BAB II

TINJAUAN PUSTAKA

2.1 Landasan teori

2.1.1 Kata sandi

Kata sandi merupakan kumpulan karakter atau *string* yang digunakan untuk mendapatkan akses tertentu, contohnya seperti pada akun sosial media, sistem komputer atau suatu ruangan yang dilindungi. Kata sandi saat ini menjadi salah satu cara keamanan yang selalu diterapkan untuk melindungi akses data pribadi, hal ini berguna untuk menjaga data agar tidak tersebar dan tidak disalah gunakan oleh pihak yang tidak bertanggung jawab (Chanda, 2016). Faktor kejadian di mana kata sandi gagal dalam memberikan keamanan, salah satunya adalah karena pembuatan kata sandi terlalu lemah, contohnya seperti kalimat yang terlalu pendek, mudah ditebak, dan tidak berdasarkan informasi pribadi sehingga menyebabkan pengguna tidak ingat dengan kata sandi yang sudah dibuat (Antonov & Georgieva, 2020). Solusi untuk mendapatkan kata sandi yang kuat dan aman, yaitu menggunakan delapan sampai dua belas karakter, menggunakan kombinasi huruf besar dan kecil, simbol, dan nomor serta menggunakan informasi pribadi.

2.1.2 Frasa

Frasa merupakan dua kata atau lebih yang bersifat *nonpredikatif* dan tidak melampaui fungsi misalnya "Gunung Tinggi" disebut dengan frasa karena merupakan kontruksi nonpredikatif (Yulia & Wahidah, dkk., 2019). Berikut adalah jenis-jenis dari frasa :

1. *Frase eksosentrik* merupakan frase yang komponennya tidak mempunyai prilaku sintaksi secara keseluruhan..
2. *Frase endosentrik* merupakan salah satu komponen yang memiliki prilaku sintaksi yang sama secara keseluruhan.
3. *Frase koordinatif* merupakan frase yang memiliki kelebihan pembentukan komponennya bisa lebih dari dua secara potensial dan sederajat sehingga dapat dihubungkan oleh *konjungsi koordinatif*.
4. Frase *apositif* merupakan frase yang keduanya komponennya saling berhubungan, maka urutan komponennya dapat saling bertukar.

2.1.3 Passphrase

Passphrase merupakan gabungan rangkaian kata yang digunakan sebagai kata sandi agar lebih panjang dari kata sandi biasa dengan tujuan untuk meningkatkan keamanan. Salah satu contoh *passphrase* seperti "SayaInginMakan" (Lennartsson dkk., 2019), pembuatan *passphrase* yang lebih panjang akan mendapatkan tingkat *efektivitas* dan *efesiensi* keamanan yang baik.

Pembuatan *Passphrase* panjang merupakan *depelopment* kata sandi terbaru saat ini, yang dapat meningkatkan keamanan yang lebih baik. Akan tetapi,

pembuatan *passphrase* panjang akan menyulitkan pengguna karena harus mampu mengingat selama pengguna akan memakainnya(Keith dkk., 2009).

2.1.4 Diceware

Diceware merupakan metode untuk mendapatkan sebuah kata sandi menggunakan dadu secara *random*. Metode ini bekerja dengan melemparkan dadu sebanyak tiga kali dengan rentan angka satu sampai enam sehingga akan mendapatkan angka tiga digit dadu (Buchyk dkk., 2021). Angka tiga digit tersebut kemudian digunakan untuk mencari sebuah kata dalam daftar khusus yang disebut dengan daftar kata *Diceware*. Berikut contoh kutipan singkat dari metode *Diceware* dalam Bahasa *Inggris,* jika pengguna ingin mendapatkan tiga kata frasa kata sandi, maka membutuhkan sembilaan kali lemparan dadu, contoh hasil lemparan dadu

1,2,3,2,4,5,3,4, dan 3

Tuliskan hasil lemparan dadu secara acak tersebut, kemudian cari setiap kelompok nomor dalam daftar kata *Diceware* untuk menemukan daftar kata, berikut tabel 2.1 yang menjelaskan isi code kalimat kata lemparan dadu sebanyak 3 kali

