Pengembangan Sistem Manajemen Naskah Soal dengan Keamanan *Pre-Hash Coding*

Development of Question Sheet Management System with Pre-Hash Coding Security

Prajanto Wahyu Adi[1], Retno Kusumaningrum[2]
[1,2] Departemen Informatika, Universitas Diponegoro
E-mail: [1]prajanto@live.undip.ac.id, [2]retno@live.undip.ac.id

Abstrak

Sistem pengelolahan dokumen secara elektronik sudah menjadi salah satu kebutuhan penting dalam institusi pendidikan khususnya dalam pengelolaan naskah soal. Masalah utama dalam pengelolaan naskah elektronik adalah adanya berbagai format yang digunakan serta kekhawatiran terhadap tingkat keamanan akun. Sistem keamanan akun dengan menggunakan *password* yang sederhana akan mudah diretas sedangkan penggunaan sistem yang kompleks akan mempersulit pengguna. Departemen Informatika Universitas Diponegoro mengembangkan sistem manajemen naskah soal yang mampu menghasilkan format naskah soal sesuai dengan standar tunggal serta memiliki sistem keamanan yang sederhana namun kuat melalui sistem pre-*hash* coding dengan nilai unik pengguna melalui dua skema. Pengujian pertama yang dilakukan berhasil membuktikan kemampuan sistem dalam menghasilkan naskah soal sesuai dengan standar tunggal. Percobaan kedua dilakukan untuk menguji tingkat keamanan terhadap nilai *hash* dari sampel *password* melalui uji *brute-force* menggunakan sistem *Hashcat*. Sistem yang diusulkan mampu menggagalkan peretasan sebesar 40% pada karakter alfanumerik pada skema pertama dengan operator *bitwise xor* sedangkan pada skema kedua dengan operator penjumlahan mampu menggagalkan seluruh peretasan yang dilakukan. Sistem yang diusulkan mampu memenuhi kebutuhan pengguna terhadap *password* yang sederhana namun kuat.

Kata kunci: dokumen elektronik, sistem keamanan , *pre-hash coding, Hashcat*

Abstract

The electronic document management system has become one of the important needs in educational institutions, especially in the management of question sheet. The main problem in the management of electronic manuscripts is the various formats used and concerns about the level of account security. Account security systems using simple passwords will be easy to hack while using complex systems will make it difficult for users. The Department of Informatics, Diponegoro University, developed a question sheet management system that is able to produce a question script format according to a single standard and has a simple but strong security system through a pre-hash coding system with unique user values. The first test that was carried out succeeded in proving the system's ability to produce question texts according to a single standard. The second experiment was conducted to test the security level of the hash value of the sample password through brute-force testing using the Hashcat system. The proposed system is able to thwart hacks by 40% on alphanumeric characters in the first scheme with the bitwise xor operator while in the second scheme with the addition operator it is able to thwart all hacks carried out. The proposed system is able to meet user needs for simple but strong passwords.

Keywords: electronic document, security system, pre-hash coding, Hashcat

1. PENDAHULUAN

Selama satu dekade terakhir otomatisasi alur kerja telah banyak digunakan pada

organisasi besar seperti organisasi komersial maupun organisasi ilmiah untuk meningkatkan kualitas kerja dan mengurangi waktu melalui sistem pengelolaan dokumen elektronik (EDMS) yang dapat melakukan otomatisasi dan integrasi seluruh proses di dalam organisasi, mulai dari pekerjaan individu pada departemen hingga proses analisis data [1]. EDMS juga dimanfaatkan untuk menyimpan dokumen secara aman dan meningkatkan proses bisnis [2]. Sistem tersebut menjadi kebutuhan penting dalam organisasi ilmiah termasuk institusi pendidikan yang banyak bekerja dengan dokumen seperti dokumen akreditasi, notulen kegiatan, administrasi dosen, karyawan dan mahasiswa, hingga dokumen naskah soal. Saat ini Departemen Informatika Universitas Diponegoro menerapkan pengelolaan naskah soal yang belum terintegrasi. Setiap dosen mengusulkan soal kepada koordinator mata kuliah lalu usulan soal tersebut diolah menggunakan aplikasi pengolah dokumen dan selanjutnya diterima oleh mahasiswa ketika ujian berlangsung. Hal ini mengakibatkan munculnya beberapa format naskah soal yang tidak seragam, selain itu proses validasi naskah soal oleh Gugus Penjamin Mutu (GPM) juga dilakukan secara terpisah. Oleh karena itu perlu dikembangkan sebuah sistem manajemen naskah soal ujian yang mampu memfasilitasi dosen, koordinator mata kuliah, GPM dan departemen dalam satu sistem.

Beberapa studi terkait dengan penerapan EDMS di Indonesia antara lain: aplikasi pengelolaan data elektronik yang dikembangkan oleh [3] dan sistem e-legalisir pada fakultas ilmu komputer Universitas Mercubuana [4] yang dikembangkan dengan menggunakan metode *Extreme Programming (XP)*. Sistem yang dikembangkan hanya diuji coba melalui pengujian *black-box* terhadap seluruh fungsi antarmuka tanpa adanya penjelasan tentang sistem keamanan yang digunakan dalam menjaga dokumen dari akses pihak yang tidak berwenang. Aplikasi sejenis juga dikembangkan oleh [5] dengan menerapkan fungsi *hash* MD5 untuk mengamankan *session* dan *password* pada sistem. Selain penggunaan fungsi *hash* yang bersifat satu arah, penerapan algoritma enkripsi yang bersifat dua arah juga diterapkan pada [6] dengan menggunakan pendekatan *Web Development Life Cycle (WDLC)* untuk mengelola naskah dinas dengan menerapkan sistem keamanan *Secure Hash Algorithm (SHA)* dan *Rivest Shamir Adleman (RSA)* pada tandatangan digital serta *Advanced Encryption Standard (AES)* untuk enkripsi dokumen.

