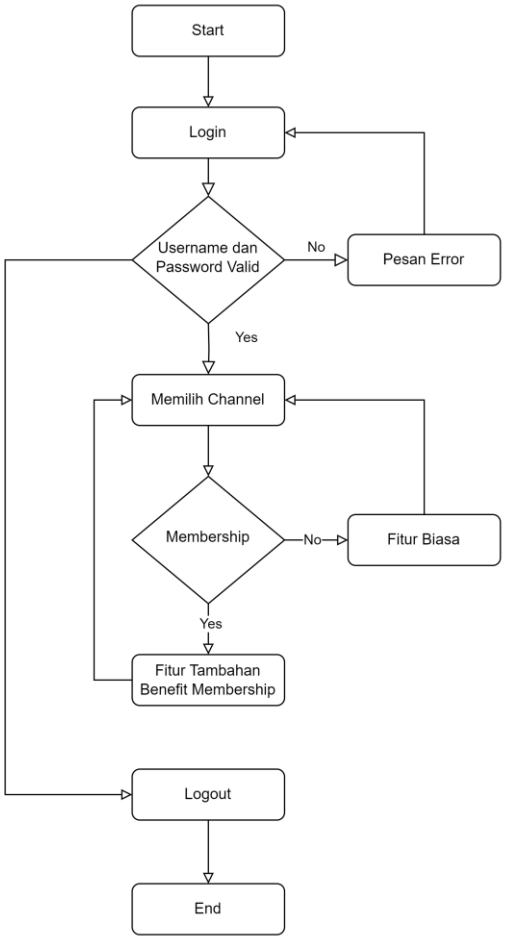

Fig. 5. Diagram Alir Solusi Sistem dari Sisi Pengguna

Dalam proses jalannya sistem, terdapat penandatanganan digital jika pengguna membeli *membership* dan juga verifikasi *digital signature*. Pada penandatanganan digital, pengguna harus terlebih dahulu membeli *membership* yang dilakukan di luar implementasi sistem. Namun, proses penandatanganan digital tetap diuji. Dengan melihat gambaran kerja *digital signature* menggunakan kombinasi *hash* dan kriptografi kunci publik, terdapat beberapa komponen pada proses *signer*, yaitu pesan, fungsi *hash*, *message digest*, proses enkripsi dengan kunci privat, dan menghasilkan *signature*. Isi pesan dalam penerapan solusi ini adalah *username* tiap pengguna. Lalu akan dijalankan fungsi *hash* menggunakan SHA-3 dan menghasilkan *message digest*. *Message digest* akan dienkripsi menggunakan kunci privat *channel* yang bersangkutan dan akan menghasilkan *digital signature*. *Digital signature* akan disimpan ke dalam *database* yang berkaitan dengan *membership channel* tersebut. Proses penandatanganan digital ini berada di luar alur solusi sistem yang akan diimplementasi. Berikut diagram alir generasi dan penyimpanan *digital signature membership.*

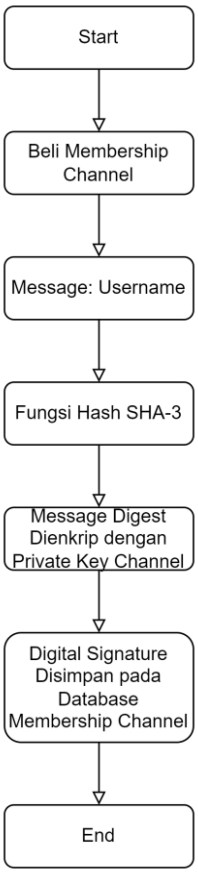

Fig. 6. Diagram Alir Generasi dan Penyimpanan *Digital Signature Membership*

Pada proses verifikasi *digital signature* untuk pengguna yang membeli *membership*, akan dibandingkan hasil tanda tangan digital dengan *message* berupa *username*. Prosesnya sama seperti sebelumnya, *username* akan di-*hash* menggunakan SHA-3, lalu didapatkan *message digest* untuk dienkrip menggunakan kunci privat *channel*. Didapatkan *digital signature* pengguna bersangkutan, lalu sistem akan mengecek apakah *digital signature* tersebut ada pada *database membership channel* atau tidak, jika ada maka pengguna tersebut memiliki benefit *membership*.