Table 2. 1 Contoh Hasil Lemparan Dadu

Angka Dadu	Kata *Diceware*
123	*Singer*
145	*Doctor*
156	*Dhani*
234	*Care*
245	*Sing*
256	*Cook*
365	*Tomato*
343	*Song*
332	*egg*

Tabel 2. 1 merupakan contoh penjelasan kamus kata dari setiap tiga lemparan dadu menjadi 3 nomor digit kode, dari tiap kode tersebut memiliki hasil kalimat kata yang nantinya akan digabungkan sehingga menjadi sebuah frasa kata sandi, setelah selesai melemparkan dadu sebanyak 9 kali dan cari isi dari kode kata tersebut

Lemparan 1 : 1 2 3 = *Singer*

Lemparan 2 : 2 4 5 = *Sing*

Lemparan 3 : 3 4 3 = *Song*

Passphrase yang didapatkan adalah ***SingerSingSong***

2.1.5 Memorability

Merupakan kemampuan pengguna yang mengacu pada kemampuan daya ingat suatu informasi dalam mengingat dan mengenang, hal ini menjadi aliran penelitian yang berfokus pada teori memori dan perilaku pengguna dalam manajemen kata sandi, oleh karena itu keamanan dalam daya ingat kata sandi menjadi sangat penting(Woods & Siponen, 2019).

2.1.6 Jaro Winkler Distance

Algoritma *Jaro Winkler Distance* merupakan metrik *string* yang digunakan dalam ilmu komputer untuk mengukur jarak dua *string*, algoritma ini bermula dari algoritma *Jaro Distance* yang ditemukan oleh Matthew A. Jaro kemudian dikembangkan oleh William E. Winkler dan thibaudeau dengan memodifikasi *Jaro Distance* untuk memberikan hasil akurasi yang lebih baik (Leonardo & Hansun, 2017). Algoritma *Jaro Winkler* apabila kedua string memiliki nilai jarak yang tinggi, maka makin mirip juga data yang akan diperoleh. Perhitungan nilai normal dari algoritma *jaro winkler distance* ialah apabila tidak ada kesamaan pada *string*, maka diberi nilai 0 dan 1 menandakan bahwa *string* memiliki kesamaan (Aritomatika dkk., 2021).

Formula perhitungan *similarity* dari *jaro winkler* terdiri dari 3 tahap :

1 Menghitung panjang *string*

2 Menemukan jumlah huruf yang sama di dua perbandingan *string*

3 Menemukan *transposisi*

algoritma *jaro* untuk menghitung *distance* antara dua string, string ke 1 (s1) dan *string* ke 2(s2) menggunakan ***Formula* 2. 1**(Aritomatika dkk., 2021)**.**

$$d_j = \frac{1}{3} * \left(\frac{m}{|s_1|} + \frac{m}{|s_2|} + \frac{m-t}{m}\right) \quad (2.1)$$

m = jumlah karakter yang sama

$|s_1|$ = panjang *string* 1

$|s_2|$ = panjang *string* 2

t = jumlah *transposisi*

Jarak teoritis dari kedua *string* yang dibandingkan dapat dianggap benar jika tidak melebihi ***Formula* 2. 2** (Aritomatika dkk., 2021)**.**

$$\left(\frac{max\ (|s_1|, |s_2|)}{2}\right) - 1 \quad (2.2)$$

Jika mengacu pada nilai yang akan dihaslikan, maka jarak maksimalnya adalah 1 yang akan menandakan adanya kesamaan *string* yang akan dibandingkan untuk mendapatkan akurasi seratus persen

Tahapan *Jaro Winkler Distance* menggunakan *prefix scale* (p) yang dapat memberikan tingkat akurasi yang lebih, dan *prefix length* (l) yang menyatakan panjang awalan karakter yang sama dari *string* yang telah dibandingkan sampai menemukan ketidak samaan apabila dilanjutkan dengan menggunakan *string* 1 dan *string* 2 yang dibandingkan, maka *Jaro Winkler distance*nya (dw) seperti pada persamaan ***Formula* 2. 3** (Tannga dkk., 2017).