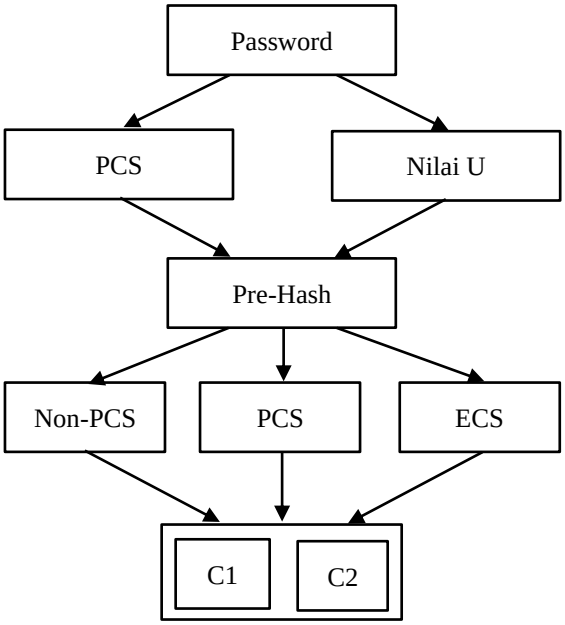
Namun demikian penggunaan sistem enkripsi khususnya fungsi *hash* memiliki kekurangan terutama ketika diterapkan pada data dengan panjang string terbatas seperti *password* atau kata kunci yang mengharuskan pengguna untuk mengingatnya. Sebuah *password* harus memiliki nilai entropi yang tinggi dengan menyertakan huruf kapital, angka, dan simbol agar tidak mudah ditebak. Tetapi sebuah *password* harus juga mudah diingat sehingga pengguna tidak perlu menyimpan *password* pada catatan khusus yang kemungkinan tidak aman. Hal serupa juga terjadi pada data yang bersifat mutlak seperti nama dan tanggal lahir, dimana data tersebut berupa karakter alfanumerik yang sangat rentan ditebak melalui serangan *bruteforce* seperti *Hashcat* [7], [8] yang mudah didapatkan di internet. Banyak penelitian yang menjelaskan tentang bahaya dari sistem pemecah *hash* seperti *Hashcat* karena dapat diaplikasikan secara luas seperti pada web open source [9], aplikasi pesan instan [10], perangkat ponsel pintar [11], hingga pada akun mata uang kriptografi [12] yang dapat dilakukan secara terdistribusi [13]. Penelitian tentang sistem peretas *password* cerdas juga terus dikembangkan [14] yang didukung oleh kecepatan perangkat komputer yang akan terus meningkat. Di sini lain penelitian tentang peningkatan keamanan *password* juga terus dilakukan untuk mendapatkan *password* dengan nilai entropi tinggi tetapi mudah diingat [15], [16].

Sistem *bruteforce* seperti *Hashcat* menggunakan *Printable Character Set (PCS)* sebagai target utama dalam menemukan nilai *hash* dari sebuah data dengan nilai entropi rendah secara efisien, bahkan dapat dilakukan dengan hanya menggunakan perangkat komputer desktop atau laptop yang banyak dimiliki oleh masyarakat saat ini. Oleh karena itu, artikel ini mengusulkan penggunaan sistem *pre-hash coding* dengan memanfaatkan *Unique ID* untuk melakukan pengkodean terhadap data sebelum dilakukan proses *hash* pada database. Tujuan dari penggunaan sistem tersebut adalah untuk melakukan pengkodean terhadap data agar berada diluar nilai PCS yang sehingga dapat meningkatkan nilai entropi dapat mengurangi tingkat keberhasilan sistem *bruteforce* pada *Hashcat* secara signifikan tanpa membebani komputasi.

2. METODE PENELITIAN

2.1 Sistem Pre-Hash Coding

Data yang telah dimasukkan akan di simpan ke database MySQL dengan standar enkripsi *password* MD5. Untuk meningkatkan keamanan data yang tersimpan pada database khususnya data *password*, penulis mengusulkan penggunaan sistem pengkodean *hash* dengan nilai *Unique* ID (nilai unik) sebelum data disimpan pada database seperti yang ditampilkan pada Gambar 1.



Gambar 1 Sistem Pre-*Hash* yang Diusulkan

Pengkodean dilakukan untuk memetakan PCS sebanyak 95 nilai menjadi 256 nilai pada *Extended Character Set* (ECS) ASCII 8-bit sebagai berikut:

1. Penentuan nilai bilangan awal (*seed*) *s* yang dapat diambil dari nilai unik setiap pengguna
2. Pembacaan panjang karakter *n* dari data / plainteks *P*
3. Pembangkitan kode unik *U* sepanjang *n*
4. Pengkodean data untuk mendapatkan nilai pre-*hash* code *C* dengan 2 alternatif sebagai berikut:

$$C1_n = P_n \oplus U_n \quad (1)$$

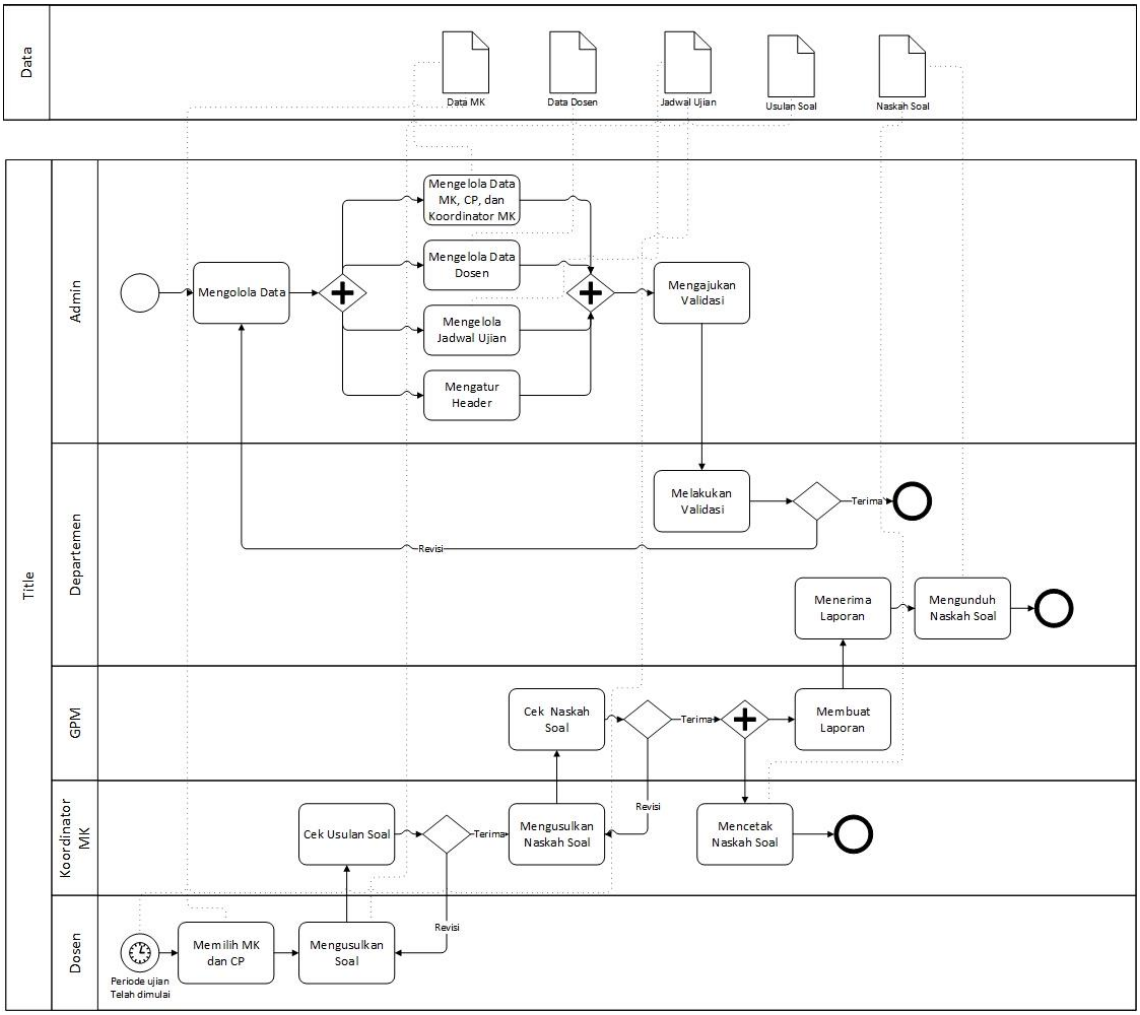
$$C2_n = P_n + U_n \quad (2)$$

Sistem pengkodean *hash* pada alternatif pertama dan kedua dilakukan melalui operator *bitwise xor* dan operator matematika penjumlahan secara berturut-turut terhadap nilai unik *U*. Kedua sistem pengkodean tersebut memiliki orde rendah sehingga mampu memetakan nilai ke rentang nilai maksimal 8-bit tanpa membebani proses komputasi penyimpanan data secara signifikan. Data yang telah dikodekan selanjutnya akan disimpan pada database yang terenkripsi oleh MD5. Pengujian tingkat keamanan akan dilakukan dengan menggunakan uji *brute-force* dengan *Hashcat* terhadap nilai P dan C lalu membandingkan hasil yang diperoleh untuk mendapatkan tingkat keberhasilan dari model yang diusulkan. Selain itu akan dilakukan juga pengujian terhadap perbandingan kecepatan proses dari kedua variabel tersebut untuk memastikan bahwa model yang diusulkan tidak membebani sistem secara signifikan.

2.2 Perancangan Sistem

Penelitian ini diawali dengan perancangan bisnis proses yang bertujuan untuk mendapatkan seluruh kebutuhan yang akan digunakan dalam sistem. Penggunaan bisnis proses

terbukti mampu menggambarkan keseluruhan seluruh sistem secara baik [17] dan dapat dikembangkan untuk mengetahui celah keamanan dari sebuah sistem [18]. Pada penelitian ini, perancangan bisnis proses dimulai dengan mengusulkan rancangan awal, menyampaikan hasil rancangan, dan perbaikan secara iteratif hingga diperoleh rancangan akhir dari bisnis proses seperti yang terlihat pada Gambar 2.