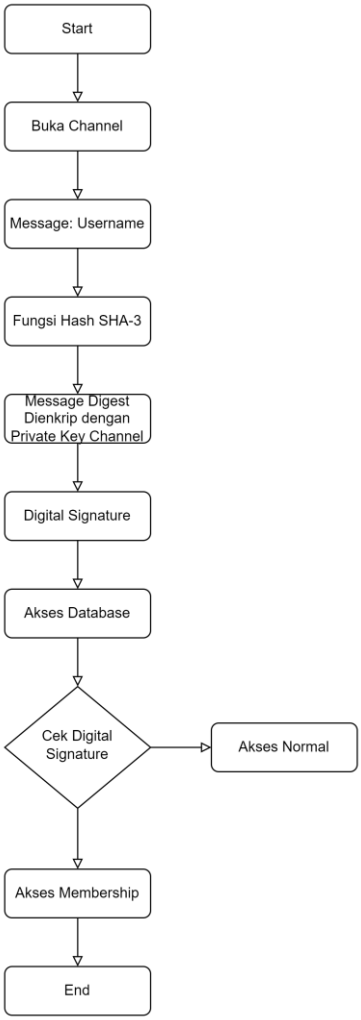

Fig. 7. Diagram Alir Verifikasi *Digital Signature Membership*

Cara lain yang dapat dilakukan untuk memverifikasi *digital signature* untuk pengguna yang membeli *membership* adalah membandingkan hasil *hash* SHA-3 menggunakan *message username* yang berupa *message digest* (h) dengan hasil dekripsi menggunakan kunci publik *channel* masing-masing *digital signature* yang tersimpan pada *database membership channel*. Namun, hal ini tidak diimplementasikan karena dinilai kurang efisien dan membutuhkan waktu yang lama dalam melakukan pengecekan masing-masing dengan proses dekripsi yang dilakukan secara berulang-ulang.

Proses pengembangan akan dilakukan dengan bahasa pemrograman Python dan berlandaskan kode dari Tugas 3 Kriptografi dan Koding: Implementasi Program Tanda Tangan Digital dengan Menggunakan Algoritma RSA dan Fungsi *Hash* SHA-3[8]. *Database* yang digunakan hanya bersifat lokal yang akan disimpan pada format berkas .txt.

A. Implementasi

Implementasi akan dilakukan menggunakan bahasa pemrograman Python. Berikut beberapa potongan program, penjelasan singkat program, dan beberapa tangkapan layar

fitur. Terdapat beberapa bagian program yang terpisah, seperti main.py untuk program utama, digitalSign.py untuk memberikan tanda tangan digital, RSA.py untuk keperluan RSA, SHA3.py untuk keperluan SHA3 dan *hash*, writeKey.py untuk menulis kunci privat dan publik *channel*. Beberapa data yang disimpan juga antar lain adalah akun dan *digital sign membership*.

Pada main.py diisi dengan segala fitur yang berhubungan dengan antarmuka pada CLI, proses *login*, dan pemilihan fitur pada sistem yaitu *channel*, profil, dan *logout*.

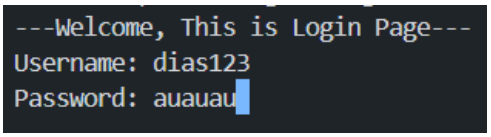

Fig. 8. *Login Page*

Login page akan meminta pengguna untuk mengisi *username* dan juga *password*. Setelah dimasukkan, sistem akan mengecek *username* dan *password* yang dimasukkan, apakah sesuai dengan data yang tersimpan pada akun.txt atau tidak. Jika sesuai maka pengguna dapat masuk ke menu utama.

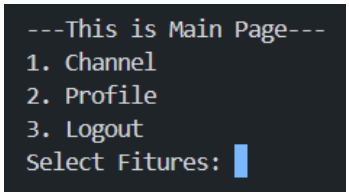

Fig. 9. *Main Page*

Pada menu utama, pengguna dapat memilih diantara 3 fitur, yaitu *channel*, *profile*, dan *logout*. Jika memilih *channel* maka akan ditampilkan konten pada *channel* tersebut. Jika memilih *profile* maka akan ditampilkan profil pengguna berupa *username*-nya. Jika *logout* maka sistem akan dimatikan.

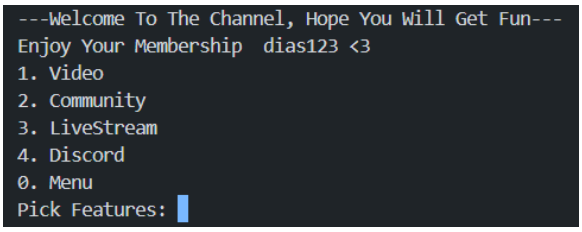

Fig. 10. *Channel Page Membership*

Pada menu *channel*, terdapat konten-konten yang bisa diakses. Tentu saja konten yang dapat diakses oleh pengguna *membership* berbeda dengan pengguna non-*membership*.