$$d_w = d_j + \left(l * p\left(1 - d_j\right)\right) * 100\% \quad (2.3)$$

d_w = *Jaro Winker Distance* untuk *string* 1 dan 2

l = panjang *prefix* di awal *string* maksimalnya 4 karakter

p = *konstanta scaling factor*

2.1.7 Password Strength

Password strength merupakan metode pengukuran yang dapat mengidentifikasi apakah kata sandi yang dibuat itu mempunyai kekuatan "lemah", "sedang", "kuat" dan "unggul" sehingga membantu pengguna untuk mendapatkan kata sandi yang unggul, kebanyakan pengguna dalam pembuatan kata sandi dapat menghasilkan yang kuat dengan kombinasi kebijakan komposisi yang telah disediakan (Chowdhury, dkk., 2024) (Carnavalet & Mannan, dkk., 2014). Akan tetapi, masalah yang sering terjadi dalam pengukuran kekuatan kata sandi yang tersedia memiliki standar yang berbeda-beda.

Table 2. 2 Standar *Password Strength* Sosial Media

no	*service*	*Streng scale*	Length limits		*Charset required*
			Min	*Max*	
1	Dropbox	*Very weak, weak, So-so, Good, Great*	6	72	∅
2	Microsoft	*weak, medium, strong, best*	1	-	∅
3	Twitter	*To short, obvious, not secure enough, could be more secure, okay, perfect*	6	>1000	∅
4	Yahoo!	*Too short, weak, strong, very strong*	6	32	∅
5	eBay	*Invalid, weak, medium, strong*	6	20	Setiap 2 *charset*
6	Google	*Weak, fair, good, strong*	8	100	∅
7	Skype	*Poor, medium, good*	6	20	2 *charset* atau hanya *upper*
8	Apple	*Weak, moderate, strong*	8	32	1 *lower*, 1 *upper*, 1 digit
9	PayPal	*Weak, fair, strong*	8	20	Setiap 2 *charset*

Pada tabel 2. 2 merupakan perbedaan standar pada pengukuran tiap *service* dan memiliki tipenya masing masing.

2.1.8 Combination Formula

Combination formula atau rumus *combination* merupakan cara menghitung jumlah dengan cara yang berbeda dalam memilih objek n dari sejumlah objek r tanpa memperlihatkan urutan pembentukannya, *formula* ini biasanya digunakan untuk memilih kelompok objek tertentu sesuai dengan total objek yang tersedia pada ***Formula* 2. 4** (Warnars & Hendric, dkk., 2010)**.**

$$n\ C\ r = \frac{n!}{r!\,(n-r)!} \tag{2.4}$$

n = jumlah total objek yang tersedia.

r = jumlah objek yang akan dipilih.

$n!$ = faktorial hasil perkalian dari semua bilangan bulat positif dari 1 sampai n.

$r!$ = faktorial hasil perkalian dari semua bilangan bulat positif dari 1 sampai r.

$(n-r)!$ = faktorial hasil perkalian dari semua bilangan bulat dari 1 sampai (n – r).

2.2 Penelitian terkait dan keterbaruan penelitian

Penelitian ini mengacu kepada penelitian-penelitian sebelumnya dengan tujuan untuk mempermudah proses pemgumpulan dan pengolahan data. Selain itu, berguna untuk mempermudah mendapatkan metode analisis yang akan digunakan. Penelitian ini dijadikan pedoman dan referensi dari penelitian yang dapat dilihat pada tabel 2. 4 sebagai berikut.

