OPEN ACCESS

Volume 10 (4), October 2026, 1543-1554

E-ISSN:2580-1643

Jurnal JTIK (Jurnal Teknologi Informasi dan Komunikasi)

DOI: https://doi.org/10.35870/jtik.v10i4.6349

Pengembangan *Service Masking* Data dengan Metode *Scramble* pada Sistem TXX: Studi Kasus PT XYZ

Fani Amanatul Khaliq [1*], Wasis Haryono [2]

[1,2] Program Studi Teknik Informatika, Universitas Pamulang, Kota Tangerang Selatan, Provinsi Banten, Indonesia.*

a r t i c l e i n f o

Article history:
Received 6 February 2026
Received in revised form
3 March 2026
Accepted 25 April 2026
Available online October
2026.

Keywords:
Service Making Data; Metode Scramble; Sistem TXX.

Kata Kunci:
Service Making Data; Metode Scramble; Sistem TXX.

a b s t r a c t

The advancement of digitalization in financial institutions demands secure system development practices, particularly regarding the use of customer data in development and UAT environments. In the TXX core system, the use of raw data without sanitization poses a potential risk of sensitive information leaks. This study aims to develop a data sanitization service using a scrambling method capable of scrambling personal data without altering the structure, format, or integrity of the testing process. The research method employs a research and development approach, comprising system analysis, database and service flow design, implementation of a TAFC-based prototype, and functional and structural testing. The implementation resulted in three main modules: Parameter Scramble, Agent Service Scramble, and Routine Service, which are integrated with PT XYZ's development environment. Black-box testing results indicate that all interface functions and service processes operate as expected. White-box testing of the program structure ensured that the internal logic, flow control, and multithreading mechanisms operated stably without race conditions. Additionally, usability testing using the System Usability Scale on 14 respondents yielded a score of 77, which falls into the "Good" and acceptable category.

a b s t r a k

Perkembangan digitalisasi pada lembaga keuangan menuntut adanya praktik pengembangan sistem yang aman, khususnya dalam penggunaan data customer pada environment development dan UAT. Pada core system TXX, penggunaan data asli tanpa sanitasi berpotensi menimbulkan kebocoran informasi sensitif. Penelitian ini bertujuan mengembangkan layanan sanitasi data menggunakan metode scramble yang mampu mengacak data pribadi tanpa mengubah struktur, format, maupun integritas proses pengujian. Metode penelitian menggunakan pendekatan research and development dengan tahapan analisis sistem, perancangan basis data dan alur layanan, implementasi prototipe berbasis TAFC, serta pengujian fungsional dan struktural. Implementasi menghasilkan tiga modul utama, yaitu Parameter Scramble, Agent Service Scramble, dan Routine Service yang terintegrasi dengan environment development PT XYZ. Hasil blackbox testing menunjukkan seluruh fungsi antarmuka dan proses layanan berjalan sesuai yang diharapkan. Pengujian whitebox pada struktur program memastikan logika internal, kontrol alur, dan mekanisme multithread bekerja stabil tanpa race condition. Selain itu, usability testing menggunakan System Usability Scale terhadap 14 responden menunjukkan skor 77, yang termasuk kategori "Good" dan acceptable.

Corresponding Author. Email: faniamanatul@gmail.com [1].

1. Pendahuluan

Perkembangan teknologi informasi yang pesat telah menjadi pendorong utama bagi institusi keuangan untuk bertransformasi ke arah digital. Hal ini sangat penting, terutama bagi lembaga keuangan yang harus mendukung layanan operasional dan transaksi pelanggan secara efisien dan aman. Salah satu sistem inti yang banyak digunakan oleh lembaga keuangan modern adalah sistem *TXX*. Sistem ini dirancang untuk mengelola berbagai aspek operasi perbankan, mulai dari pembukaan rekening hingga transaksi dan manajemen pelanggan secara real-time. Namun, di balik kompleksitas sistem ini, terdapat tanggung jawab besar yang harus diemban oleh lembaga-lembaga tersebut, yaitu menjaga keamanan data pelanggan yang bersifat rahasia dan sensitif. Dalam praktik pengembangan perangkat lunak, khususnya pada tahapan pengembangan dan pengujian sistem, sering kali digunakan data asli pelanggan untuk menguji validitas proses transaksi dan alur bisnis. Penggunaan data asli ini, meskipun diperlukan untuk memastikan keakuratan dan efektivitas sistem, dapat menimbulkan potensi pelanggaran terhadap prinsip perlindungan data pribadi jika tidak disertai dengan metode perlindungan yang memadai (Almasri & Mahmoud, 2008).