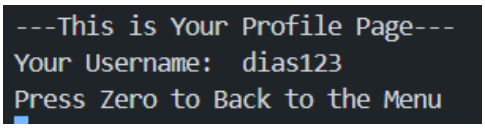

Fig. 11. *Profile Page*

Profile page akan menampilkan *username* pengguna yang masuk sekarang. Jika ingin kembali ke menu maka pengguna harus memasukkan angka 0 ke cmd.

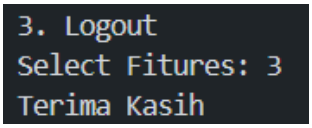

Fig. 12. *Logout*

Jika memilih *logout*, sistem secara otomatis akan dimatikan. Sistem juga menampilkan pesan berupa “Terima Kasih” kepada pengguna.

B. Pengujian

Terdapat beberapa poin pengujian yang dilakukan pada pengembangan sistem pengecekan *membership* pengguna menggunakan *digital signature*. Berikut adalah poin-poin pengujian yang dilakukan.

- Sistem dapat memvalidasi pengguna yang masuk menggunakan *username* dan *password*.
- Sistem dapat mengecek apakah pengguna tersebut memiliki fitur *membership* atau tidak.
- Sistem dapat menampilkan perbedaan akses yang didapatkan oleh pengguna *membership* dan non-*membership*

C. Hasil

Berikut hasil dari tiap-tiap poin pengujian yang sudah dilakukan. Pengujian dilakukan dengan metode *black box*, yaitu dengan mencoba kemungkinan berbagai *input* tanpa mengetahui isi kode program. Untuk poin pertama, sistem berhasil memvalidasi pengguna yang masuk menggunakan *username* dan *password*. Sistem dapat membedakan *username* dan *password* yang sesuai maupun yang tidak sesuai pada *database* akun.

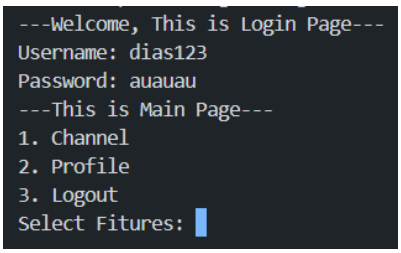

Fig. 13. Hasil Pengujian *Login* dengan Akun Sesuai

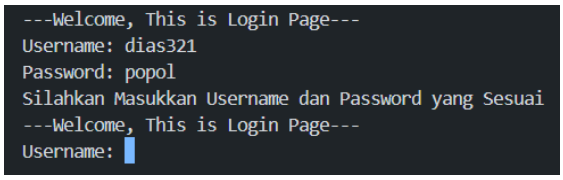

Fig. 14. Hasil Pengujian *Login* dengan Akun Salah

Untuk poin kedua pada pengujian, sistem berhasil mengecek apakah pengguna termasuk *membership* atau bukan. Sistem akan mengecek status *membership* setiap pengguna memilih fitur *channel*. Pengecekan dilakukan melalui perbandingan *digital signature* yang tersimpan pada *database membership channel* dengan hasil proses generasi *digital signature* menggunakan *message username* pengguna.

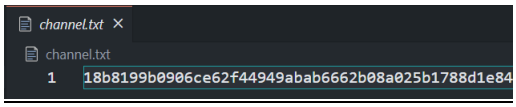

Fig. 15. *Database Membership Channel*

```
def checkMembership():
    membership = digitalSign(username)
    with open('channel.txt', 'r') as f:
        membership_list = f.readlines()
        for i in range (len(membership_list)):
            if (membership_list[i] == membership):
                return True
    return False
```

Fig. 16. Fungsi Mengecek Status Membership

Untuk poin ketiga pada pengujian, sistem berhasil menampilkan perbedaan akses yang bisa dinikmati oleh pengguna *membership* dengan pengguna non-*membership*. Dapat dilihat pada Fig. 10. Pengguna *membership* akan ditampilkan pesan tambahan berupa "Enjoy Your Membership <*username*> <3". Sedangkan untuk pengguna non-*membership*, langsung ditampilkan fitur apa saja yang bisa diakses di *channel*. Fitur tambahan lain yang hanya dapat diakses oleh pengguna *membership* adalah Discord. Terdapat pula perbedaan pada fitur Video, Community, dan LiveStream dengan adanya konten eksklusif khusus pengguna *membership*. Berikut salah satu contoh perbedaan pada fitur Video.

