Cyberspace: Jurnal Pendidikan Teknologi informasi
Volume 6, Nomor 2, Oktober 2022, hal. 105-118
ISSN 2598-2079 (print) | ISSN 2597-9671 (online)

FULLY HOMOMORPHIC ENCRYPTION (FHE) PADA PENYIMPANAN DATA *E-GOVERNMENT* BERBASIS *CLOUD*

Rini Deviani[1], Sri Azizah Nazhifah[1], dan Aulia Syarif Aziz[2]

[1]Jurusan Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Syiah Kuala, Darussalam, Banda Aceh, 23111, Indonesia
[2]Jurusan Pendidikan Teknologi Informasi, Fakultas Tarbiyah dan Keguruan, UIN Ar-Raniry, Darussalam, Banda Aceh, 23111, Indonesia
E-mail: rini.deviani@unsyiah.ac.id, sriazizah07@unsyiah.ac.id, aulia.aziz@ar-raniry.ac.id

ABSTRACT

Cloud computing is a way of providing services, networks, hardware, storage, and interfaces for the construction of E-Government infrastructure that enables the delivery of services efficiently and achieves cost savings. Cloud services enable individuals and organizations to utilize cloud providers' software and hardware resources stored remotely. The span between the client and the actual location of his data provides a barrier because this data can be obtained by a third party, risking the citizen's data privacy. We examine a method based on the Fully Homomorphic Encryption (FHE) scheme in order to facilitate the processing of sensitive information pertaining to the E-Government that does not involve the disclosure of the original data. In this paper, we consider some general data operations to evaluate the feasibility of the FHE method and show that the accuracy are similar when data operations are applied to homomorphically encrypted data. The results of the experiment highlight the potential of the various privacy-preserving data operations that can be performed under FHE approach. These methods provide results that are equivalent to those achieved by unencrypted data and models within a decent amount of time.

Keywords: *Cloud Computing, E-Government, Cryptography, Homomorphic, Encryption*

ABSTRAK

Komputasi awan adalah gabungan komponen sistem teknologi informasi seperti servis, jaringan, perangkat keras, penyimpanan, dan *interface* untuk membuat infrastruktur *E-Government* yang dapat memberikan layanan secara efektif. Layanan *cloud* menyediakan kemampuan kepada setiap pengguna untuk dapat mengakses perangkat lunak dan perangkat keras yang disimpan secera jarak jauh oleh penyedia layanan *cloud*. Pada saat proses pengiriman data atau permintaan data ke penyedia layanan *cloud*, proses ini dapat menyebabkan data tersebut diperoleh oleh pihak ketiga. Pada penelitian ini kami menguji sebuah metode skema *Fully Homomorphic Encryption* (FHE) untuk memfasilitasi perlindungan kerahasiaan dan keamanan data pribadi yang berkaitan dengan *E-Government* yang tidak melibatkan pengungkapan data asli (*plaintext*). Dalam penelitian ini, kami memperhitungkan beberapa operasi

data yang umum dilakukan dan menguji apakah hasil dari operasi data tersebut memiliki akurasi serupa, ketika operasi data diterapkan pada data yang telah dienkripsi secara homomorfik maupun pada data yang tidak dienkripsi. Hasil pengujian menunjukkan bahwa operasi data dapat dilakukan dengan metode FHE. Penerapan metode FHE pada beberapa operasi data yang dilakukan pada penelitian ini memberikan hasil yang sebanding dengan hasil yang diperoleh pada data dan model yang tidak dienkripsi. Hasil didapatkan dalam waktu komputasi yang layak.

Kata Kunci: *komputasi awan, E-Government, kriptografi, Homomorfik, enkripsi*

1. PENDAHULUAN

Berbagai jenis *framework* untuk pengembangan sistem *Electronic Government* atau biasa disebut *E-Government* sangat berkembang pesat seiring dengan perkembangan teknologi informasi dan komunikasi. Bagian utama dari serangan bisa berupa data yang tidak diautentifikasi dan adanya perubahan informasi pada *E-Government* yang drastis. Maka dari itu framework *E-Government* harus dilindungi secara kuat sehingga data dan informasinya tetap aman dari serangan apapun.

Umumnya data yang disimpan pada *cloud* akan dilakukan proses enkripsi (*encryption*). Akan tetapi, jika pengguna akan melakukan pengolahan data, maka layanan *cloud* akan dilakukan proses dekripsi (*decryption*) dan mengirimkannya kembali ke pengguna dalam bentuk yang telah didekripsi (*decrypted*). Pada saat proses *encryption-decryption* di *cloud,* kemungkinan terjadinya kebocoran dan peretasan angatlah besar. Hal ini dapat dihindari dengan menggunakan teknik *Fully Homomorphic Encryption* (FHE).

Beberapa penelitian terkait keamanan data pada penyimpanan berbasis *cloud* menggunakan metode enkripsi homomorphik telah dilakukan pada beberapa tahun terakhir. Algoritma kriptografi homomorfik baru dan efektif yang ringan untuk tujuan keamanan data berbasis *cloud computing* dilakukan pada [1]. Adapun optimalisasi enkripsi homomorfik menggunakan *Application Programming Interface* (API) pada penyimpanan cloud dalam penelitian ilmu material yang melibatkan kumpulan data besar telah dilakukan pada [2]. Penelitian [3] mengusulkan skema enkripsi homomorfik praktis yang dapat memungkinkan pengguna data dalam sistem IoT untuk mengoperasikan data dengan aman melalui data terenkripsi, yang secara efektif dapat melindungi privasi data kunci dalam sistem.