Gambar 2 Rancangan Bisnis Proses Sistem Manajemen Naskah Soal

Gambar di atas menjelaskan tentang alur dari sistem yang dirancangan sebagai berikut:

1. Admin memasukkan dan mengelola data berikut:
    a. Data dosen yang terdiri dari Nama, Nomor Induk Pegawai (NIP), dan *password*
    b. Data Mata Kuliah (MK) yang meliputi Capaian Pembelajaran (CP) dan Koordinator MK yang merupakan perwakilan dosen pengampu dari kelompok pengajar (*team teaching*)
    c. Header Soal yang akan digunakan sebagai kop naskah soal
    d. Jadwal ujian dari seluruh mata kuliah pada semester berjalan
2. Admin mengusulkan setiap perubahan atau masukkan data baru kepada departemen
3. Departemen melakukan validasi terhadap ajuan data dari admin
    - Jika ajuan ditolak, maka admin harus melakukan perbaikan ajuan sesuai dengan catatan yang diberikan oleh departemen
    - Jika ajuan diterima, maka data akan valid dan tidak dapat dirubah

4. Ketika sesi ujian dimulai, Dosen dapat mengajukan usulan soal kepada koordinator sesuai dengan MK dan CP yang dipilih
5. Koordinator melakukan validasi terhadap usulan soal dari dosen yang meliputi tingkat kesulitan soal dan kesesuaian butir soal dengan CP
    - Jika ajuan ditolak, maka dosen pengusul harus melakukan perbaikan ajuan sesuai dengan catatan yang diberikan oleh koordinator
    - Jika ajuan diterima, maka soal akan valid dan dapat diajukan di dalam naskah soal
6. Koordinator melakukan pengajuan naskah soal yang berisi soal-soal yang telah valid kepada GPM
7. GPM melakukan validasi terhadap usulan naskah soal dari koordinator yang meliputi data kelompok kelas, sifat ujian, durasi, ruang, dan petunjuk pengerjaan soal
    - Jika ajuan ditolak, maka koordinator harus melakukan perbaikan ajuan sesuai dengan komentar yang diberikan oleh GPM
    - Jika ajuan diterima, maka naskah soal valid dan dapat digunakan untuk dicetak atau diunggah ke dalam LMS
8. Ketika sesi ujian telah selesai, departemen dapat melihat laporan dari ujian yang telah berlangsung serta dapat mengunduh seluruh soal yang digunakan dalam ujian.

3. HASIL DAN PEMBAHASAN

Bab ini membahas tentang pengujian terhadap sistem keamanan yang diusulkan serta implementasi website yang telah dirancang sesuai dengan bisnis proses.

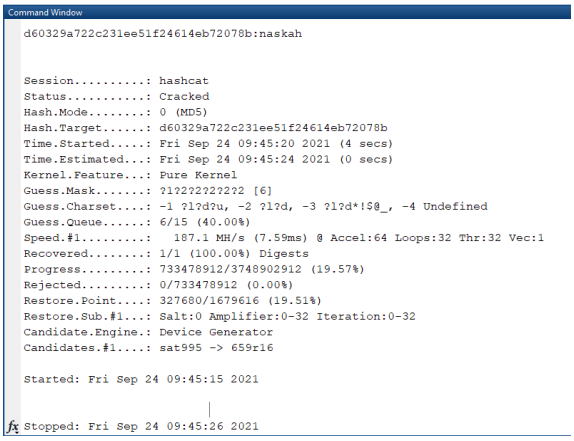
3.1 Pengujian Sistem Keamanan

Pada kasus peretasan data, umumnya peretas menyalin database maupun aliran data yang terenkripsi lalu melakukan peretasan pada perangkat lokal. Hal tersebut dilakukan karena saat ini hampir tidak mungkin untuk melakukan peretasan langsung kepada sistem secara daring yang telah terlindungi secara ketat antara lain dengan firewall dan captcha.

Penelitian ini dilakukan dengan perangkat laptop dengan spesifikasi berikut:

- Tipe : Lenovo V15-IIL
- Prosesor : Core i3-1005G1 @1,2 ~ 3,4 GHz
- VGA : Intel UHD Graphic @2GB Shared Memory
- RAM : 4GB DDR4 @2667MHz
- Sistem Operasi : Windows 10 Home 20H2

Pengujian pertama dilakukan dengan menggunakan *password* sederhana yang terdiri dari 5-6 karakter alfanumerik yang banyak digunakan karena mudah diingat oleh pengguna. Nilai *hash* yang tersimpan pada database akan dibandingkan dengan hasil pembangkitan secara *bruteforce* menggunakan *Hashcat* dengan character set alfanumerik. Hasil pengujian pada data terakhir dapat dilihat pada Gambar 3.



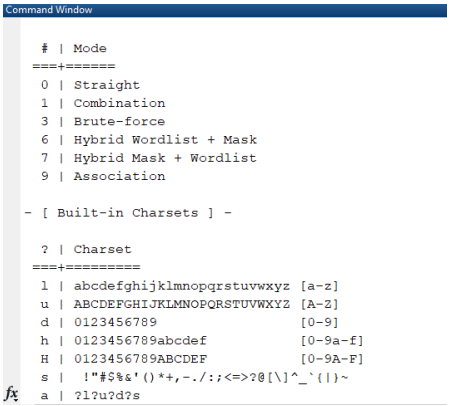
Gambar 3 Proses pengujian pada data terakhir dengan *Hashcat*

Dapat dilihat bahwa perangkat laptop kelas menengah dapat menyelesaikan proses peretasan nilai *hash* dari *password* dengan panjang 7 karakter hanya dalam waktu maksimal 11 detik. Hasil keseluruhan pengujian dapat dilihat pada Tabel 1.