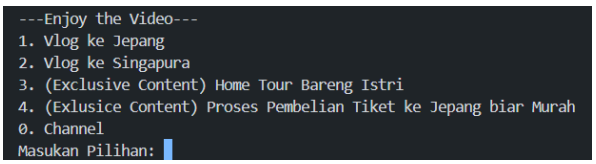

Fig. 17. Menu Video pada Pengguna *Membership*

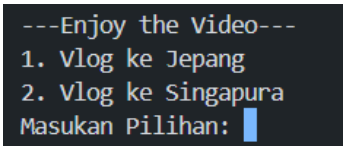

Fig. 18. Menu Video pada Pengguna Non-*Membership*

IV. KESIMPULAN DAN SARAN

Membership merupakan fitur yang disediakan oleh platform media sosial untuk para *content creator*. Fitur ini bertujuan untuk memperoleh penghasilan lebih di samping pemasukan yang lainnya. Melalui fitur ini, pengguna dan *content cretor* mendapatkan benefit masing-masing. Teruntuk pengguna, dengan adanya fitur ini, mereka dapat berhubungan langsung dengan idolanya, mendapatkan *privilege* lebih di setiap konten, dan lainnya.

Namun, penerapan fitur ini harus menerapkan proses otentikasi terlebih dahulu kepada penggunanya. Hal ini dilakukan agar tidak sembarang pengguna juga mendapatkan benefit yang sama dengan pengguna membership. Maka dari itu, dirancang sebuah sistem berlandaskan kriptografi yang akan mengotentikasi penggunanya dengan menggunakan konsep *digital* signature kombinasi hash dan kriptografi kunci publik.

Berdasarkan hasil pengujian, didapatkan hasil dan kesimpulan bahwa penggunaan *digital signature* untuk melakukan otentikasi status *membership* pada pengguna dapat dilakukan dan diterapkan dengan baik. Namun, karena implementasi ini masih dalam tahapan prototipe dengan banyaknya asumsi yang digunakan, masih banyak terdapat saran pengembangan ke depannya. Beberapa diantaranya adalah penambahan fitur pembelian *membership* dengan menerapkan prinsip kriptografi dalam menjaga transaksinya, penambahan *channel*, pengakomodasian *membership* yang *expired*, dan lainnya. Diharapkan dengan adanya rancangan pengimplementasian sistem ini, otentikasi fitur *membership* dapat lebih dikembangkan dengan baik.

SOURCE CODE LINK AT GITHUB

https://github.com/DnA2702/DigitalSignatureForMembership

REFERENCES

[1] "Apa itu Sosial Media – Universitas Pasundan," Universitas Pasundan, Mar. 2012. [Online]. Available: https://www.unpas.ac.id/apa-itu-sosial-media/. [Accessed: May 21, 2023]

[2] Dessy Rizdha Nasution, "Apa Itu Membership Program untuk Retensi Pelanggan?," Usetada.com, 2022. [Online]. Available: https://blog.usetada.com/id/mengapa-harus-ada-membership-program-simak-jawabannya-di-sini-1. [Accessed: May 21, 2023]

[3] "Kriptografi: Definisi, Sejarah, Jenis. Dan Algoritmanya," Software House & System Integrator di Malang, Indonesia, Apr. 19, 2021. [Online]. Available: https://www.sekawanmedia.co.id/blog/pengertian-kriptografi/. [Accessed: May 21, 2023]

[4] R. Munir, "Algoritma RSA Bahan kuliah II4031 Kriptografi dan Koding," 2023 [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/10-Algoritma-RSA-2023.pdf

[5] R. Munir, "Bahan kuliah II4031 Kriptografi dan Koding," 2023 [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/13-Fungsi-hash-2023.pdf

[6] R. Munir, "SHA-3 (Keccak) Bahan kuliah II4031 Kriptografi dan Koding" [Online]. Available: https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/15-SHA-3-2023. [Accessed: May 21, 2023]

[7] R. Munir, "Tanda-tangan Digital II4031 Kriptografi dan Koding" [Online]. Available:

https://informatika.stei.itb.ac.id/~rinaldi.munir/Kriptografi-dan-Koding/2022-2023/16-Tanda-tangan-digital-2023.pdf. [Accessed: May 21, 2023]

[8] "18220008_18220051_18220103 - Google Drive," Google.com, 2013. [Online]. Available: https://drive.google.com/drive/folders/10A-JCE3y7Lb3AezSJMV12y4fMzJJiIZ5. [Accessed: May 22, 2023]

[9] "ChatGPT," Openai.com, 2023. [Online]. Available: https://chat.openai.com/. [Accessed: May 22, 2023]

Bandung, 22 Mei 2023

Natanael Dias - 18220051

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.