Penelitian ini bertujuan untuk melakukan simulasi dan menganalisa kemampuan skema FHE dalam pengolahan data terenkripsi terhadap data yang disimpan menggunakan penyimpanan berbasis *cloud*. Penelitian ini membahas tentang hasil percobaan terhadap beberapa skema data yang umum digunakan pada aplikasi *E-Government*. Pengolahan data yang akan dilakukan percobaan diantaranya pencarian data, pengelompokkan data menggunakan K-Means, dan klasifikasi berbasis *Convolutional Neural Network* (CNN).

2. ENKRIPSI HOMOMORPHIC

Enkripsi digunakan untuk melindungi semua data yang disimpan pada *cloud*. Data akan didekripsi jika pengguna akan melakukan pemrosesan di dalam *cloud* agar data tidak rentan dari *hacker*. Pada saat pengguna mengakses atau mengolah data di penyimpanan berbasis *cloud*, hacker akan lebih mudah meretas data tersebut, maka dari itu Enkripsi Homomorfik dikembangkan untuk menghindari peretasan data.

Rivest pertama kali mengusulkan "*privacy homomorphism*" sebagai dasar untuk *Homomorphic Encryption* (HE) pada tahun 1978. Kemudian ia menguji HE untuk keamanan *computation* pada *Big Data* [4]. HE adalah jenis metode enkripsi yang memungkinkan modifikasi langsung dari *ciphertext*. Ide dasarnya adalah untuk memungkinkan mengolah data pada data terenkripsi dengan memanfaatkan sifat matematika tertentu dari berbagai skema enkripsi. Infomasi yang didapatkan tetap dalam keadaan terenkripsi dan dapat diproses lebih lanjut atau didekripsi.

Pada tahun 2009, Gentry mengusulkan teknik enkripsi homomorfik pertama berdasarkan *lattice* ideal [5]. Skema ini mampu melakukan operasi penjumlahan dan perkalian menggunakan *ciphertext* dalam beberapa kali, ini adalah versi pertama dari enkripsi homomorfik. Kemudian, perkembangan teknologi enkripsi homomorfik mengalami kemajuan yang sangat pesat. Teknologi enkripsi homomorfik dapat dipecah menjadi kategori berikut:

1. Teknik FHE pertama berbasis *lattice* ideal yang diusulkan oleh Gentry. Metode ini melibatkan konstruksi *SomeWhat Homomorphic Encryption* (SWHE), dimana operasi dilakukan pada bilangan terbatas baik operasi penjumlahan maupun perkalian.
2. Berdasarkan konsep Gentry tentang skema enkripsi homomorfik berbasis bilangan bulat [6], yang tidak memerlukan operasi berdasarkan *lattice* ideal dari *ring* polinomial ideal. Hanya integer yang digunakan dalam semua operasi. Ini juga disebut *Partially Homomorphic Encryption* (PHE), dalam jenis enkripsi ini, hanya satu operasi yang dapat dilakukan pada data terenkripsi baik dengan penambahan atau perkalian. Kriptosistem Pillar hanya melakukan operasi penjumlahan sedangkan kriptosistem RSA melakukan operasi perkalian pada data.
3. Metode terakhir adalah sistem *Fully Homomorphic Encryption* (FHE) yang didasarkan pada *Learning with Errors* (LWE) atau *Learning with Errors over Ring* (R-LWE). Sistem ini dibangun di atas pembelajaran yang toleran terhadap kesalahan, dan menghasilkan skema enkripsi yang sepenuhnya homomorfik dengan memanfaatkan non-linearisasi. Salah satu contoh skema enkripsi jenis ini adalah skema enkripsi Brakerski-Gentry-Vaikuntanathan (BGV) [7].

Karakteristik FHE yang bersifat *multi-layer* menyebabkan sistem berjalan sangat lambat. Untuk mengatasi masalah ini, banyak peneliti telah menggabungkan beberapa skema. Dalam beberapa tahun terakhir, sejumlah *library* HE yang bersifat *open source* telah berkembang. Semuanya memiliki kelebihan dan kekurangannya masing-masing sesuai algoritma enkripsi yang diterapkan [8]. *Library* yang paling sering digunakan adalah IBM HELib [9], dimana *library* tersebut mengadopsi skema yang dikembangkan oleh Brakerski-Gentry-Vaikuntanathan (BGV) [7], dan skema pada Microsoft Simple Encrypted Arithmetic Library (SEAL) [10]. Skema ini didukung oleh skema Brakerski/Fan-Vercauteren (BFV) [11] dan skema Cheon-Kim-Kim-Song (CKKS) [12].

3. DATA *E-GOVERNMENT*

E-Government menyediakan otomatisasi terhadap semua kegiatan pemerintah dan meningkatkan efisiensi dari sebuah organisasi dan warga yang berpartisapasi dalam pemerintahan [13]. *E-Government* yang efektif pada suatu negara dapat ditunjukan dengan diterapkannya *E-Governance Requirements* dan *E-Governance Components*. *E-Government* diharapkan mampu meningkatkan kinerja pemerintah dan mudah berbagi informasi dengan warga kapanpun dan dimanapun. Dalam proses implementasi praktis

