Jurnal EurekaMatika

Journal homepage: https://ejournal.upi.edu/index.php/JEM

Implementasi *QR Code* dengan Algoritma *Secure Hash Algorithm* (SHA)-256 dan Rivest Shamir Adleman (RSA) yang Ditingkatkan untuk Autentikasi Dokumen Digital

Fatimah Az-Zahra, Rini Marwati, dan Ririn Sispiyati*

Program Studi Matematika, Fakultas Pendidikan Matematika dan Ilmu Pengetahuan Alam, Universitas Pendidikan Indonesia, Indonesia

*Correspondence: E-mail: rinimarwati@upi.edu

A B S T R A K

Teknologi digital berkembang dengan sangat pesat, termasuk salah satunya adalah digitalisasi dokumen. Dengan adanya peralihan bentuk dokumen, tanda tangan basah juga beralih menjadi tanda tangan elektronik. Selama proses distribusi dokumen digital, tidak ada jaminan bahwa isi dokumen tidak dimodifikasi oleh pihak lain sehingga diperlukan alat untuk melakukan autentikasi keaslian dokumen. Ilmu kriptografi dapat menjadi solusi, khususnya kriptografi kunci asimetri. Algoritma Rivest-Shamir-Adleman (RSA) adalah jenis kriptografi asimetri dengan keamanan yang terletak pada masalah pemfaktoran bilangan bulat. Untuk mempercepat proses enkripsi dan dekripsi, digunakan RSA yang ditingkatkan dengan cara menambahkan satu bilangan prima pada proses pembangkitan kunci. Selain kriptografi kunci asimetri, terdapat satu fungsi yang berguna dalam pembuatan tanda tangan elektronik, yaitu fungsi hash. Penggabungan algoritma RSA dan fungsi hash jenis SHA-256 dapat menjadi salah satu solusi untuk autentikasi dokumen digital. Selain itu, adanya penambahan QR Code juga dapat lebih mempermudah dalam proses penandatanganan dan pemeriksaan keaslian dokumen.

INFORMASI ARTIKEL

Sejarah Artikel:
Diterima 20 Januari 2024
Direvisi 25 Februari 2024
Disetujui 25 April 2024
Tersedia online 1 Mei 2024
Dipublikasikan 2 Mei 2024

Kata Kunci:
Autentikasi Dokumen,
QR Code,
RSA yang Ditingkatkan,
SHA-256,
Tanda Tangan Elektronik.

A B S T R A C T

Digital technology is developing very rapidly, including document digitization. With the shift in the form of documents, wet signatures have also shifted to electronic signatures. During the distribution process of digital documents, there is no guarantee that the contents of the document are not modified by other parties, so a tool is needed to authenticate the authenticity of the document. Cryptography can be a solution, especially asymmetric key cryptography. The Rivest-Shamir-Adleman (RSA) algorithm is a type of asymmetric cryptography with security that lies in the integer factoring problem. To speed up the encryption and decryption process, an improved RSA is used by adding one prime number to the key generation process. In addition to asymmetric key cryptography, there is one function that is useful in creating electronic signatures, which is the hash function. The combinations of the RSA algorithm and the SHA-256 hash function can be one solution for digital document authentication. In addition, the addition of a QR Code can also make it easier to sign and check the authenticity of documents.

Keywords:
Document Authentication,
Electronic Signature,
Enhanced RSA,
QR Code,
SHA-256.

1. PENDAHULUAN

Kriptografi berasal dari Bahasa Yunani, yaitu *crypto* yang berarti *secret* (rahasia) dan *graphia* yang berarti *writing* (tulisan). Secara terminologi, kriptografi adalah ilmu dan seni dalam menjaga keamanan isi pesan ketika pesan tersebut didistribusikan. Dalam kriptografi, terdapat beberapa istilah penting yang sering digunakan, seperti enkripsi (proses mengacak atau menyembunyikan pesan), dekripsi (proses merapikan atau mengembalikan pesan), plainteks (pesan asli), dan cipherteks (pesan acak). Selain dibagi ke dalam jenis klasik dan modern, kriptografi dapat dibedakan juga ke dalam jenis kunci simetris dan kunci asimetri. Kriptografi kunci simetris menggunakan kunci yang sama dalam proses enkripsi dan dekripsi sehingga algoritma ini disebut juga dengan *single-key algorithm*. Sementara itu, kriptografi kunci asimetri menggunakan kunci yang berbeda dalam proses enkripsi dan dekripsi, yaitu kunci umum (*public key*) dan kunci rahasia (*private key*) (Atika, 2018).

Pada era serba teknologi saat ini, terlebih sejak terjadinya pandemi Corona virus 2019 (Covid-19) yang menerapkan *work from home* (WFH), hampir semua kegiatan dilakukan secara *online*, salah satunya dalam kebutuhan tanda tangan sebuah dokumen. Pada Peraturan Presiden No. 82 Tahun 2012 tentang Penyelenggaraan Sistem Transaksi Elektronik, disebutkan bahwa tanda tangan elektronik terdiri atas informasi elektronik yang dilekatkan, terasosiasi atau terikat dengan informasi elektronik lain yang berfungsi sebagai alat autentikasi dan verifikasi. Dengan demikian, tanda tangan elektronik bukanlah tanda tangan basah hasil *scan* yang ditaruh ke dalam dokumen digital. Salah satu metode yang dapat digunakan untuk menjamin keaslian tanda tangan elektronik adalah sistem kriptografi kunci publik atau algoritma kriptografi kunci asimetri (Melina *et al.*, 2022).

Salah satu algoritma kriptografi asimetri yang masih banyak digunakan saat ini adalah algoritma RSA (*Rivest Shamir Adleman*) dengan keamanannya yang terletak pada masalah pemfaktoran bilangan bulat atau *integer factorization problem* (IFP). Algoritma ini terdiri dari tiga proses, yaitu pembangkitan kunci yang akan menghasilkan kunci publik (e, n) dan kunci privat (d, n), enkripsi plainteks, dan dekripsi cipherteks. Dibanding algoritma kriptografi asimetri ElGamal, algoritma RSA lebih unggul dalam kecepatan waktu enkripsi dan dekripsi, serta menghasilkan jumlah karakter cipherteks yang lebih sedikit (Himawan *et al.*, 2016). Meskipun begitu, algoritma ini terbilang lebih lemah dalam hal tingkat keamanan dibanding algoritma ElGamal dan membutuhkan waktu hingga empat miliar tahun jika n yang dipilih memiliki panjang 200 digit. Oleh sebab itu, untuk mempercepat durasi waktu proses dan meningkatkan keamanan, digunakan algoritma RSA yang ditingkatkan dengan menambahkan satu bilangan prima pada proses pembangkitan kunci sehingga mengakibatkan bilangan n yang diperoleh lebih besar. Ketika bilangan n semakin besar, proses perhitungan $a \bmod n$ akan lebih cepat (Firdaus *et al.*, 2018).