Table 2. 3 Penelitian Terkait Dan Kebaruan Penelitian

No	Peneliti	Judul	Tahun	Metode	Hasil penelitian
1.	Serhii Buchky, Nataliia Lukova-Chuiko, Serhii Toliupa, Vitalii Piatyhor, dan Oleksandr Buchky	*Diceware Password Generation Algorithm Modification based on Pseudo-Random Sequences*	2021	*Diceware & Pseudo-Random*	Hasil modifikasi lebih tahan terhadap serangan. akan tetapi, memiliki kesulitan dalam mengingat kata sandi.
2.	Avirup Mukherjee, Kousshik Murali, Shivam Kumar Jha, Niloy Ganguly, Rahul Chatterjee, dan Mainack Mondal	*MASCARA: Systematically Generating Memorable and Secure PassPhrases*	2023	*MASCARA*	Berhasil membuat metode baru yang dapat membuat generator kata sandi yang mudah di ingat dan tetap aman, dalam prosesnya pun memberikan pengukuran daya ingat dan kemampuan menebak kata sandi meskipun hasilnya diutamakan untuk keamanan,
3.	Abejide Ade-Ibijola & Blessing Ogbuokiri	*Syntactic Generation of Memorable Password*	2019	*Context-Free Grammer (CFG)*	Hasil kata sandi yang didapatkan mudah diingat dengan menggunakan *Context-Free Grammer* (CFG) dengan *antonasi semantic PassGen*. Sehingga dapat menghasilkan jutaan set kata sandi yang berbeda beda.

4.	Alessia Michela Di Campi, Luccio Flaminia dan Focardi Riccardo	*Password guessing: learn the nature of passwords by studying the human behavior*	2021	*Behavioral Analysis*	Menunjukkan bahwa pengguna sebenarnya sudah mengetahui cara untuk mendapatkan kata sandi yang kuat, tetapi mereka tidak melakukan cara tersebut.
5.	Taku Sugai, Toshihiro Ohigashi, Yoshio dan Kakizaki, Akira Kanaoka	*Password Strength Measurement without Password Disclosure*	2019	*Sistem Usability Scale*	Metode pengukuran kekuatan kata sandi ini telah di survei dan telah dipastikan bahwa pengukuran kekuatan kata sandi sudah dilakukan sebagai perlindungan terhadap ancaman. Hasilnya percobaan pengguna pun menunjukkan bahwa metodenya tidak mengurangi kegunaan dan sudah baik. Evaluasi kinerja menunjukkan bahwa parameter seperti rentang angka acak dan ukuran database memiliki pengaruh yang sangat terhadap waktu pemrosesan.
6.	Nikola Vaptsarov Naval Academy, Vaarna, Bulgaria	*Security Analysis Of Passphrasess*	2020	*Diceware dan PassPhrase*	Penggunaan metode *Diceware* tidak lagi efektif dan harus digunakan dengan hati-hati, Untuk meningkatkan keamanan *Diceware* Panjang frasa sandinya minimal 8 untuk memenuhi kunci rahasia 128 bit.
7.	Naomi Woods dan Mikko Siponen	*Improving password memorability, while not inconveniencing the user*	2019	*Password Security dan Password Security Behavior*	Hasil penelitiannya menguji pada peningkatan jumlah waktu verfikasi kata sandi dapat meningkatkan daya ingat kata sandi, verfikasi dapat membuat kata sandi menjadi lebih mudah diingat namun secara bersamaan meningkatkan ketidaknyamanan pengguna karena dapat mengurangi perilaku katasandi yang tidak aman dan masalah keamanannya terkait dengan perilaku.
8,	Seok Jun Kim dan Byung Mun Lee	*Multi-class Classification prediction model for password strength based on deep learning*	2023	*Deep Learning*	Hasil menunjukkan bahwa model yang dilatih sangat *efektif* sehingga metode yang diusulkan memberikan Solusi yang lebih baik untuk evaluasi kekuatan kata sandi.

9.	Jaryn Shen, Kim-Kwang Raymond Choo, dan Qingkai Zeng	*Multi-item Passphrases: A Self-adaptive Approach Against Offline Guessing Attacks*	2019	*Authentication PassPhrases*	Hasilnya tidak realistis karena sistem berbasis kata sandi akan menghilang dalam waktu dekat, maka disarankan untuk merancang sistem autentikasi berbasis kata sandi yang kuat untuk mengurangi keterbatasan katasandi yang mudah diretas.
10.	Bongkeum Jeong, Alexander Vallat, Chris Csikszentmihalyi, Junwu Park, dan Dulce Pacheco	*Mementokey: Keeping Password in Mind*	2019	*Password Manager*	Dapat menghasilkan kata sandi dengan pasangan kata yang dihasilkan secara acak dengan mengikuti aturan kompleksitas kata sandi biasa. Studi validasinya mendukung untuk meningkatkan pekerjaan dalam menyajikan kata acak untuk dihafal dengan memperkenalkan pasangan kata dan asosiasi visual.