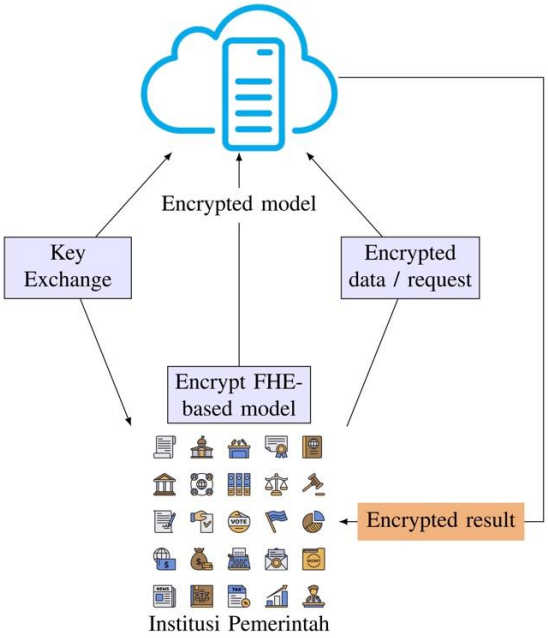
E-Government, ada beberapa hal penting untuk mengidentifikasi faktor-faktor khusus yang akan memainkan peran penting selama penerapan *E-Government*. Ada tiga aspek *E-Governance Requirements*:

1. *Government to Government*: Konektivitas antar pemerintah merupakan prasyarat untuk administrasi yang efektif, koordinasi antar pemerintah, dan akuntabilitas publik. Ini berfokus pada komunikasi antar internal pemerintahan, dan berbagai bentuk komunikasi antar lembaga pemerintahan.
2. *Government to Business*: Entitas bisnis sangat diperlukan bagi semua negara dan kontribusi secara signifikan terhadap perkembangannya. Pemerintah juga mengawasi organisasi ini untuk menjalankan kebijakan, standarisasi, dan akuntabilitas. Manajemen tender, administrasi kontrak, pembayaran pajak, dan transaksi *government to business* lainnya harus diotomatisasi.
3. *Government to Citizen:* Tugas utama setiap pemerintah adalah memberikan layanan kepada rakyatnya. Kebutuhan dasar, seperti makanan, tempat tinggal, dan pakaian, serta memberikan hak pendidikan yang sangat baik, perawatan kesehatan, dan layanan sosial lainnya, tidak dapat dicapai tanpa keterlibatan pemerintah. *E-Government* menyediakan kebutuhan warga, yang mungkin dapat dilakukan dengan sistem pemerintahan satu jendela.

4. PENERAPAN FHE PADA DATA *E-GOVERNMENT*

Keamanan data *E-Government* adalah salah satu masalah yang sangat penting pada skema *cloud computing*. Pengguna didorong untuk mengenkripsi data mereka sebelum menyimpannya di server untuk menjaga kerahasiaan informasi pribadi mereka. Teknik enkripsi homomorfik dapat segera mencari, menghitung, dan mengoperasikan data terenkripsi di cloud sekaligus melindungi kerahasiaan data *ciphertext* yang sedang diproses.

Pertama, pengguna masuk dan menggunakan pembuatan kunci (*key-generation*) yang disediakan server untuk menghasilkan sebuah kunci rahasia (*secret key*), yang hanya dipegang oleh pengguna. Pengguna kemudian mengenkripsi data sebelum mengirimnya ke *cloud*. Keutuhan dan keamanan data dapat dipastikan selama transmisi menggunakan teknologi kriptografi seperti tanda tangan digital. Pengguna dapat mengirimkan permintaan terenkripsi ke server *cloud* ketika ingin server melakukan pengolahan pada data terenkripsi ini (seperti pencarian). Server menjalankan proses yang diperlukan dan mengirimkan hasil terenkripsi kepada pengguna. Pengguna kemudian mendekripsi data dengan kunci rahasianya untuk menerima hasil yang benar [14]. Gambar 1 menggambarkan penerapan FHE pada data *E-Government* berbasis *cloud*.



Gambar 1. Penerapan FHE pada *E-Government*

Ada empat komponen utama untuk penerapan teknologi enkripsi homomorfik dalam komputasi awan [15]:

1. Pengambilan data terenkripsi dalam komputasi awan. Dengan menggunakan enkripsi yang menggunakan metode FHE, keamanan data komputasi awan dapat dipastikan. Ide dasarnya adalah bahwa data dienkripsi menggunakan enkripsi homomorfik dan disimpan di server *cloud*, yang memberikan manfaat yang besar. Kemudian, muncul masalah pada bagaimana memulihkan data terenkripsi. Pengambilan data berdasarkan teknologi enkripsi homomorfik tidak hanya dapat memperoleh data terenkripsi secara langsung, tetapi juga dapat memastikan bahwa data yang dikembalikan tidak dihitung atau diperiksa. Pada tahun 2011, Turaisingham mengakses dan mengambil data di platform cloud menggunakan teknologi Hive dan Hadoop [16]. Saat ini, ada beberapa perusahaan tertentu yang sudah menyediakan layanan pengambilan data terenkripsi.
2. Server *cloud* dapat memproses *ciphertext* secara langsung. Dalam komputasi awan, pemrosesan data terenkripsi sebagian besar terdiri dari pengambilan data, pengolahan data, statistik, dan analisis. Server *cloud* bekerja langsung pada data terenkripsi untuk memenuhi kebutuhan pengguna dan mengirimkan data yang telah diubah kembali ke pengguna. Setelah menerima *ciphertext*, pengguna dapat memecahkan kodenya yang mengurangi jumlah data yang telah dikirim. Ristenpart dkk menerbitkan sebuah jurnal pada tahun 2009 di mana mereka merekomendasikan penggunaan enkripsi homomorfik dalam platform layanan komputasi awan pertanian dan mengukur kemampuan beberapa teknik enkripsi [17]. Pada tahun 2013, perkembangan yang sangat pesat dibuat dalam penggunaan FHE dalam proses identifikasi biometrik. Dalam proses membandingkan dua vektor ciri biologis, jarak Hamming sering digunakan sebagai indikasi. Untuk menentukan jarak Hamming, Yasuda et al. menyajikan teknik FHE yang didasarkan pada lattice ideal [18].
3. Tempat penyimpanan data rahasia. Sejak konsep FHE pertama kali muncul, para peneliti mulai menyatakan untuk membuat gudang informasi yang dapat disimpan

secara rahasia. Data pengguna dapat disimpan di awan menggunakan komputasi awan, dan teknologi FHE dapat menjamin bahwa informasi yang diperlukan tidak diolah dan dianalisis. Teeba menyarankan pada tahun 2015 penggunaan enkripsi homomorfik hibrida untuk menjamin privasi bank data cloud [19].