Dalam melakukan enkripsi, terdapat sebuah fungsi yang disebut dengan *hashing*. Fungsi hash adalah fungsi yang menerima masukan bentuk *string* apa saja dengan panjang sembarang dan mengonversinya menjadi sebuah teks khusus dengan panjang tetap yang disebut *message digest*. Fungsi hash memiliki banyak jenis, salah satunya adalah *Secure Hash Algorithm* (SHA). Pada penelitian tahun 2018, Atika menyimpulkan bahwa algoritma RSA dan SHA-1 dapat dikombinasikan dengan baik dalam membuat tanda tangan digital, serta dapat diandalkan dalam autentikasi file. Namun, Wang *et al.* (2005) telah menemukan kolisi pada SHA-1 dengan mengubah kompleksitas dari 2^{80} menjadi 2^{69} sehingga dianggap sudah tidak

aman dan tidak disarankan untuk digunakan kembali. Salah satu fungsi hash jenis SHA lainnya yang belum ditemukan kolisi hingga saat ini adalah SHA-256. SHA-256 merupakan fungsi hash satu arah (*One-Way Function*) versi SHA dengan ukuran *digest* sebesar 256 bit pada versi SHA-2 yang dirancang oleh *The National Institute of Standards and Technology* (NIST) pada tahun 2002. Berdasarkan *Secure Hash Signature Standard*, pesan masukan dengan panjang yang lebih pendek dari 2^{64} bit harus dioperasikan oleh 512 bit dalam kelompok dan menghasilkan sebuah *message digest* 256 bit (Sulastri & Putri, 2018).

Quick Response Code (QR Code) adalah sebuah tipe kode batang yang memuat lebih dari 4000 karakter data dalam bentuk kontak berwarna hitam dan putih. Kode tersebut dapat berisi sebuah tautan yang merujuk kepada alamat email, informasi kontak, data geografi, pesan teks, gambar, serta informasi video dan audio. QR Code menjadi satu satu solusi dalam peralihan kegiatan *offline* ke *online* karena tautan apapun dapat diakses secara manual dengan menggunakan kamera *smartphone* (Deineko *et al.*, 2022).

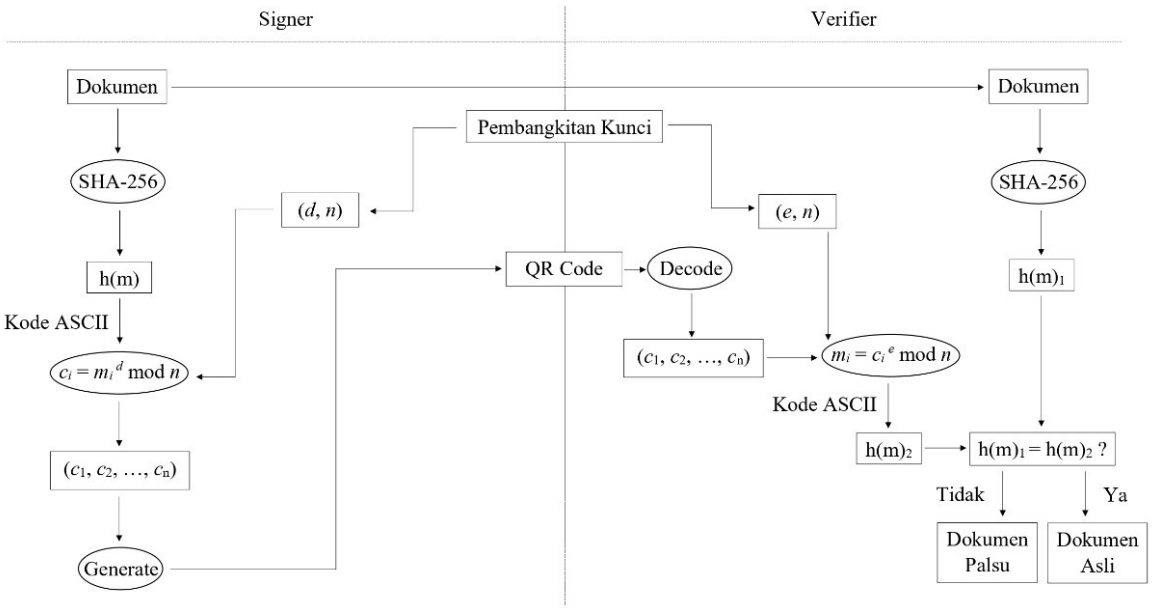
Beberapa penelitian sebelumnya yang telah menggabungkan algoritma kriptografi RSA dan QR Code, di antaranya dilakukan oleh Hendrawaty *et al.* (2016) yang menyimpulkan bahwa penggabungan QR Code dan algoritma RSA dapat berjalan dengan baik. Penelitian lain dilakukan oleh Sarasvananda & Iswara (2022) yang menyimpulkan bahwa penggunaan algoritma RSA dengan QR Code dalam melakukan tanda tangan elektronik dapat mengamankan keaslian dokumen dan mempermudah proses penandatanganan. Penelitian lain dilakukan oleh Pangan *et al.* (2022) yang berhasil menciptakan VacciFied.net, Sistem Pencatatan Covid-19 Terpusat, dengan penggabungan QR Code dan algoritma RSA.

Pada artikel ini, penulis menggabungkan algoritma RSA yang ditingkatkan dan algoritma *hashing* SHA-256 untuk membuat tanda tangan elektronik. Kemudian, tanda tangan tersebut akan dibuat ke dalam QR Code sehingga mempermudah proses autentikasi dokumen digital. Semua proses tersebut akan dikonstruksi menjadi sebuah program aplikasi menggunakan bahasa pemrograman Python dengan bantuan *software* Visual Studio Code.