Tabel 1 Hasil Pengujian dengan karakter Alfanumerik 5 - 6 Karakter

Password	**Nilai *Hash***	**Status**	**Waktu (detik)**
kunci	fe6b5f11f069a561c511bb171471c9ae	Cracked	6,54
bunda	55b0c86ed75326a42b7a48c3fbf67baf	Cracked	6,09
13mei	91021ce71ee4dec535a8da42f0b13eff	Cracked	6,02
undip	57613548e7d0a0aef7f373cb50e50994	Cracked	5,62
masuk	f3770595e0cb4d9a988bd5da98d2187d	Cracked	5,91
jagoan	52a0e427e7f45db5027c2b83df67b363	Cracked	6,62
220785	b6d3cfa8a95de93749533230b9dc5c63	Cracked	7,57
9maret	8b497f4f2f8e0921c3ace636752a90c5	Cracked	8,68
41983	560ee3fc0be699ce313ef275c25e1f6c	Cracked	11,53
naskah	d60329a722c231ee51f24614eb72078b	Cracked	10,91

Tabel di atas menunjukkan bahwa *password* yang umum digunakan dengan panjang karakter yang pendek dapat dengan mudah diretas meskipun telah terenkripsi dengan algoritma *hash*. Rata-rata panjang *password* yang digunakan oleh pengguna adalah sepanjang 5 karakter dimana sekitar 60% hanya menggunakan karakter huruf. Dengan kondisi tersebut rata-rata waktu yang diperlukan untuk meretas nilai *hash* adalah sekitar 7,5 detik, hal ini tentunya sangat membahayakan kerahasiaan data yang tersimpan. Solusi yang umum dipakai pada sistem keamanan adalah mewajibkan pengguna untuk menyertakan angka dan simbol dalam *password* yang akan digunakan yang diharapkan dapat meningkatkan keamanan *password* dengan jumlah karakter yang pendek.



Gambar 4 Character Set pada *Hashcat*

Untuk menguji hal tersebut, pada percobaan kedua dilakukan pengujian dengan menggunakan pembangkitan *password* yang memiliki kombinasi alfanumerik dan simbol.

Pengujian dilakukan dua kali dengan character set normal (**`l`**, **`u`**, dan **`d`**) dan set luas (**`a`**) seperti yang ditampilkan pada gambar 4.

Tabel 2 Hasil Pengujian dengan kombinasi Alfanumerik dan Simbol

Password	**Nilai *Hash***	***Charset***		
		`l, u, d`	**`a`**	
		Status	**Status**	**Waktu (detik)**
!}/*C	a78d2c3924c586d643db3b920b5903e2	Exhausted	Cracked	24,99
2N@zw	40e5b60ec22ac8f05cf611e8c6d19020	Exhausted	Cracked	15,66
%f9HT	6ec0046e363f8867d44395c58c123a43	Exhausted	Cracked	7,94
yG}<b	e7a2aff21ff9a00103eed118be4ab47d	Exhausted	Cracked	10,63
_Sb_0	0676e182b13d26262926905411d9f9a8	Exhausted	Cracked	5,37
,~0#U	60fca7cab6f75d9efda1a8cd35078c28	Exhausted	Cracked	38,78
s_2CK	556e5160ae19253c2bff32b39971bff3	Exhausted	Cracked	4,50
}.q]C	c7d131d243989b8c8ffb7c7688e890c5	Exhausted	Cracked	27,04
2HM+X	efad4b6c1c43c89b8e5a72a4ce303ea6	Exhausted	Cracked	44,90
5DW7;	8612d26616dc512cd0e9f92f2481e1af	Exhausted	Cracked	41,70

Tabel 2 menunjukkan bahwa dengan menyertakan kombinasi simbol pada *password* dapat menghindari peretasan nilai *hash* pada percobaan dengan *charset* normal. Seluruh percobaan yang dilakukan mengalami kegagalan karena karakter simbol berada di luar *charset* normal. Namun pada percobaan dengan menggunakan *charset* yang lebih luas dengan menyertakan simbol diperoleh tingkat keberhasilan 100%. Seluruh *password* yang diuji berhasil diretas dengan waktu paling cepat sebesar 5,37 detik, waktu paling lambat adalah 44,9 detik, dan waktu rata-rata sebesar 22,15 detik. Penggunaan *charset* yang lebih luas memang memerlukan sumber daya yang besar serta waktu komputasi yang lebih lambat hingga 3 kali lipat jika dibandingkan pada percobaan pertama. Namun, demikian rata-rata waktu yang dibutuhkan untuk meretas nilai *hash* dari *password* pada percobaan ke dua termasuk sangat singkat. Hal ini membuktikan bahwa penggunaan kombinasi huruf dan angka tidak menjamin keamanan data yang ada.

Percobaan terakhir dilakukan dengan menggunakan metode yang diusulkan yakni pre-*hash* coding. Pada alternatif pertama dilakukan pengujian dengan persamaan (1) diperoleh hasil sebagai berikut:

Tabel 3 Hasil Pengujian dengan Pre-*Hash* C1

Password	**Nilai *Hash***	**Status**	**Waktu (detik)**
kunci	68fc0bbe17622f9bd96a4bcc88f1dd9c	Cracked	10,14
bunda	5ec4ca73f1f8914a1c1eb8c5b4dd45a3	Exhausted	50,54
13mei	b0aa1e1ff694b6ddeb6dbfb50b0c2501	Cracked	6,18
undip	0d7a674fbc4fc64364dcf6cf14485391	Exhausted	49,74
masuk	de79b16411e90f85e1ff0fab03f61213	Cracked	36,96
US$10	f86716639dcf06c2802a70a342a9db4a	Exhausted	49,83
AIK21	4730968e41f46a29723643670e969386	Exhausted	49,43
i-l-u	1e81d0ea880f922f6b5b62cbde91862a	Exhausted	49,29
20/21	873045db326e17bc74ea65e6d176f97a	Cracked	36,87
10_06	58ef3dd65b51fe27a0df1cc1bd73c33b	Exhausted	49,79