4. *Multi-party cryptography* yang aman dan terhubung langsung ke enkripsi homomorfik adalah topik yang didiskusikan dalam penelitian Bendlin dkk. Dokumen [20] menunjukkan bagaimana enkripsi homomorfik dapat diterapkan untuk mengatasi tantangan multi-party cryptography yang aman dan luas.

5. EKSPERIMEN

Untuk memvalidasi penerapan FHE dalam berbagai jenis data *E-Government*, kami memfokuskan pada tiga jenis operasi data: pencarian data, clustering menggunakan K-Means, dan klasifikasi multikelas menggunakan *Convolutional Neural Network* (CNN). Percobaan ini dilakukan dengan menggunakan HElayers. HElayers adalah wadah Docker berbasis Linux yang dapat diimplementasikan sepenuhnya dalam perangkat lunak dan penyimpanan berbasis cloud. HELayers ditulis dalam C++ dan menyertakan API Python yang dirancang untuk memungkinkan pengembang aplikasi dan ilmuwan data dalam menerapkan teknik privasi tingkat lanjut dalam lingkungan Python terintegrasi dengan baik [21].

Tujuan dari eksperimen yang dilakukan bukanlah untuk mendapatkan metode pencarian data mutakhir, parameter pengelompokan, dan *deep learning-based outcomes* dalam penyelesaian masalahnya. Eksperimen ini dilakukan untuk melihat kelayakan dalam mempertahankan data yang bersifat privasi sekaligus dapat mengakses basis data, pengelompokan, dan model jaringan saraf untuk melakukan perhitungan dengan benar pada data dalam bentuk *ciphertext*. Penjelasan berikut ini akan membahas pengumpulan data dan metode untuk pengambilan sampel yang akan digunakan terhadap masalah yang sudah disebutkan diatas.

A. Privacy-Preserving Search

Privacy-preserving search adalah skenario umum untuk menunjukkan manfaat enkripsi homomorfik. Mampu melakukan pencarian data sambil menjaga privasi dan kerahasiaan dari parameter *query* yang memiliki banyak aplikasi di berbagai bidang industri mulai dari genomik hingga keuangan.

Contoh ini menunjukkan bahwa bagaimana seorang pengguna dapat menggunakan teknik berbasis enkripsi homomorfik untuk menghasilkan sebuah *mask*, dengan tujuan mengambil data dari database pasangan nilai kunci. Sehubungan dengan realisme data, dataset yang digunakan ialah data yang berisi nama negara-negara di dunia beserta ibukota negara tersebut dapat diakses pada laman website [22]. Dalam dunia nyata, contoh dataset ini bisa berupa informasi tentang data pelanggan atau catatan keuangan.

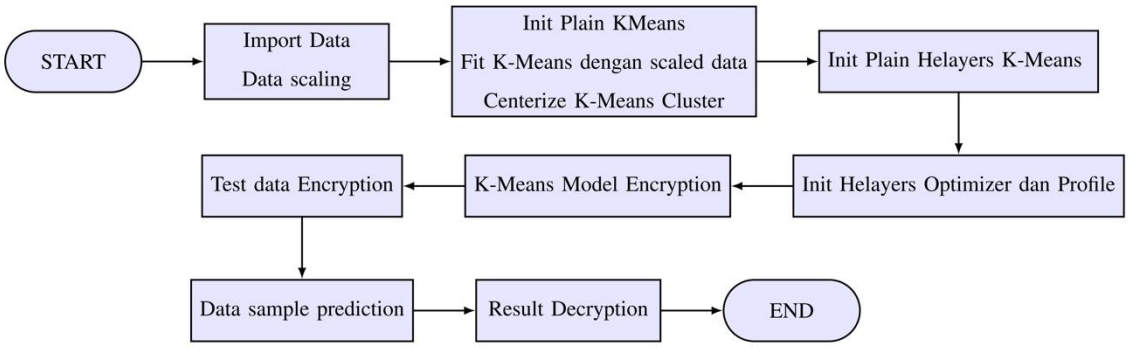
B. K-Means *Clustering*

K-Means *Clustering* adalah salah satu teknik pengelompokkan data tanpa pengawasan (*unsupervised*) yang paling sederhana dan paling populer. Biasanya, algoritma tanpa pengawasan membuat kesimpulan berdasarkan kumpulan data yang hanya menggunakan vektor sebagai input tanpa mengetahui hasilnya akan seperti apa, atau diberi label. Salah satu kasus penggunaan FHE potensial yang menggunakan K-Means adalah keamanan deteksi anomali dan dapat diterapkan pada kasus penggunaan rantai pasokan di berbagai banyak industri mulai dari otomotif hingga energi serta pertahanan.