2. METODE

Metode yang digunakan dalam pembuatan tanda tangan elektronik dan proses autentikasi dokumen digital pada artikel ini adalah penggabungan algoritma *hashing* SHA-256 dan algoritma kriptografi asimetri RSA, lalu hasil dari tanda tangan elektronik tersebut diringkas ke dalam QR Code. Dalam algoritma ini, digunakan dua model dasar, yaitu model fungsi hash, serta model enkripsi dan dekripsi RSA. Selain digunakan untuk enkripsi dan dekripsi, algoritma RSA juga dapat digunakan dalam pembuatan tanda tangan digital dan proses autentikasi dokumen digital.

Gambar 1 menunjukkan bahwa terdapat tiga proses utama yang dilalui dalam penelitian ini, antara lain pembangkitan kunci, *signer*/pembuatan tanda tangan digital, dan *verifier*/autentikasi dokumen digital.



Gambar 1. Skema Implementasi QR Code dalam Autentikasi Dokumen Digital

2.1 Pembangkitan Kunci

Pada proses ini, dibangkitkan sebuah kunci publik dan kunci privat oleh pengirim dokumen. Kunci publik akan digunakan pada proses autentikasi dokumen oleh penerima dokumen dan tidak masalah jika diketahui oleh orang lain, sedangkan kunci privat akan digunakan pada proses pembuatan tanda tangan oleh pengirim dokumen sehingga harus dirahasiakan dan tidak boleh diketahui oleh orang lain.

Dalam melakukan pembangkitan kunci, dibutuhkan tiga bilangan prima (p, q, r) dan satu bilangan prima lain (e). Selain itu, akan dihasilkan bilangan n dari perkalian p, q, dan r. Kunci publik (e, n), sedangkan kunci privat (d, n). Algoritma ini dilakukan oleh pengirim dokumen.

1. Pilih tiga bilangan prima besar berbeda secara acak, p, q, dan r.
2. Hitung $n = pqr$.
3. Hitung $\varphi(n) = (p - 1)(q - 1)(r - 1)$.
4. Pilih sebuah bilangan bulat e sebagai kunci publik dan harus relatif prima terhadap $\varphi(n)$, di mana $2 < e < \varphi(n)$.
5. Hitung kunci dekripsi, d, dengan persamaan:
$$ed \equiv 1 \,(mod\ \varphi(n))\ atau\ d \equiv e - 1 \,(mod\ \varphi(n))$$
6. *Output* yang diperoleh, yaitu:
   a. Kunci publik adalah pasangan (e, n) dan tidak rahasia.
   b. Kunci privat adalah pasangan (d, n) dan harus dirahasiakan.

2.2 Pembuatan Tanda Tangan Digital

Pada proses ini, dilakukan *hashing* terhadap dokumen yang akan ditandatangani dengan menggunakan SHA-256. Setelah dokumen diringkas menjadi pesan singkat berukuran 256 bit, dokumen akan ditandatangani dengan cara enkripsi menggunakan kunci privat yang telah dibangkitkan, lalu hasil tanda tangan elektronik diringkas menjadi sebuah QR Code.

Dalam melakukan pembuatan tanda tangan, hanya dibutuhkan dokumen dan kunci privat, serta nama pemberi tanda tangan. Keluaran dari proses ini adalah sebuah tanda tangan elektronik berbentuk QR Code yang berisi informasi tanda tangan elektronik. Algoritma ini dilakukan oleh pengirim dokumen.

1. Pilih sebuah dokumen digital yang akan diberi tanda tangan.
2. Lakukan *hashing* atau proses meringkas dokumen menjadi pesan singkat ukuran 256-bit.
3. Enkripsi *message digest* dengan kunci privat yang sudah dibangkitkan.
4. *Generate* QR Code yang berisi tanda tangan digital hasil enkripsi.
5. Kirim sepasang dokumen digital dan QR Code kepada penerima.

2.3 Autentikasi Dokumen Digital

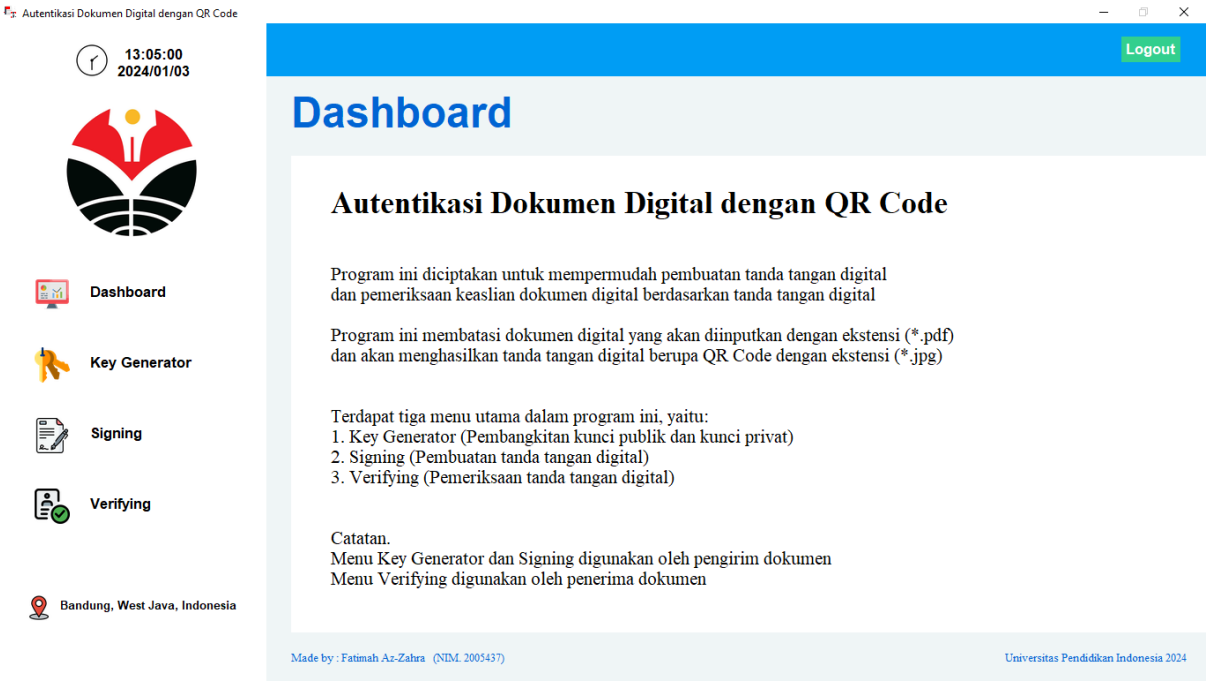
Pada proses ini, dilakukan pengecekan terhadap tanda tangan elektronik dan keaslian dokumen digital. Tanda tangan elektronik yang berada di dalam QR Code akan didekripsi dengan kunci publik yang telah dibangkitkan. Untuk autentikasi keaslian dokumen dan tanda tangan elektronik, hasil dekripsi dicocokkan dengan hasil *hashing* dokumen digital.