Sanitasi data menjadi aspek yang sangat penting dalam konteks ini. Sanitasi data merupakan proses memodifikasi atau menyamarkan informasi sensitif agar tetap berguna namun tidak menimbulkan risiko keamanan. Metode yang umum digunakan dalam sanitasi data meliputi *masking*, *anonymization*, *pseudonymization*, *tokenization*, dan *scrambling*. Metode *scramble*, misalnya, adalah teknik sanitasi yang mengacak karakter atau struktur data sehingga tidak dapat dibaca oleh pihak yang tidak berwenang. Teknik ini tidak bersifat kriptografis dan biasanya tidak dapat dibalik, serta memiliki proses yang cepat dan menjaga panjang data, sehingga cocok untuk kebutuhan non-produksi seperti pengujian (Gupta *et al.*, 2022). Dalam perancangan sistem informasi, pemodelan basis data juga memegang peranan penting. *Entity Relationship Diagram* (ERD) adalah salah satu metode yang digunakan untuk memodelkan basis data. ERD berfungsi sebagai alat bantu dalam pembuatan database dan memberikan gambaran tentang bagaimana kerja database yang akan dibuat (Pulungan *et al.*, 2023). Selain itu, *Logical Record Structure* (LRS) adalah representasi logis yang menggambarkan bagaimana data disusun dan diakses pada tingkat logis tanpa memperhatikan implementasi fisiknya. Menurut Pratama *et al.* (2020), LRS merupakan transformasi dari penggambaran ERD dalam bentuk yang lebih jelas dan mudah dipahami. *Unified Modeling Language* (UML) juga menjadi alat yang penting dalam pengembangan perangkat lunak. UML adalah bahasa standar untuk memvisualisasikan, menspesifikasikan, dan mendokumentasikan sistem perangkat lunak. Dikenal luas di kalangan pengembang perangkat lunak, UML membantu dalam merancang sistem yang kompleks dan memastikan bahwa semua elemen sistem dapat terintegrasi dengan baik (Barjakly *et al.*, 2021). Dengan demikian, pemahaman yang mendalam tentang sanitasi data, pemodelan basis data, dan penggunaan alat pemodelan seperti UML sangat penting bagi lembaga keuangan dalam rangka menjaga keamanan data pelanggan dan mematuhi regulasi yang berlaku.

2. Metode Penelitian

Penelitian ini menggunakan metode *research and development* dengan pendekatan rekayasa perangkat lunak untuk merancang dan membangun layanan sanitasi data pada *Core System TXX* berbasis framework *TAFC*. Tahap pertama yang dilakukan adalah analisis sistem, yang mencakup identifikasi proses bisnis pada sistem yang sedang berjalan, pemetaan alur pemrosesan data pelanggan, serta evaluasi penggunaan data produksi pada lingkungan pengembangan dan pengujian. Analisis ini bertujuan untuk menemukan permasalahan utama, yaitu tidak adanya proses sanitasi data, yang berpotensi menimbulkan kebocoran data pribadi. Hasil dari analisis tersebut digunakan sebagai dasar untuk merumuskan kebutuhan fungsional dan non-fungsional dari sistem sanitasi data yang diusulkan. Tahap berikutnya adalah perancangan sistem, yang dirancang agar dapat diintegrasikan dengan lingkungan pengembangan PT XYZ. Lingkungan ini terdiri dari *Oracle Database*, bahasa pemrograman *jBC*, serta spesifikasi perangkat lunak yang digunakan, termasuk *Windows OS*, text editor *Notepad++* dengan plugin *jBC*, dan terminal emulator *TeraTerm*.

3. Hasil dan Pembahasan

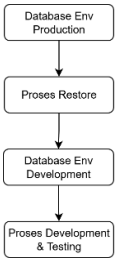
Hasil

Perancangan Sistem

Perancangan sistem dilakukan sebagai tahap awal sebelum implementasi untuk menghasilkan desain aplikasi yang terstruktur dan terencana dengan baik. Pada tahap ini, analisis sistem berjalan dilakukan untuk memahami bagaimana sistem saat ini beroperasi.