Dalam eksperimen ini, kami akan menggunakan masalah segmentasi pelanggan grosir. Data yang digunakan merupakan data dari repository *UCI Machine Learning* [23]. Tujuannya adalah untuk mengelompokkan data klien distributor grosir berdasarkan pengeluaran tahunan mereka dengan mempertimbangan kategori produk yang beragam, seperti susu, bahan makanan, wilayah, dan lain sebagainya [24]. Uraian topologi K-Means yang digunakan dapat dilihat pada Tabel 1. Langkah-langkah dalam proses penerapan FHE pada operasi klustering menggunakan K-Means ditunjukkan pada Gambar 2.

TABEL 1 PARAMETER DARI K-MEANS *CLUSTERING* PADA KASUS DATA GROSIR

Tipe Parameter	**Ukuran Parameter**
Dimensi	8
Centroid	2
Batch	8192
Test data	8192



Gambar 2. Flowchart Penerapan FHE pada K-Means *Clustering* untuk Data Grosir

C. Convolutional Neural Network

Klasifikasi adalah tipe subjek yang diuji berkaitan dengan operasi jaringan saraf. Lebih khusus lagi, bidang klasifikasi citra digital berdasarkan informasi yang tergambar dalam citra digital. Sebagai patokan, untuk sistem klasifikasi gambar, basis data *Modified National Institute of Standards and Technology* (MNIST) berisi gambar digit tulisan tangan.

Dataset MNIST memiliki data training sebanyak 60,000 gambar dan data test 10,000 gambar digital dengan skala abu-abu dengan dimensi 28 × 28. Setiap gambar digital diberi label dengan nomor yang diwakilinya yaitu 0 hingga 9, seperti yang ditunjukkan pada Gambar 3. Gambar hitam putih (bilevel) asli dari MNIST dinormalisasi ukurannya agar berukuran 20 × 20 piksel namun tetap mempertahankan rasio aspeknya. Gambar yang dihasilkan mengandung tingkat keabuan sebagai hasil dari teknik *anti-aliasing* yang digunakan oleh algoritma normalisasi. Gambar dipusatkan dalam gambar 28 × 28 dengan menghitung pusat massa piksel, dan menerjemahkan gambar sehingga meletakkan titiknya di tengah bidang dimensi 28 × 28 [25].

Gambar 3. Contoh gambar dari MNIST *Dataset*

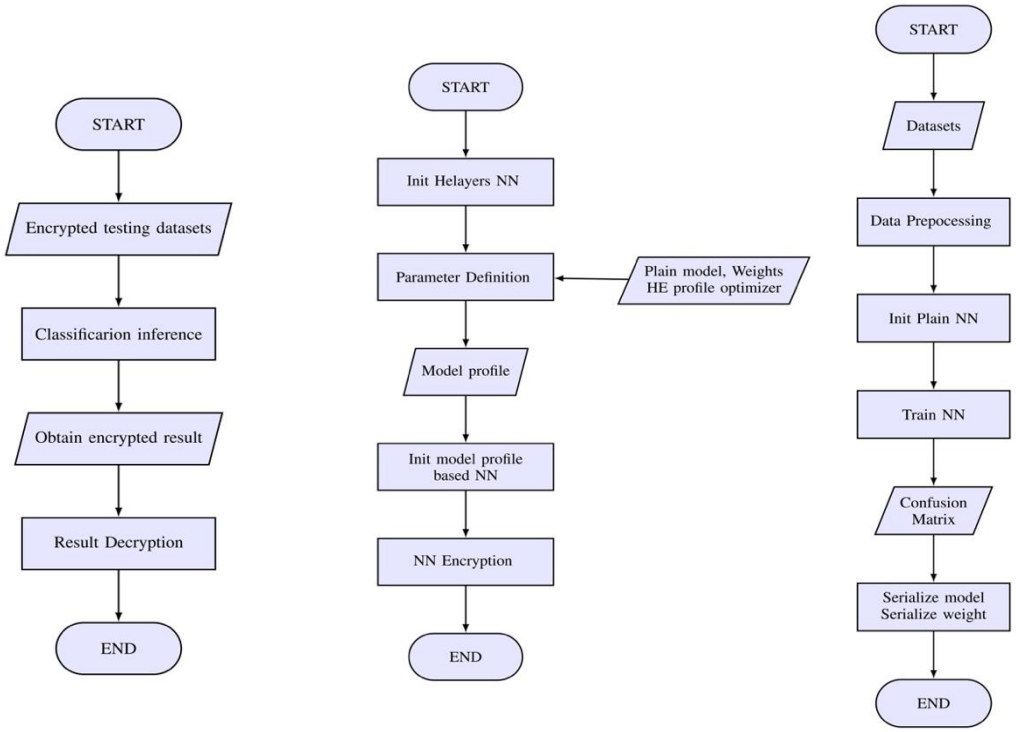
Percobaan ini dilakukan untuk mengidentifikasi satu set sampel dari dataset MNIST secara akurat menggunakan model jaringan saraf yang dikembangkan dan dilatih dengan menggunakan Python dengan *library* Keras [26]. Pada dasarnya, penerapan FHE pada *neural network* memiliki tiga alur aplikasi. Pertama, kita akan membangun model jaringan syaraf tiruan dengan mengimpornya dari *library* Python Keras. Langkah pertama ialah membuat model jaringan syaraf tiruan dalam membentuk *unenrypted (plain*) ditunjukkan pada diagram alur pada Gambar 4(a). Tabel 2 menguraikan topologi model CNN yang digunakan.