Dalam melakukan autentikasi dokumen, dibutuhkan dokumen, kunci publik, dan tanda tangan elektronik berbentuk QR Code. Proses verifikasi dilakukan dengan cara mencocokkan hasil perhitungan ketiga komponen tersebut. Keluaran dari proses ini adalah penjelasan terkait keaslian dokumen digital dan informasi elektronik dari tanda tangan elektronik. Algoritma ini dilakukan oleh penerima dokumen.

1. Lakukan *hashing* terhadap dokumen digital yang diterima sehingga diperoleh *message digest* dokumen (MD1).
2. Lakukan *decoding* terhadap QR Code yang diterima sehingga diperoleh tanda tangan digital.
3. Dekripsi tanda tangan digital dengan kunci publik yang sudah dibangkitkan sehingga diperoleh *message digest* tanda tangan (MD2).
4. Lakukan pencocokan terhadap MD1 dan MD2 dengan hasil:
   a. Jika cocok, maka dokumen digital dan QR Code autentik atau keduanya masih asli.
   b. Jika tidak cocok, maka dokumen digital dan QR Code tidak autentik atau terjadi perubahan pada salah satunya.

3. HASIL DAN PEMBAHASAN

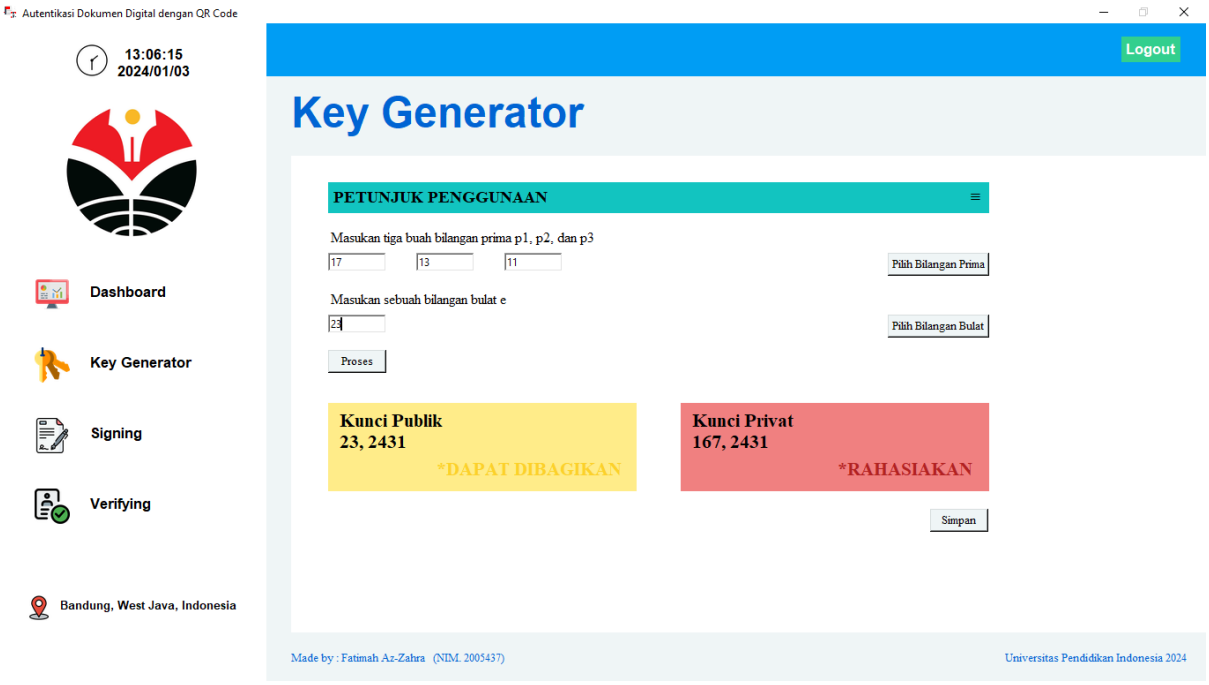
Untuk mempermudah proses autentikasi dokumen digital sebagai salah satu implementasi QR Code, dibuatlah sebuah program aplikasi komputer. Program dibuat menggunakan bahasa pemrograman Python versi 3.10 dan bantuan *software* Visual Studio Code pada komputer dengan spesifikasi Windows 10 64-bit, *processor* AMD Athlon Silver 3050U with Radeon Graphics, dan RAM 4 GB. Program aplikasi tersebut dikonstruksi menjadi tiga proses, yaitu pembangkitan kunci, pembuatan tanda tangan digital, dan autentikasi dokumen digital. Antarmuka utama program aplikasi ditampilkan pada Gambar 2.



Gambar 2. *Dashboard* Program Aplikasi

3.1 Pembangkitan Kunci

Dalam proses pembangkitan kunci, dimasukkan tiga bilangan prima berbeda (p, q, r) dan satu bilangan bulat yang relatif prima dengan $\varphi(n) = (p-1)(q-1)(r-1)$. Keluaran yang dihasilkan dari proses ini adalah sepasang kunci publik dan sepasang kunci privat. Hasil dari proses pembangkitan kunci dalam program aplikasi ditampilkan pada Gambar 3.