Analisis Sistem Berjalan

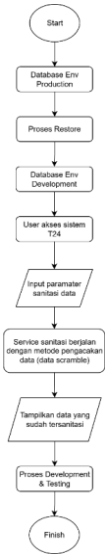
Pada sistem yang sedang berjalan saat ini, proses pengolahan data pelanggan dilakukan secara langsung di lingkungan produksi. Namun, dalam beberapa aktivitas tertentu, seperti pengembangan fitur baru, pengujian sistem (*system testing*), pelatihan pengguna (*user training*), atau penerapan fitur ke lingkungan staging atau UAT, salinan data produksi digunakan tanpa adanya proses sanitasi atau pengaburan data. Penggunaan data asli dalam konteks ini dapat meningkatkan risiko kebocoran informasi sensitif, yang dapat berpotensi melanggar prinsip perlindungan data pribadi. Oleh karena itu, penting untuk mengidentifikasi dan mengatasi kelemahan ini dalam perancangan sistem yang baru.



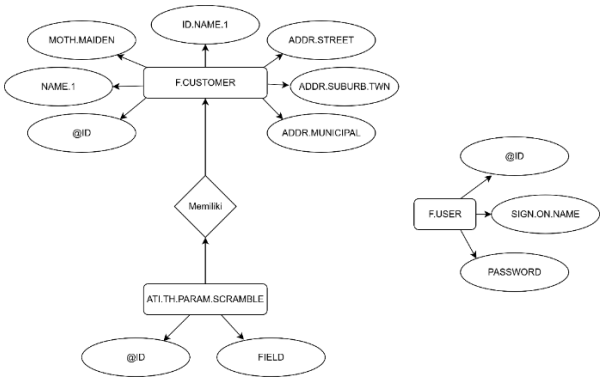
Gambar 1. Alur Sistem Berjalan

Analisis Sistem Usulan

Sistem usulan ini dirancang untuk menghadirkan mekanisme layanan sanitasi data yang memanfaatkan metode *scramble* di dalam aplikasi *Core System TXX* berbasis *TAFC*. Inti dari pengembangan sistem ini adalah pengguna akan mendefinisikan parameter yang berisi cakupan data pelanggan yang akan disanitasi. Parameter ini nantinya akan dibaca oleh layanan sanitasi data untuk memastikan bahwa data yang sensitif dapat dikelola dengan aman tanpa mengubah struktur dan format aslinya. Dengan pendekatan ini, diharapkan proses sanitasi data dapat dilakukan secara efisien dan efektif, mengurangi risiko kebocoran informasi sensitif, serta memastikan bahwa data yang digunakan dalam lingkungan pengembangan dan pengujian tetap aman. Selain itu, sistem ini juga akan dilengkapi dengan antarmuka yang memudahkan pengguna dalam mendefinisikan parameter sanitasi, sehingga meningkatkan pengalaman pengguna dalam menggunakan aplikasi.



Gambar 2. Alur Sistem Usulan



Gambar 3. Entity Relationship Diagram

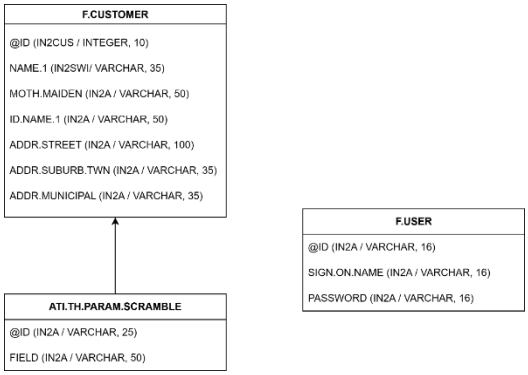
Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) merupakan model konseptual yang digunakan untuk menggambarkan struktur basis data melalui representasi entitas, atribut, serta relasi antar entitas. Model ini memudahkan perancangan dan pemahaman organisasi data dalam suatu sistem. Pada ERD yang ditampilkan, terdapat tiga entitas utama, yaitu *F.CUSTOMER*, *F.USER*, dan *ATI.TH.PARAM.SCRAMBLE*. Entitas *F.CUSTOMER* menyimpan atribut data pribadi seperti *NAME.1*, *MOTH.MAIDEN*, *ID.NAME.1*,