TABEL 2 PARAMETER DARI CNN UNTUK PENGENALAN TULISAN TANGAN MNIST

Layer (Tipe)	Output Shape	Jumlah Parameter
Conv2D	(13,13,5)	130
Flatten	845	0
Square Activation	845	0
Dense	100	84600
Square Activation	100	0
Dense	10	1010
Total Params		85740
Trainable Params		85740
Non-trainable Params		0
Batch size		500
Epoch		10

Langkah berikutnya adalah membangun objek jaringan saraf; yaitu versi yang terenkripsi dari jaringan dengan menggunakan perpustakaan Helayers. Seperti yang ditunjukkan pada Gambar 4(b), pengoptimal otomatis akan memeriksa jaringan dan memberikan profil HE, yang terdiri dari berbagai konfigurasi yang akan memfasilitasi eksekusi inferensi yang efisien di bawah enkripsi.

Langkah terakhir yang dilakukan ialah mengenkripsi *training dataset* menggunakan jaringan terenkripsi yang sudah dibuat untuk melakukan inferensi, dan membandingkan hasilnya dengan label yang diharapkan, seperti yang ditunjukkan pada Gambar 4(c).



(a) (b) (c)

Gambar 4. Diagram alur penerapan FHE pada klasifikasi digit MNIST berbasis CNN

Hasil klasifikasi dapat dilihat nilai akurasinya pada setelah dilakukan klasifikasi pada data uji. Dalam hal memahami peringkat akurasi, ada aturan umum untuk menilai apakah nilai akurasi tersebut dapat dinilai baik atau tidak [27], seperti terlihat sebagai berikut:

TABEL 3 ATURAN UMUM NILAI AKURASI PADA PEMBELAJARAN MESIN

Nilai Akurasi	Kategori
> 90 %	Sangat Baik
70%-90%	Baik
60%-70%	Cukup
< 60%	Kurang

6. HASIL

Untuk menganalisis hasil dari percobaan di atas, yaitu pada bagian pencarian dengan menjaga privasi penggunanya (*privacy-preserving search*), *unsupervised clustering* menggunakan K-Means dan klasifikasi menggunakan *CNN*, ada dua analisis yang diterapkan. Kedua analisis yang dinilai yaitu konsistensi dan kepraktisan dalam mengolah data *E-Goverment*. Setiap model akan diuji kinerja basis datanya dan hasil yang diperolehnya dengan menggunakan *privacy-preserving search, clustering* dan *deep learning models* terhadap model data *plaintext* dan *cyphertext.* Hasil yang akan didapatkan berasal dari kedua jenis data tersebut. Kami menganalisis dan mengukur kemampuan *privacy-preserving search*, *unsupervised clustering* dan klasifikasi menggunakan *CNN*. Setiap model akan dilihat akurasi dan kepraktisannya.

Untuk mendapatkan hasil yang lebih akurat, penelitian ini juga mengukur waktu (*runtime*) dari *privacy-preserving model, unsupervised clustering* menggunakan K-

Means dan klasifikasi menggunakan *CNN. Runtime* merupakan parameter yang penting dalam prosedur pengolahan data. Oleh karena itu, pada penelitian ini, pencarian data, *clustering*, dan klasifikasi baik terenkripsi maupun tidak terenkripsi, akan diuji secara detail.

A. Akurasi

Hasil data dekripsi yang diperoleh adalah sama dilihat dari segi operasi pencarian data parameter yang dipelajari modelnya, bahkan sampai level akurasi mesinya dengan data telah dicari dan dipelajari oleh model yang tidak terenkripsi. Kinerja keseluruhan model pencarian data, pengelompokan, dan *deep-learning* pada sampel pengujian adalah identik, baik model tersebut dilatih dengan enkripsi atau tanpa enkripsi.

1) Pencarian data nama negara. Proses ini merupakan pencarian nama ibukota negara dari nama negara yang dimaksud. Proses ini diuji dengan melihat hasil pencarian *database* yang terenkripsi menggunakan *query* terenkripsi. Hasil pencarian data terenkripsi memiliki nilai akurasi 100% dibandingkan dengan pencarian pada *database* tidak terenkripsi.
2) *Clustering* data konsumen grosir. Skema enkripsi FHE mendapatkan nilai akurasi 100% dengan membandingkan *cluster* yang kami dapatkan menggunakan algoritma K-Means pada data terenkripsi (*ciphertext*) dengan *cluster* yang didapat dari data data tidak terekripsi (*plaintext)*.
3) Pengenalan digit pada MNIST. Pengklasifikasi (*classifiers*) pada dataset MNIST dinilai berdasarkan akurasinya, dimana pengelompokan tulisan angka yang benar akan ditunjukan dengan sebuah presentase kebenaran dari gambar angka yang diklasifikasi. Pada penelitian ini, jaringan saraf terenkripsi secara FHE memperoleh akurasi klasifikasi 98,2% pada dataset yng diuji proses dilakukan di jaringan yang terenkripsi. Pada data MNIST, distribusi kelas pada set data pengujian, 10 label digit angka dilakukan pada jumlah gambar yang seimbang. Oleh karena itu, akurasi merupakan indikator kinerja klasifikasi yang dapat dipercaya. Meskipun model pada skema FHE tidak mencapai akurasi yang diklaim 99,21% seperti yang disampaikan pada [28], pengujian pada klasifikasi dan pengenalan digit MNIST terenkripsi dengan akurasi 98,2% dianggap dapat diterima, mengingat pengenalan digit pada CNN terekripsi memiliki keamanan data pada penyimpanan *cloud* yang lebih baik.

B. Runtime

Semua *runtime* yang dilaporkan telah diukur dengan menggunakan HELayers yang berjalan di Docker Engine: 20.10.12 dengan sumber daya 8 core CPU, 8 GB memori, dan 3 GB memori *swap*. Pada percobaan ini, *runtime* diuji menggunakan sistem operasi MacOS Monterey pada Apple M1 MacBook Air dengan RAM 8 GB, CPU 8 core dan GPU 8 core.