Gambar 3. *Key Generator* Program Aplikasi

Dipilih bilangan prima $p = 17, q = 13$, dan $r = 11$ sehingga diperoleh:

$$n = p \times q \times r = 17 \times 13 \times 11 = 2431$$

$$\varphi(n) = (p-1) \times (q-1) \times (r-1) = 16 \times 12 \times 10 = 1920$$

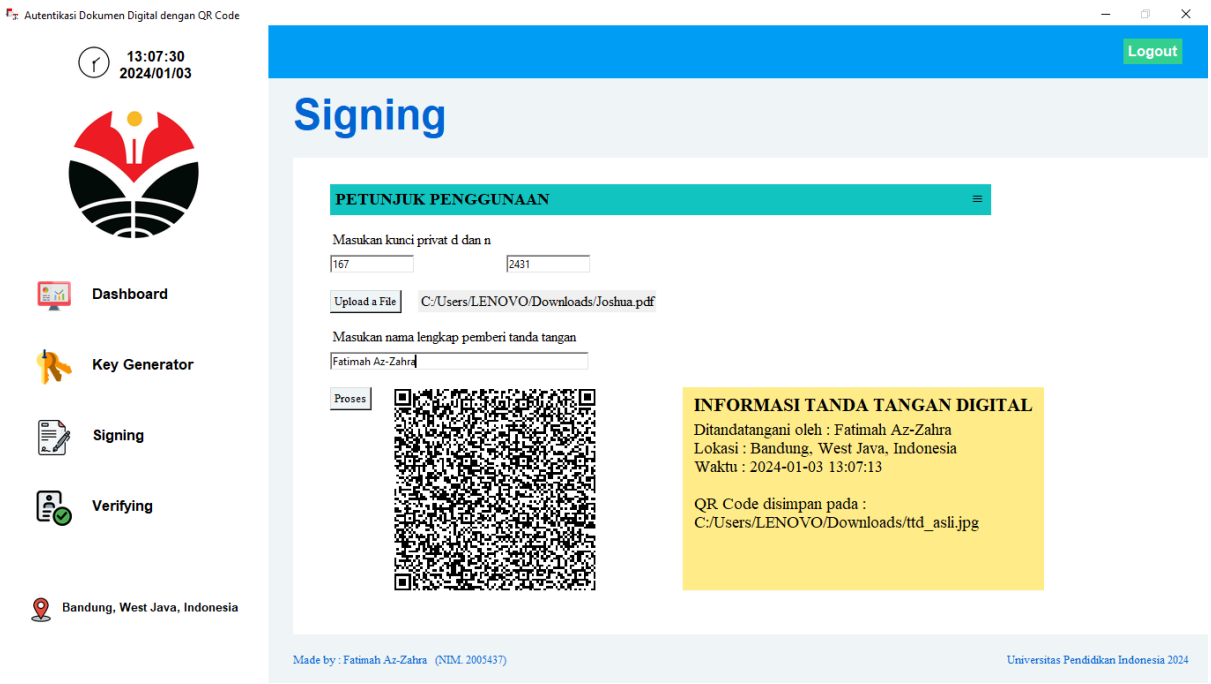
Dipilih juga bilangan bulat $e = 23$ yang relatif prima dengan $\varphi(n) = 1920$ sehingga diperoleh sepasang kunci publik $(e, n) = (23, 2431)$ yang dapat dibagikan dan digunakan untuk proses autentikasi dokumen digital. Selanjutnya, dicari bilangan bulat d dengan persamaan:

$e \times d \equiv 1 \ (mod\ \varphi(n))$, di mana d adalah invers e dalam modulo $\varphi(n)$

Dengan bantuan program invers modulo, dihasilkan bilangan bulat $d = 167$ sehingga diperoleh sepasang kunci privat $(d, n) = (167, 2431)$ yang harus dirahasiakan dan digunakan untuk proses pembuatan tanda tangan digital.

3.2 Pembuatan Tanda Tangan Digital

Setelah membangkitkan dua pasang kunci, proses dilanjutkan kepada pembuatan tanda tangan digital. Dalam proses pembuatan tanda tangan digital, dimasukkan sepasang kunci privat (d, n), sebuah dokumen digital dengan ekstensi PDF, dan nama lengkap pemberi tanda tangan. Keluaran yang dihasilkan dari proses ini adalah sebuah tanda tangan digital berupa QR Code yang berisi informasi tanda tangan. Hasil dari proses pembuatan tanda tangan digital dalam program aplikasi ditampilkan pada Gambar 4.



Gambar 4. *Signing* Program Aplikasi

Dipilih sebuah dokumen sertifikat digital yang akan ditandatangani. Dengan bantuan *library* yang disediakan oleh Python, nilai keluaran SHA-256 dari dokumen digital tersebut adalah b7eb993e5e90defc7921f57c423ed9b4547bf4e2cf767aa4fe21f749aa2466b1.

Selanjutnya, dibuat tanda tangan digital dengan enkripsi RSA yang ditingkatkan menggunakan kunci privat sehingga menghasilkan nilai keluaran [769, 880, 1954, 769, 1646, 1646, 2261, 1954, 1522, 1954, 1646, 159, 705, 1954, 1241, 2046, 880, 1646, 305, 212, 1241, 1522, 880, 2046, 1157, 305, 2261, 1954, 705, 1646, 769, 1157, 1522, 1157, 880, 769, 1241, 1157, 1954, 305, 2046, 1241, 880, 1099, 880, 466, 466, 1157, 1241, 1954, 305, 212, 1241, 880, 1157, 1646, 466, 466, 305, 1157, 1099, 1099, 769, 212]