Tabel 3 menunjukkan sampel percobaan dengan menggunakan 10 sampel *password* yang terdiri dari 5 sampel A dengan karakter alfanumerik dan 5 sampel B dengan menyertakan simbol. Dari 10 sampel yang digunakan terdapat 6 sampel yang gagal diretas dan 4 sampel yang berhasil diretas. Sampel A tingkat peretasan sebesar 60% sedangkan pada sampel B tingkat peretasan jauh lebih rendah yakni sebesar 20%. Hal ini disebabkan karena 66% karakter simbol berada pada rentang 6-bit sedangkan pada karakter alfanumerik hanya 15% yang berada pada rentang tersebut. Penggunaan operasi *bitwise xor* dengan nilai unik akan lebih banyak memetakan nilai 6-bit ke nilai 8-bit. Penggunaan sistem *pre-hash* dengan operator *bitwise xor* akan bekerja lebih baik untuk *password* yang menyertakan karakter simbol. Pada alternatif ke dua dilakukan pengujian dengan menggunakan operator matematika penjumlahan dengan persamaan (2) dan diperoleh hasil sebagai berikut:

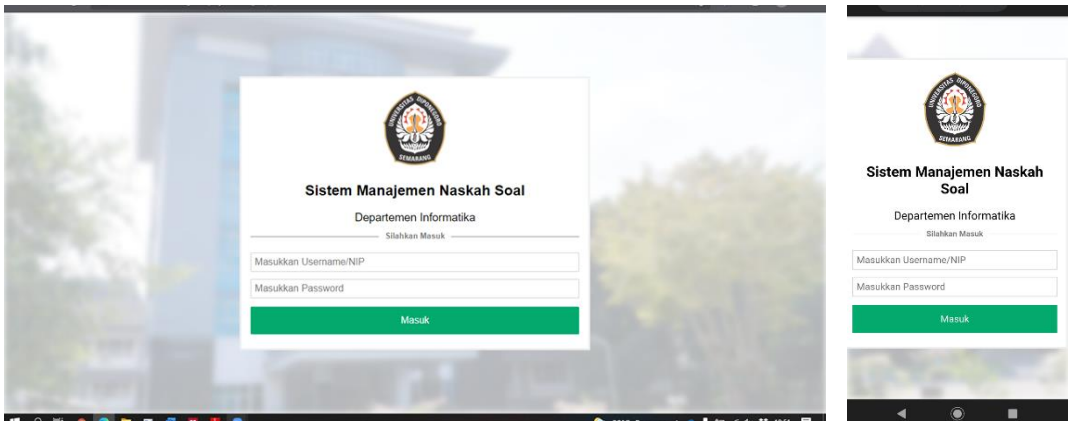
Tabel 4 Hasil Pengujian dengan Pre-*Hash* C2

Password	**Nilai *Hash***	**Status**	**Waktu (detik)**
kunci	7500572b012055a9ca14863c42ed07de	Exhausted	51,51
bunda	2205bbebf5c3a0faafd53723393e9f47	Exhausted	51,89
13mei	a81e99095ee29888aa47b16258848f52	Exhausted	51,24
undip	92fdf1a2574951ba26cc0d777ce36551	Exhausted	53,10
masuk	609f8749e759717b257c63fa39293e81	Exhausted	51,21
US$10	a8dcebe8fb548704d68a50d2684cce7a	Cracked	12,60
AIK21	1e2cec327ba22a335ff1872a89b2c437	Exhausted	49,81
i-l-u	5bdd75d840f0d7082be38b43328890e8	Exhausted	48,83
20/21	ae33cb4957e9bdfd270be16a67e6a19c	Cracked	15,58
10_06	89dfe583bf97a7379837efa8ec306240	Cracked	20,59

Tabel 4 menunjukkan hasil yang berbanding terbalik terhadap alternatif 1. Seluruh *password* pada sampel A gagal diretas sedangkan pada sampel B tingkat peretasan adalah sebesar 60%. Hal ini sesuai dengan cara kerja dari operator penjumlahan terhadap nilai unik dimana nilai yang berada pada rentang 6-bit akan lebih banyak dipetakan ke rentang 7-bit yang masih berada pada PCS sedangkan nilai pada rentang 7-bit akan lebih banyak dipetakan ke 8-bit yang berada di luar PCS. Sebanyak 85% karakter alfanumerik berada pada rentang 7-bit sedangkan pada karakter simbol hanya 33% nilai yang berada pada rentang tersebut. Penggunaan operator penjumlahan akan bekerja lebih optimal untuk *password* dengan karakter alfanumerik. Karakter alfanumerik lebih sesuai dengan kebutuhan sistem yang dikembangkan karena pengguna menginginkan *password* yang pendek dan mudah diingat dan memiliki keamanan yang baik.