Tabel 4 hingga tabel 6 berikut memberikan analisis yang komprehensif membandingkan lama waktu yang dibutuhkan untuk menyelesaikan setiap operasi data. Setiap proses dilakukan sebanyak 5 kali percobaan dan diambil nilai rata-rata dan dihitung standard deviasinya.

TABEL 4
RUNTIME (RATA-RATA DAN STANDARD DEVIASI) PADA *CIPHERTEXT* DAN *PLAINTEXT* UNTUK *PRIVACY-PRESERVING* PENCARIAN DATA NAMA IBUKOTA NEGARA.

Proses	***Runtime (detik)* pada *Ciphertext***	***Runtime (detik)* pada *Plaintext***
Enkripsi database	-	0.780 ± 0.012
Enkripsi query	-	0.004 ± 0.2
Pencarian pada database terenkripsi	42.490 ± 0.01	-
Dekripsi hasil pencarian	0.008 ± 0.03	-

Tabel 4 menunjukkan perhitungan *runtime* pada pencarian nama ibukota negara. Tahap pertama yang dilakukan ialah melakukan enkripsi pada basis data yang berisi data nama negara dan nama ibukota negara tersebut. Enkripsi pada database rata-rata dapat dilakukan dalam 0.78 detik dengan standard deviasi yang cukup kecil yaitu 0.012. Hal ini menunjukkan bahwa tahap enksripsi dilakukan dengan cukup singkat, dan tidak berbeda jauh dalam setiap percobaannya. Tahap selanjutnya ialah melakukan enkripsi terhadap baris perintah (*query*)yang diperlukan untuk melakukan pencarian. Proses enkripsi baris perintah hanya memerlukan waktu sekitar 0.004 detik. Selanjutnya dilakukan pencarian data pada database dan query yang telah dienksipsi (*ciphertext*). Proses pencarian memerlukan waktu sekitar 42 detik. Hasil pencarian masih dalam bentuk terenkripsi (ciphertext), lalu untuk dapat mengetahui hasil pencarian tersebut, dilakukan proses dekripsi yang memerlukan waktu sekitar 0.008 detik.

TABEL 5
RUNTIME (RATA-RATA DAN DAN STANDARD DEVIASI PADA *CIPHERTEXT* DAN *PLAINTEXT* K-MEANS PADA DATA KOSTUMER GROSIR.

Proses	***Runtime (detik)* pada *Ciphertext***	***Runtime (detik)* pada *Plaintext***
Bangun dan load model K-Means	-	0.670 ± 0.02
Enkripsi model	-	4.723 ± 0.03
Enkripsi data uji	-	0.356 ± 0.01
Run clustering	0.212 ± 0.01	-
Dekripsi hasil	0.071 ± 0.03	-

Tabel 5 menunjukkan perhitungan *runtime* pada proses clustering data kostumer grosir. Tahap pertama yang dilakukan ialah membuat dan membangun model K-Means. Tahap ini memerlukan waktu sekitar 0.67 detik. Selanjutnya dilakukan enkripsi terhadapa model yang telah dibangun dimana membutuhkan waktu sekitar 4.723 detik. Enkripsi juga dilakukan terhadap data uji, yaitu sekitar 0.356 detik. Proses *clustering* dilakukan pada model dan data uji yang telah terenkripsi (*ciphertext*) dalam waktu 0.212 detik. Hasil dari *clustering* perlu didekrip agar dapat dibaca oleh pengguna, proses ini dilakukan dalam waktu 0.071 detik.

Tabel 6 menunjukkan proses klasifikasi gambar angka dari dataset MNIST. Tahap awal yang dilakukan adalah mendefinisikan *Neural Network* (NN) dalam waktu sekitar 1420

detik. Waktu ini merupakan training dari 10 epoch yang dilakukan, dimana waktu per epoch ialah rata-rata 141 detik. Selanjutnya NN dan data uji dienkrip masing-masing dalam waktu 19.05 dan 2.12 detik. Proses *inference*, merupakan proses klasifikasi data dilakukan dalam 7.82 detik pada data yang telah dienkripsi (*ciphertext*). Hasil klasifikasi kemudian didekrip dalam waktu sekitar 0.056 detik, dimana rata-rata waktu dekripsi per sampel yaitu 0.004 detik.

TABEL 6
RUNTIME (RATA-RATA DAN STANDARD DEVIASI) PADA *CIPHERTEXT* DAN *PLAINTEXT* UNTUK KLASIFIKASI DIGIT MNIST MENGGUNAKAN CNN.

Proses	***Runtime (detik)* pada *Ciphertext***	***Runtime (detik)* pada *Plaintext***
Bangun Plain NN	-	1420 ± 0.01
Training (1 epoch)	-	141 ± 0.05
Enkripsi Plain NN	-	19.05 ± 0.03
Enkripsi Test Data	-	2.12 ± 0.01
Inference	7.82 ± 0.32	-
Klasifikasi per sample	0.478 ± 0.002	-
Dekripsi hasil	0.056 ± 0.009	-
Dekripsi per sample	0.004 ± 0.023	-

7. KESIMPULAN

Teknik enkripsi homomorfik memberikan pendekatan inovatif untuk masalah dalam menjaga data yang disimpan pada penyimpanan berbasis *cloud*. Ini memungkinkan penyedia layanan *cloud* untuk menawarkan kepada pelanggan dengan cara yang lebih efisien dengan menjaga integritas dan kerahasiaan data terutama pada penerapan aplikasi *e-government*. Sudah banyak penelitian tantang Enkripsi Homomorfik, karena tingkat *noise* dan kompleksitas yang terus meningkat, namun teknik FHE tidak sepenuhnya diadopsi. Dalam penelitian ini, kami telah memberikan pandangan bahwa salah satu skema homomorfik terbaru dan menyajikan studi komparatif dari skema FHE tersebut. Hasil menunjukkan bahwa skema FHE yang digunakan memberikan hasil yang sangat baik dengan *runtime* yang wajar.