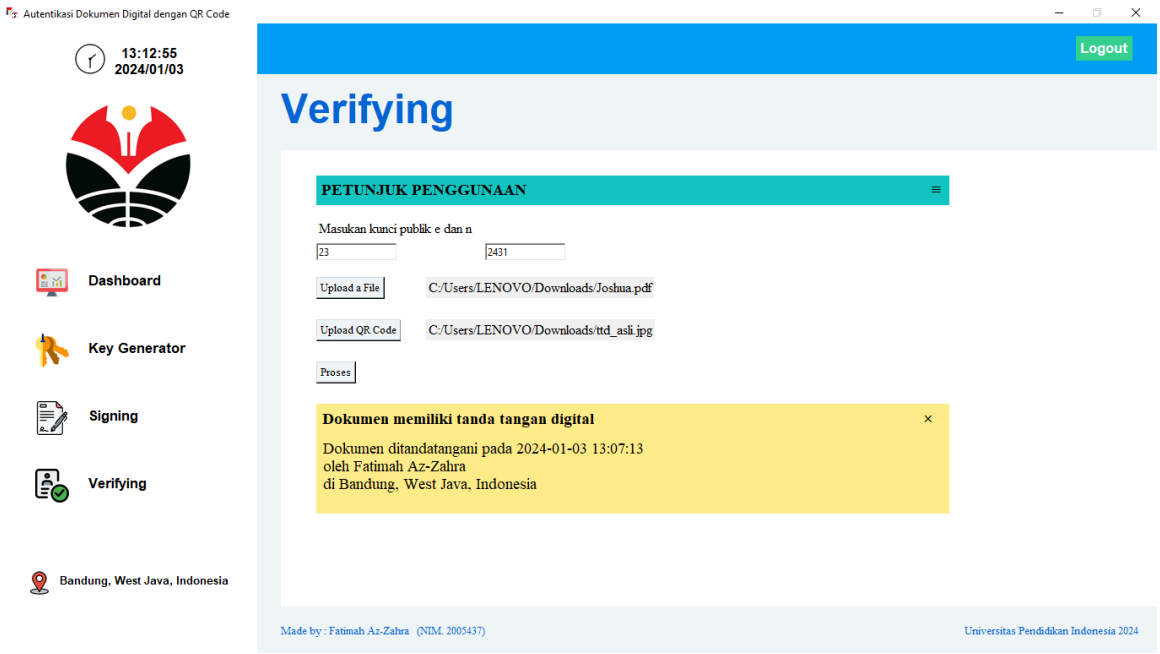
Hasil enkripsi RSA yang ditingkatkan tersebut dibuat ke dalam QR Code. Selain itu, dimasukkan pula lokasi dan waktu penandatanganan, serta pemberi tanda tangan sebagai informasi tambahan terkait tanda tangan digital.

3.3 Autentikasi Dokumen Digital

Dalam proses autentikasi dokumen digital, dimasukkan sepasang kunci publik (e, n), sebuah dokumen digital dengan ekstensi PDF, dan sebuah tanda tangan digital berupa QR Code. Keluaran yang dihasilkan dari proses ini adalah penjelasan terkait tanda tangan digital. Proses ini dibagi ke dalam tiga kasus, yaitu:

1. Autentikasi pada Dokumen Asli

   Misal penerima mendapatkan dokumen PDF asli dan tanda tangan QR Code asli, serta sepasang kunci publik. Hasil dari proses autentikasi pada dokumen asli dalam program aplikasi ditampilkan pada Gambar 5.



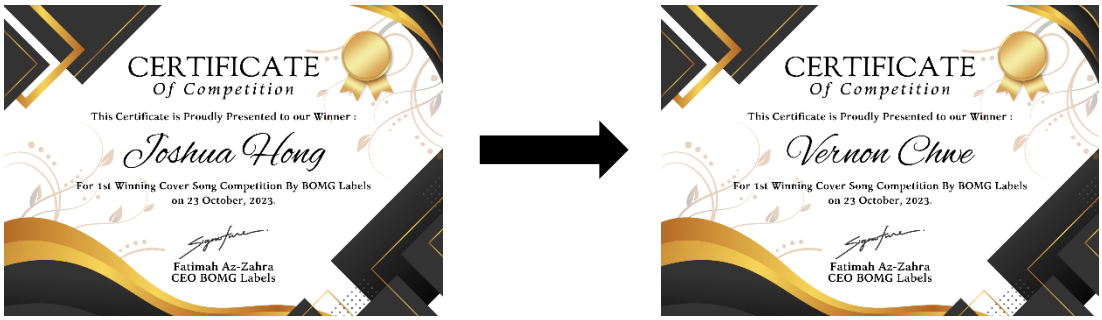
Gambar 5. *Verifying* Dokumen Asli Program Aplikasi

Dengan bantuan *library* yang disediakan oleh Python, diperoleh nilai hash dokumen b7eb993e5e90defc7921f57c423ed9b4547bf4e2cf767aa4fe21f749aa2466b1. Kemudian, dilakukan *decoding* terhadap tanda tangan QR Code dan hasilnya didekripsi dengan kriptografi RSA yang ditingkatkan sehingga menghasilkan nilai hash tanda tangan b7eb993e5e90defc7921f57c423ed9b4547bf4e2cf767aa4fe21f749aa2466b1.

Proses pencocokan kedua nilai hash tersebut menunjukkan bahwa dokumen digital dan tanda tangan digital autentik sehingga program aplikasi memunculkan beberapa informasi tambahan terkait tanda tangan digital, seperti waktu dan lokasi penandatanganan, serta pemberi tanda tangan.