2.3 Matriks penelitian

Matriks penelitian menjelaksan mengenai penelitian sebelumnya yang memiliki hubungan dengan penelitian ini dan perbedaan pada setiap penelitian.

Table 2. 4 Matriks Penelitan

no	Peneliti	Parameter				Sistem Operasi		
		Diceware Algoritma	Modifikasi	Pengukuran Memorability	*Password Strength*	Mac OS	Windows	Linux
1.	Serhii Buchky, Nataliia Lukova-Chuiko, Serhii Toliupa, Vitalii Piatyhor, dan Oleksandr Buchky., 2021	✓	✓	✗	✗	✗	✓	✗
2.	Avirup Mukherjee, Kousshik Murali, Shivam Kumar Jha, Niloy Ganguly, Rahul Chatterjee, dan Mainack Mondal., 2022	✗	✓	✓	✗	✗	✗	✓
3.	Abejide Ade-Ibijola & Blessing Ogbuokiri., 2020	✗	✓	✓	✗	✗	✗	✗
4.	Alessia Michela Di Campi, Luccio Flaminia dan Focardi Riccardo., 2021	✗	✗	✓	✗	✗	✗	✗

5.	Taku Sugai, Toshihiro Ohigashi, Yoshio dan Kakizaki, Akira Kanaoka., 2019	✗	✗	✓	✗	✗	✗	✗
6.	Nikola Vaptsarov Naval Academy, Vaarna, dan Bulgaria., 2020	✓	✗	✗	✓	✗	✗	✗
7	Naomi Woods dan Mikko Siponen., 2019	✗	✗	✓	✓	✗	✗	✗
8	Seok Jun Kim dan Byung Mun Lee., 2023	✗	✓	✗	✓	✗	✗	✗
9	Jaryn Shen, Kim-Kwang Raymond Choo, dan Qingkai Zeng., 2019	✗	✓	✗	✗	✗	✗	✗
10	Bongkeum Jeong, Alexander Vallat, Chris Csikszentmihalyi, Junwu Park, dan Dulce Pacheco., 2019	✗	✓	✗	✓	✗	✓	✗
11	Marco Antonio Carnut dan Evandro Curvelo Hora., 2005	✓	✓	✓	✗	✗	✓	✗
12	Penelitian yang dilakukan saat ini,. 2024	✓	✓	✓	✓	✗	✓	✗

2.4 Relevansi penelitian

Relevansi penelitian digunakan untuk menilai untuk menunjukan sejauh mana penelitian ini relevan terhadap topik yang sedang dipelajari. Relevansi penelitian yang berjudul Implementasi *Jaro Winkler Distance* dalam pengukuran *Memorability* dan *Password Strength* pada Algoritma *Diceware* pada table 2. 5

Table 2. 5 Relevansi Perbandingan

Judul	*Diceware Password Generator Algorithm Modification Based On Peseudo-Random Sequence*	Implementasi *Jaro Winkler Distance* Dalam Pengukuran *Memorability* dan *Password Strength* pada Algoritma *Diceware*
Peneliti	(Buchky dkk., 2021)	Penelitian yang dilakukan saat ini
Masalah penelitian	Kelemahan perlindungan terhadap pembuatan kata sandi tanpa perhatian yang cukup	Belum diketahui tingkat daya ingat memorability dan password strength terhadapt kata sandi yang telah dihasilkan
Objek penelitian	Pembuatan kata sandi	Pembuatan kata sandi
Algoritma / metode	*Algorithm Diceware & pseudo-random sequence*	*Algorithm Diceware ,Jaro Winkler Distance & password strength*