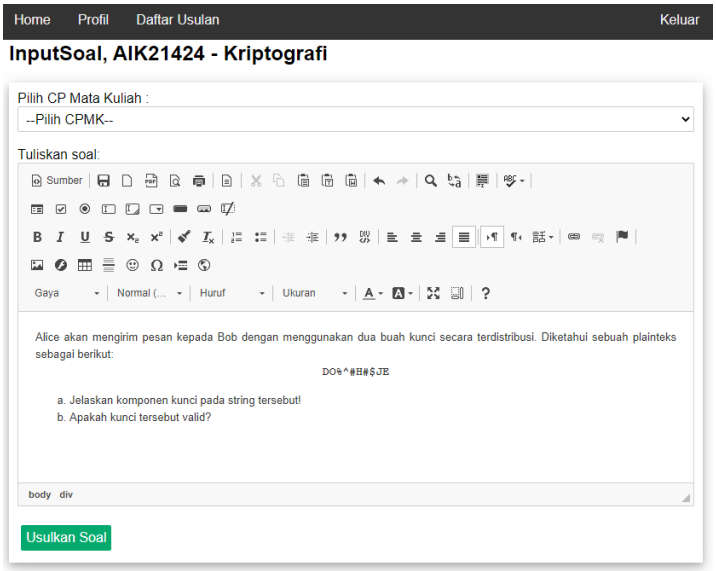
3.2 Implementasi Sistem

Pembahasan selanjutnya dilakukan pada fungsi-fungsi utama sistem yaitu: halaman login, daftar usulan soal, input dan validasi soal, pengolah teks, serta pengelola halaman PDF. Sistem dikembangkan agar dapat diakses melalui perangkat komputer dan perangkat ponsel sehingga perlu dilakukan penyesuaian tampilan sesuai dengan jenis perangkat yang digunakan. Hasil implementasi halaman awal website dapat dilihat pada Gambar 5.



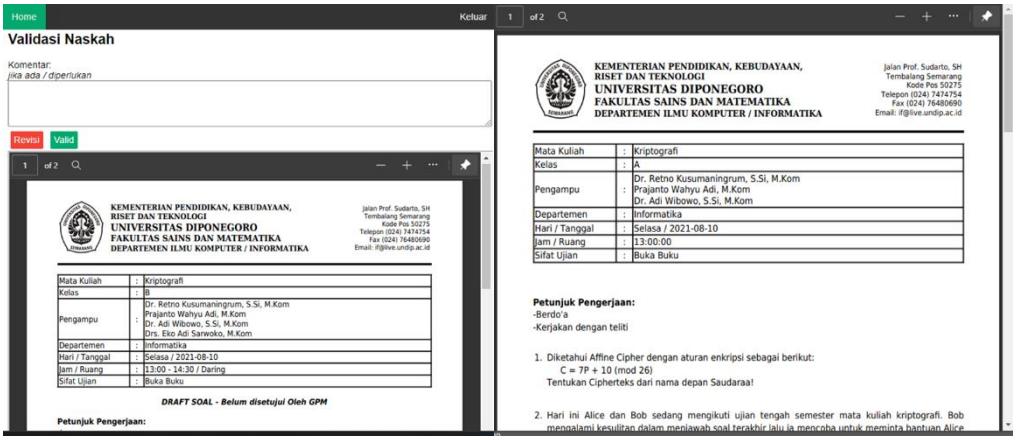
Gambar 5 Tampilan awal website pada perangkat komputer (kiri) dan ponsel (kanan)

Untuk mempermudah pengolahan usulan soal, sistem ini menggunakan *library* pengolah dokumen CKEditor 4 yang memiliki fitur yang sesuai dengan kebutuhan dalam pengolahan teks soal. Menu-menu utama seperti pengaturan teks, pemberian sub nomor soal, pembacaan persamaan (*equation*), hingga menambahkan gambar melalui sebuah URL. Penambahan fungsi dilakukan pada fitur tambah gambar agar pengguna dapat mengunggah gambar dari perangkatnya secara langsung ke server. Hasil implementasi pengolah dokumen dapat dilihat pada Gambar 6.



Gambar 6 Fitur Pengolah Teks dengan CKEditor 4

Usulan soal yang telah valid selanjutnya dapat diusulkan oleh koordinator kepada GPM. Soal-soal tersebut akan diproses oleh fungsi pengelola naskah yang dapat mengatur tata letak (*layout*), penomoran, header, dan identitas soal sesuai dengan data ada. Sistem ini menggunakan library mPDF versi 7 dengan menggunakan Composer sebagai *dependency manager* PHP untuk mengelola naskah soal menjadi draft soal dengan format PDF. Draft soal yang telah diusulkan selanjutnya dapat divalidasi oleh GPM dan selanjutnya dapat dicetak oleh koordinator mata kuliah. Hasil pengolahan draft soal dan naskah soal yang telah valid ditampilkan pada Gambar 7.



Gambar 7 Proses Validasi Draft Soal (kiri) dan Naskah Soal yang telah Valid (kanan)

4. KESIMPULAN DAN SARAN

Penelitian ini dilakukan untuk mengembangkan sebuah sistem manajemen naskah soal yang mampu menghasilkan naskah yang sesuai dengan format standar, memiliki tingkat keamanan akun yang baik, dan mudah digunakan. Dari hasil pengembangan dan pengujian yang telah dilakukan diperoleh hal-hal sebagai berikut:

1. Sistem dikembangkan dengan menggunakan *pre-hash coding* yang mampu mengamankan *password* alfanumerik dengan jumlah karakter yang pendek. Hal ini sesuai dengan kebutuhan pengguna dalam sistem yang menginginkan *password* yang pendek dan mudah diingat namun tetap aman.
2. Sistem pre-*hash* code dengan operator *bitwise xor* terhadap nilai unik pengguna

mampu menggagalkan peretasan hingga 40% pada karakter alfanumerik dengan panjang minimal 5 karakter.

3. Sistem *pre-hash code* dengan operator matematika penjumlahan dengan nilai unik penjumlahan mampu menggagalkan peretasan hingga 100% pada karakter alfanumerik dengan panjang minimal 5 karakter.