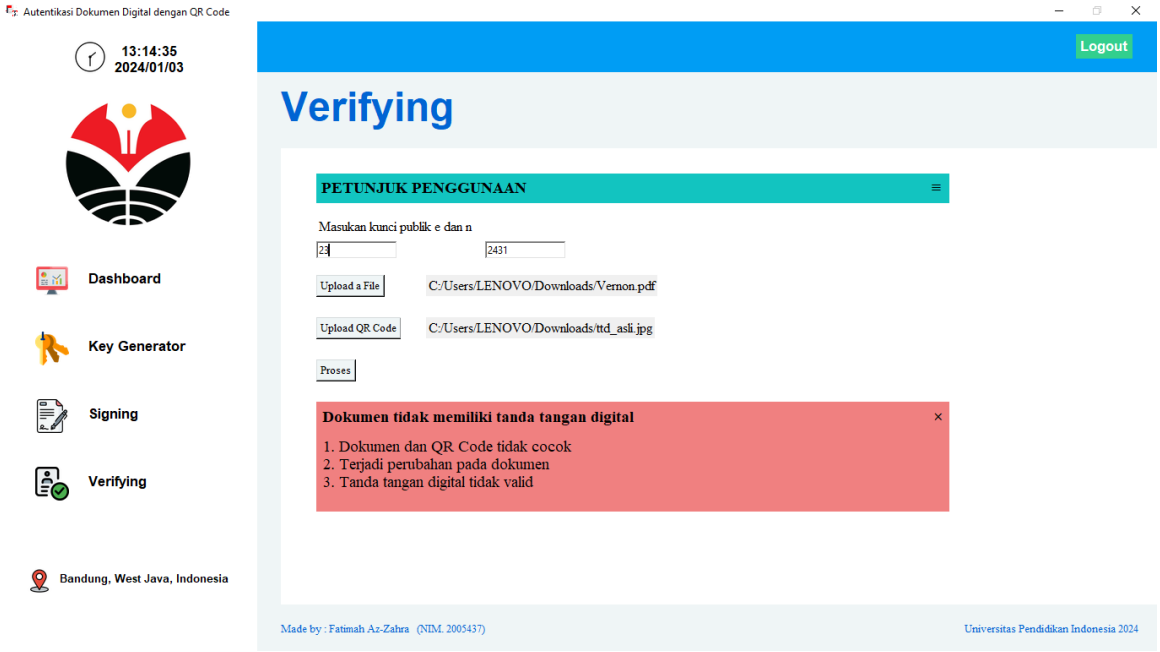
2. Autentikasi pada Perubahan Dokumen

Misal penerima mendapatkan dokumen PDF yang telah diubah dan tanda tangan QR Code asli, serta sepasang kunci publik. Perubahan dokumen PDF ditampilkan pada Gambar 6.



Gambar 6. Perubahan Dokumen

Perubahan pada isi konten dokumen akan memengaruhi nilai hash dokumen sehingga tidak akan cocok jika dilakukan autentikasi dengan tanda tangan asli. Hasil dari proses autentikasi pada dokumen asli dalam program aplikasi ditampilkan pada Gambar 7.



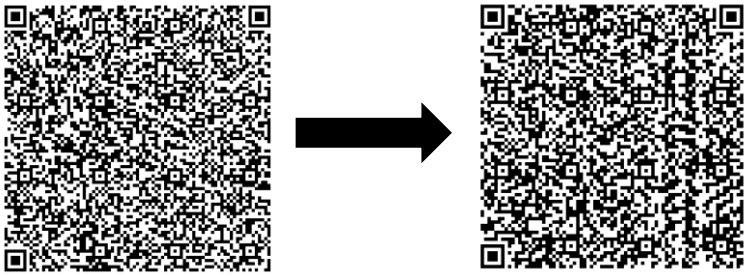
Gambar 7. *Verifying* Perubahan Dokumen Program Aplikasi

Dengan bantuan *library* yang disediakan oleh Python, diperoleh nilai hash dokumen aab86644c2687008a8790899201f287cfd8a44d27b05acf4d6a23b2a8988463a.
Kemudian, dilakukan *decoding* terhadap tanda tangan QR Code dan hasilnya didekripsi dengan kriptografi RSA yang ditingkatkan sehingga menghasilkan nilai hash tanda tangan b7eb993e5e90defc7921f57c423ed9b4547bf4e2cf767aa4fe21f749aa2466b1.

Proses pencocokan kedua nilai hash tersebut menunjukkan bahwa dokumen digital dan tanda tangan digital tidak autentik sehingga program aplikasi memunculkan beberapa kemungkinan alasan dokumen digital tersebut tidak memiliki tanda tangan digital. Pada kasus ini, kemungkinan yang paling cocok adalah alasan kedua, yaitu terjadi perubahan pada dokumen.

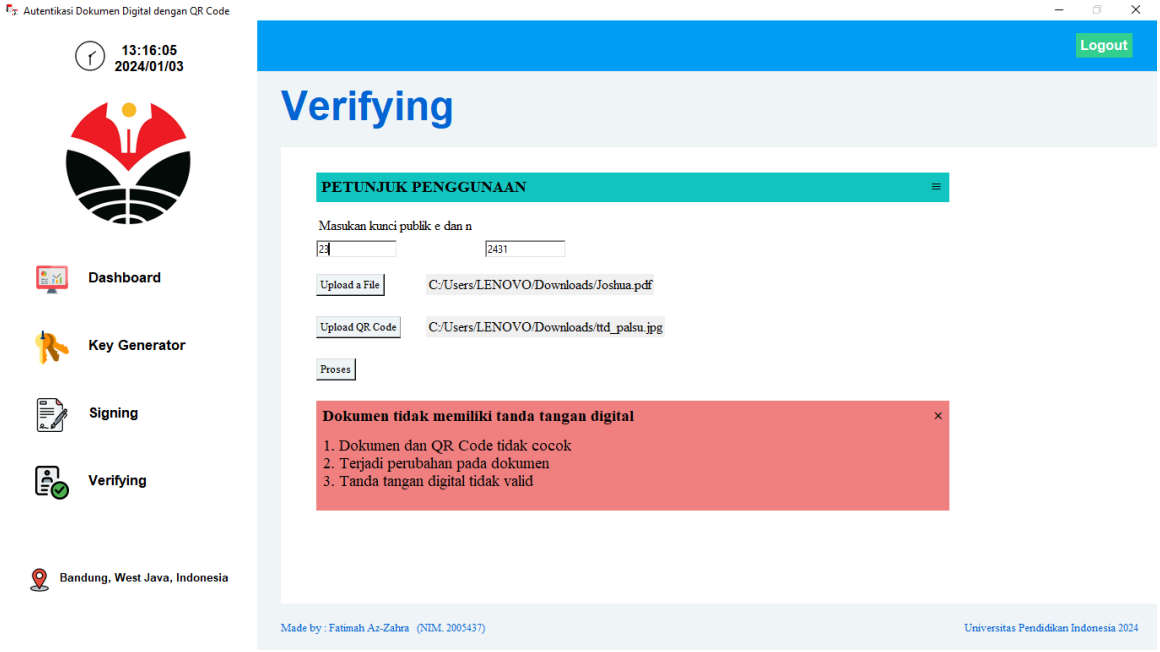
3. Autentikasi pada Pemalsuan Tanda Tangan

Misal penerima mendapatkan dokumen PDF asli dan tanda tangan QR Code yang telah dipalsukan, serta sepasang kunci publik. Pemalsuan tanda tangan QR Code ditampilkan pada Gambar 8.



Gambar 8. Pemalsuan Tanda Tangan

Pemalsuan pada tanda tangan akan mengacaukan perhitungan dekripsi sehingga karakter yang dihasilkan dari proses dekripsi tidak hanya akan memuat alfabet dan angka, serta tidak akan cocok jika dilakukan autentikasi dengan dokumen asli. Hasil dari proses autentikasi pada dokumen asli dalam program aplikasi ditampilkan pada Gambar 9.