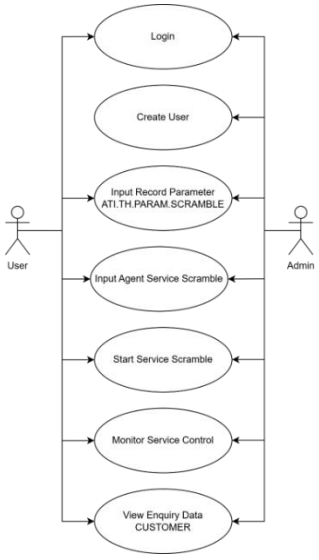
serta atribut alamat. Entitas *F.USER* menyimpan data pengguna sistem, termasuk *SIGN.ON.NAME* dan *PASSWORD*. Sementara itu, entitas *ATI.TH.PARAM.SCRAMBLE* menyimpan parameter field yang akan disanitasi. Ketiga entitas tersebut dihubungkan melalui relasi "Memiliki," yang menunjukkan keterkaitan antara data pelanggan, pengguna, dan parameter layanan dalam proses sanitasi data.

Logical Record Structure **(LRS)**

Dalam proses pengolahan data, penulis memanfaatkan ERD untuk memodelkan hubungan antar tabel pada basis data. Hasil rancangan tersebut kemudian diterjemahkan ke dalam bentuk *Logical Record Structure* (LRS) dan dilanjutkan dengan penyusunan spesifikasi setiap tabel secara lebih detail. Tahapan ini divisualisasikan pada gambar berikut:

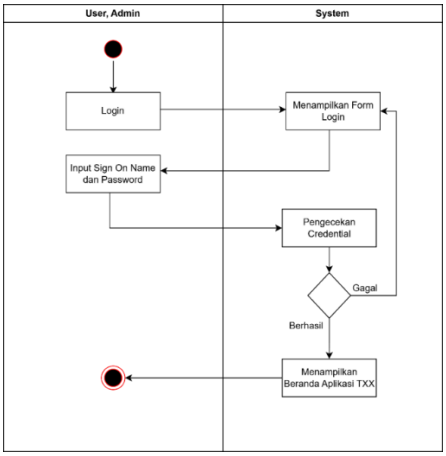


Gambar 4. Logical Record Structure (LRS)



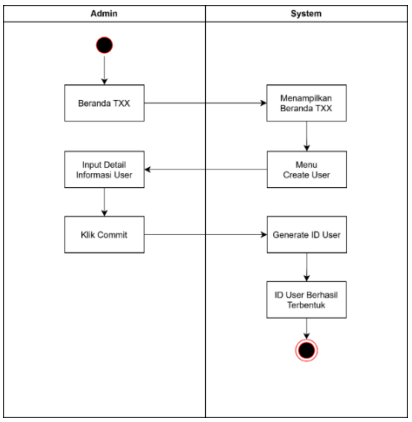
Gambar 5. Use Case Diagram

Use case diagram merupakan diagram UML yang digunakan untuk menggambarkan fungsi utama sistem dan interaksi antara aktor dengan sistem secara umum. Pada diagram yang ditampilkan, alur dimulai dari proses Login, dilanjutkan dengan *Create User*, kemudian pengguna mengatur parameter melalui Input *Record Parameter* ATI.TH.PARAM.SCRAMBLE dan Input *Agent Service Scramble*, menjalankan proses melalui *Start Service Scramble,* memantau eksekusi melalui *Monitor Service Control*, dan pada tahap akhir melakukan *View Enquiry Data CUSTOMER* untuk melihat hasil proses sanitasi data.