REFERENSI

[1] F. Thabit, O. Can, S. Alhomdy, G. H. Al-Gaphari, and S. Jagtap, "A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing," *International Journal of Intelligent Networks*, vol. 3, pp. 16–30, 2022.

[2] A. Jawed Khan and S. Mehfuz, "Big-data driven approaches in materials science on cloud storage API through optimal homomorphic encryption," *Mater Today Proc*, May 2021.

[3] W. Ren *et al.*, "Privacy-preserving using homomorphic encryption in Mobile IoT systems," *Comput Commun*, vol. 165, pp. 105–111, Jan. 2021.

[4] C. Lefebvre, "On data," *Journal of Pidgin and Creole Languages*, vol. 15, no. 2, pp. 313–337, 2000.

[5] C. S. Gu, "Fully homomorphic encryption from approximate ideal lattices," *Ruan Jian Xue Bao/Journal of Software*, vol. 26, no. 10, pp. 2696–2719, 2015.

[6] J. H. Cheon *et al.*, "Batch fully homomorphic encryption over the integers," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 7881 LNCS, pp. 315–335, 2013.

[7] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," *ACM Transactions on Computation Theory*, vol. 6, no. 3, 2014.

[8] S. S. Sathya, P. Vepakomma, R. Raskar, R. Ramachandra, and S. Bhattacharya, "A Review of Homomorphic Encryption Libraries for Secure Computation," pp. 1–12, 2018.

[9] S. Halevi and V. Shoup, "Algorithms in HElib," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8616 LNCS, no. PART 1, pp. 554–571, 2014.

[10] Microsoft Corp., "microsoft/SEAL: Microsoft SEAL is an easy-to-use and powerful homomorphic encryption library.," 2021. https://github.com/microsoft/SEAL

[11] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," *Proceedings of the 15th international conference on Practice and Theory in Public Key Cryptography*, pp. 1–16, 2012.

[12] J. H. Cheon, K. Han, A. Kim, M. Kim, and Y. Song, "Bootstrapping for approximate homomorphic encryption," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 10820 LNCS, pp. 360–384, 2018, doi: 10.1007/978-3-319-78381-9_14.

[13] M. Kumar, M. Shukla, and S. Agarwal, "An E Governance model using cloud computing technology for Developing Countries," no. January, 2013.

[14] S. Hemalatha and R. Manickachezian, "Performance of Ring Based Fully Homomorphic Encryption for securing data in Cloud Computing," vol. 3, no. 11, pp. 8496–8500, 2014.

[15] M. Z. E, Y. Geng, and Y. Geng, "ScienceDirect ScienceDirect ScienceDirect Homomorphic Encryption Technology for Cloud Computing Homomorphic Encryption Technology for Intangible Cloud Computing Research on the Innovation of Protecting Cultural Heritage the Plus " Era Min in E * " Internet," *Procedia Comput Sci*, vol. 154, pp. 73–83, 2019.

[16] B. Thuraisingham, V. Khadilkar, A. Gupta, M. Kantarcioglu, and L. Khan, "Secure Data Storage and Retrieval in the Cloud".

[17] S. Savage and T. C. Clouds, "Hey , you , get off of my cloud Hey , You , Get Off of My Cloud ".

[18] M. Yasuda, T. Shimoyama, and J. Kogure, "Packed Homomorphic Encryption Based on Ideal Lattices and Its Application to Biometrics," pp. 55–74.

[19] M. Tebaa, K. Zkik, and S. El Hajji, "Hybrid homomorphic encryption method for protecting the privacy of banking data in the cloud," *International Journal of Security and its Applications*, vol. 9, no. 6, pp. 61–70, 2015.

[20] R. Bendlin, I. Damgård, C. Orlandi, and S. Zakarias, "Semi-homomorphic encryption and multiparty computation," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6632 LNCS, pp. 169–188, 2011.

[21] "ibmcom/helayers-pylab - Docker Image | Docker Hub."

https://hub.docker.com/r/ibmcom/helayers-pylab
[22] F. Sabir ,"Countries With Capitals _ Kaggle." [Online].Available: https://www.kaggle.com/datasets/faransabir/countries-with-capitals..
[23] D. Dua and C. Graff, "UCI Machine Learning Repository: Data Sets," *Irvine, CA: University of California, School of Information and Computer Science.* 2019.
[24] P. Sharma, "K Means Clustering | K Means Clustering Algorithm in Python," *Analytics Vidhya*. 2019.
[25] Y. LeCun, C. Cortes, and C. J. C. Burges, "MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges." 1998.
[26] N. Dowlin *et al.*, "CryptoNets: Applying neural networks to Encrypted data with high throughput and accuracy - Microsoft research," *Microsoft Research TechReport*, vol. 48, pp. 1–12, 2016.
[27] Stephen Allwright, "What is a good accuracy score in machine learning?," May 14, 2022.
[28] F. Siddique, S. Sakib, and M. A. B. Siddique, "Recognition of handwritten digit using convolutional neural network in python with tensorflow and comparison of performance for various hidden layers," *2019 5th International Conference on Advances in Electrical Engineering, ICAEE 2019*, pp. 541–546, 2019.