Gambar 9. *Verifying* Pemalsuan Tanda Tangan Program Aplikasi

Dengan bantuan *library* yang disediakan oleh Python, diperoleh nilai hash dokumen b7eb993e5e90defc7921f57c423ed9b4547bf4e2cf767aa4fe21f749aa2466b1. Kemudian, dilakukan *decoding* terhadap tanda tangan QR Code dan hasilnya didekripsi dengan kriptografi RSA yang ditingkatkan sehingga menghasilkan nilai hash tanda tangan ЩЩDŽыы̆ФїыIDŽIõõDŽЩDŽISõDŽSSїо̊о̊їDŽIФóʿDŽЩ̆ʿїIӧѓЩФóʿыЩїсїЩDŽSDŽDŽысїЩ.

Penggabungan kunci publik dari pengirim asli dan tanda tangan yang sudah dipalsukan oleh oknum menyebabkan perhitungan dekripsi RSA yang ditingkatkan menjadi kacau sehingga nilai hash tanda tangan menghasilkan kumpulan karakter khusus selain alfabet dan angka.

Proses pencocokan kedua nilai hash tersebut menunjukkan bahwa dokumen digital dan tanda tangan digital tidak autentik sehingga program aplikasi memunculkan beberapa kemungkinan alasan dokumen digital tersebut tidak memiliki tanda tangan digital. Pada kasus ini, kemungkinan yang paling cocok adalah alasan ketiga, yaitu tanda tangan digital tidak valid.

4. KESIMPULAN DAN SARAN

Berdasarkan pembahasan hasil yang telah dipaparkan, dapat disimpulkan bahwa konsep *digital signature* dalam ilmu kriptografi dapat dimanfaatkan untuk proses autentikasi dokumen digital dengan memanfaatkan *Graphical User Interface* (GUI) Python, salah satunya adalah penggabungan fungsi hash jenis SHA-256 (*Secure Hash Algorithm* 256-bit) dan kriptografi kunci asimetri RSA (Rivest-Shamir-Adleman) yang ditingkatkan dengan cara penambahan satu bilangan prima pada proses pembangkitan kunci. Peningkatan algoritma RSA dalam proses pembuatan program aplikasi dapat lebih menjaga keamanan dan keaslian sebuah dokumen. Penggunaan QR Code menjadi salah satu cara agar proses akses tanda tangan digital dan autentikasi dokumen lebih mudah. Program aplikasi yang dihasilkan terdiri dari tiga menu utama. yaitu *key generator* dan *signing* yang digunakan oleh pengirim untuk membuat tanda tangan digital, serta *verifying* yang digunakan oleh penerima untuk melakukan autentikasi dokumen dan tanda tangan digital.

Untuk penelitian selanjutnya, penulis menyarankan adanya penggunaan gabungan algoritma kriptografi dan fungsi hash lain, lalu dikaji pula terkait kelebihan dan kekurangannya dari setiap algoritma sehingga mendapatkan algoritma yang terbaik untuk autentikasi dokumen digital menggunakan QR Code. Dalam segi program aplikasi, dapat disediakan dua opsi pada *input* autentikasi dokumen, yaitu *input* dokumen digital dan tanda tangan yang terpisah, serta *input* dokumen digital dan tanda tangan yang sudah disisipkan. Selain itu, *output* dapat dikategorikan berdasarkan kesalahan *input* sehingga penjelasan tidak terlalu umum ketika dokumen digital dan tanda tangan tidak autentik.

5. DAFTAR PUSTAKA

Atika, P. D. (2018). Digital signature dengan algoritma SHA-1 dan RSA sebagai autentikasi. *Jurnal Cendikia*, *16*(1), 74-83.

Deineko, Z., Kraievska, N., & Lyashenko, V. (2022). QR Code as an element of educational activity. *International Journal of Academic Information Systems Research (IJAISR)*, *6*(4), 26-31.

Firdaus, J., Marwati, R., & Muhtar, S. (2018). Penyandian pesan menggunakan kombinasi algoritma RSA yang ditingkatkan dan algoritma Elgamal. *Jurnal EurekaMatika*, *6*(1), 23-32.

Hendrawaty, H., Azhar, A., & Atthariq, A. (2016). Implementasi algoritma RSA dan QR code untuk keamanan transkrip nilai di Politeknik Negeri Lhokseumawe. *Jurnal Infomedia: Teknik Informatika, Multimedia & Jaringan*, *1*(2), 22-32.

Himawan, C., Wibowo, T., Sulityo, B., Roestam, R., Wahyu, Y., & Wahyu, R. B. (2016). Studi perbandingan algoritma RSA dan algoritma El-Gamal. *Semin. Nas. APTIKOM, 6*(1), 28-29.

Melina, M., Sukono, F., Napitupulu, H., & Kusumaningtyas, V. A. (2022). Verifikasi tanda tangan elektronik dengan teknik otentikasi berbasis kriptografi kunci publik sistem menggunakan algoritma kriptografi Rivest-Shamir-Adleman. *Jurnal Matematika Integratif*, *18*(1), 27-39.

Pangan, A. M. S., Lacuesta, I. L., Mabborang, R. C., & Ferrer, F. P. (2022). Authenticating data transfer using RSA-Generated QR Codes. *European Journal of Information Technologies and Computer Science*, *2*(4), 18-30.

Sarasvananda, I. B. G., & Iswara, I. B. A. I. (2022). Tanda tangan elektronik menggunakan algoritma Rivest Shamir Adleman (RSA) pada sistem informasi surat menyurat LPIK INSTIKI. *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, *11*(2), 289-296.

Sulastri, S., & Putri, R. D. M. (2018). Implementasi enkripsi data Secure Hash Algorithm (SHA-256) dan Message Digest Algorithm (MD5) pada proses pengamanan kata sandi sistem penjadwalan karyawan. *Jurnal Teknik Elektro*, *10*(2), 70-74.

Wang, X., Yin, Y. L., & Yu, H. (2005, August). Finding collisions in the full SHA-1. In *Proceedings of the 25th annual international conference on Advances in Cryptology* (pp. 17-36).