Pada penelitian selanjutnya dapat dikembangkan percobaan dengan menggunakan perangkat komputer terdistribusi dengan jumlah sampel yang lebih banyak sehingga mampu menghasilkan pengujian yang lebih baik. Penggunaan algoritma *hash*ing lain seperti SHA dan NTLM juga dapat dilakukan untuk sebagai bahan perbandingan antara kebutuhan komputasi dan tingkat keamanan yang dihasilkan.

DAFTAR PUSTAKA

[1] A. Artamonov, K. Ionkina, E. Tretyakov, and A. Timofeev, "Electronic document processing operating map development for the implementation of the data management system in a scientific organization," *Procedia Comput. Sci.*, vol. 145, pp. 248–253, 2018, doi: 10.1016/j.procs.2018.11.053.

[2] A. Ayaz and M. Yanartaş, "An analysis on the unified theory of acceptance and use of technology theory (UTAUT): Acceptance of electronic document management system (EDMS)," *Comput. Hum. Behav. Reports*, vol. 2, no. March, p. 100032, 2020, doi: 10.1016/j.chbr.2020.100032.

[3] L. Rusdiana, "Extreme programming untuk rancang bangun aplikasi pengelolaan surat keterangan kependudukan," *Regist. J. Ilm. Teknol. Sist. Inf.*, vol. 4, no. 1, pp. 49–55, 2018, doi: 10.26594/register.v4i1.1191.

[4] Y. Permana, H. D. Wijaya, P. Studi, T. Informatika, and U. M. Buana, "Implementasi E-Legalisir Untuk Legalisir Ijazah dan Transkrip Online pada Fakultas Ilmu Komputer Universitas Mercu Buana," *Techno.COM*, vol. 19, no. 2, pp. 103–114, 2020.

[5] K. Fitri, Z. Zulhendra, and D. Kurniadi, "Perancangan Sistem Informasi Legalisir Dokumen Berbasis Web Di Fakultas Teknik Universitas Negeri Padang," *Voteteknika (Vocational Tek. Elektron. dan Inform.*, vol. 2, no. 2, 2018, doi: 10.24036/voteteknika.v2i2.4075.

[6] N. A. K. Febriyani and R. B. Hadiprakoso, "Rancang Bangun Aplikasi Naskah Dinas Elektronik Berbasis Web Menggunakan WDLC," *Ultim. InfoSys J. Ilmu Sist. Inf.*, vol. 12, no. 1, pp. 43–51, 2021, doi: 10.31937/si.v12i1.1747.

[7] R. R. Asaad, "Penetration Testing: Wireless Network Attacks Method on Kali Linux OS," *Acad. J. Nawroz Univ.*, vol. 10, no. 1, p. 7, 2021, doi: 10.25007/ajnu.v10n1a998.

[8] B. Hitaj, P. Gasti, G. Ateniese, and F. Perez-Cruz, *PassGAN: A deep learning approach for password guessing*, vol. 11464 LNCS. Springer International Publishing, 2019.

[9] C. Ntantogian, S. Malliaros, and C. Xenakis, "Evaluation of *password hash*ing schemes in open source web platforms," *Comput. Secur.*, vol. 84, pp. 206–224, 2019, doi: 10.1016/j.cose.2019.03.011.

[10] G. Kim, S. Kim, M. Park, Y. Park, I. Lee, and J. Kim, "Forensic analysis of instant messaging apps: Decrypting Wickr and private text messaging data," *Forensic Sci. Int. Digit. Investig.*, vol. 37, p. 301138, 2021, doi: 10.1016/j.fsidi.2021.301138.

[11] M. Park, G. Kim, Y. Park, I. Lee, and J. Kim, "Decrypting *password*-based encrypted backup data for Huawei smartphones," *Digit. Investig.*, vol. 28, pp. 119–125, 2019, doi: 10.1016/j.diin.2019.01.008.

[12] S. F. Dyson, W. J. Buchanan, and L. Bell, "Scenario-based creation and digital investigation of ethereum ERC20 tokens," *Forensic Sci. Int. Digit. Investig.*, vol. 32, p. 200894, 2020, doi: 10.1016/j.fsidi.2019.200894.

[13] R. Hranický, L. Zobal, O. Ryšavý, and D. Kolář, "Distributed *password* cracking with BOINC and *Hashcat*," *Digit. Investig.*, vol. 30, pp. 161–172, 2019, doi: 10.1016/j.diin.2019.08.001.

[14] A. Kanta, I. Coisel, and M. Scanlon, "A survey exploring open source Intelligence for smarter *password* cracking," *Forensic Sci. Int. Digit. Investig.*, vol. 35, p. 301075, 2020, doi: 10.1016/j.fsidi.2020.301075.

[15] Q. Guo *et al.*, "PUFPass: A *password* management mechanism based on software/hardware codesign," *Integration*, vol. 64, no. July 2018, pp. 173–183, 2019, doi: 10.1016/j.vlsi.2018.10.003.

[16] Y. Guo, Z. Zhang, and Y. Guo, "Optiwords: A new *password* policy for creating memorable and strong *password*s," *Comput. Secur.*, vol. 85, pp. 423–435, 2019, doi: 10.1016/j.cose.2019.05.015.

[17] C. Tsagkani and A. Tsalgatidou, "Process model abstraction for rapid comprehension of complex business processes," *Inf. Syst.*, vol. 103, p. 101818, 2022, doi: 10.1016/j.is.2021.101818.

[18] E. Hariyanti, A. Djunaidy, and D. Siahaan, "Information security vulnerability prediction based on business process model using machine learning approach," *Comput. Secur.*, vol. 110, p. 102422, 2021, doi: 10.1016/j.cose.2021.102422.