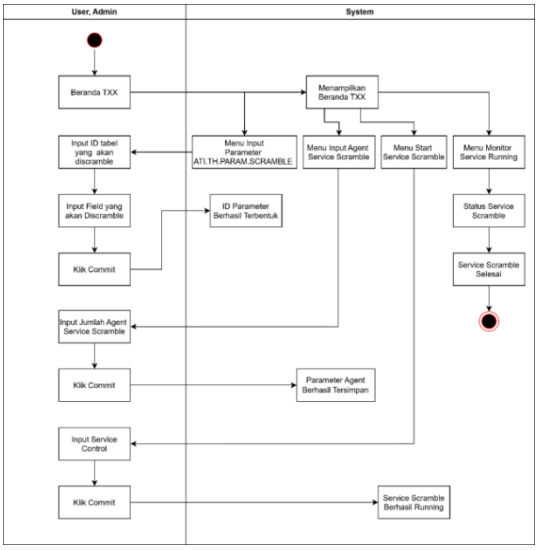
Gambar 6. Activity Diagram Login

Activity diagram login menjelaskan alur proses autentikasi ketika user akan mengakses aplikasi *TXX*. Proses dimulai ketika user memasukkan *sign on name* dan *password* yang sebelumnya telah dibuat oleh *admin*. Sistem kemudian melakukan proses validasi terhadap *credential* yang *diinput* dengan data yang tersimpan di *database*. Apabila *sign on name* atau *password* tidak sesuai, sistem akan menolak akses dan user tidak dapat masuk ke dalam aplikasi. Sebaliknya, jika data yang dimasukkan valid, sistem akan mengizinkan user masuk dan menampilkan halaman beranda aplikasi *TXX*.



Gambar 7. Activity Diagram Create User

Activity diagram create user menggambarkan proses pembuatan akun baru yang hanya dapat dilakukan oleh *admin* melalui menu *create user* di halaman beranda *admin*. Pada proses ini, *admin* menginput detail informasi user seperti identitas dan *credential* yang diperlukan, kemudian menekan tombol *commit* untuk menyimpan data. Sistem selanjutnya akan melakukan proses *generate user id* beserta *credential* sesuai dengan data yang *diinput* dan menyimpannya ke dalam *database*. Menu *create user* bersifat terbatas karena hanya muncul dan dapat diakses oleh *admin* sebagai bagian dari kontrol manajemen pengguna.

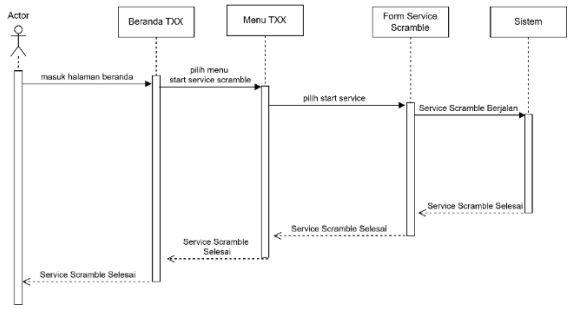


Gambar 8. Activity Diagram Input Record Parameter ATI.TH.PARAM.SCRAMBLE, Agent Service Scramble, Service Scramble, Monitor Service Scramble

Activity diagram ini menjelaskan rangkaian proses utama dalam pelaksanaan *service scramble* data di aplikasi *TXX* yang melibatkan beberapa menu, yaitu *input parameter scramble, input agent service, start service, dan monitor service*. *User* atau *admin* terlebih dahulu mendefinisikan parameter tabel dan kolom yang akan diproses melalui menu ATI.TH.PARAM.SCRAMBLE, kemudian menentukan jumlah *agent* yang digunakan pada menu *input agent service*. Setelah parameter siap, *service* dijalankan melalui menu *start service* dengan mengubah status *service control* menjadi *start*. Selama proses berjalan, *user* dan *admin* dapat memantau status eksekusi *service* melalui menu *monitor service running* hingga seluruh data selesai diproses dan status *service* berubah menjadi selesai.

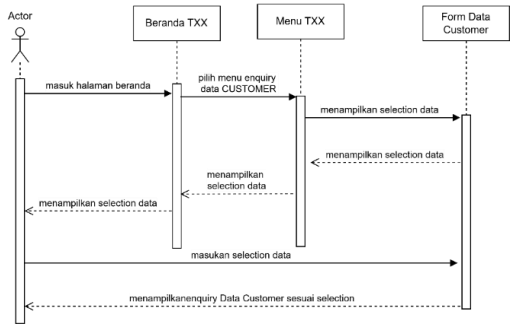
Gambar 9. Activity Diagram View Enquiry Data CUSOMTER

Activity diagram view enquiry data *CUSTOMER* menjelaskan alur proses penampilan dan verifikasi data pada tabel *CUSTOMER* oleh *user* dan *admin*. Menu ini digunakan untuk membandingkan kondisi data sebelum dan sesudah *service scramble* dijalankan guna memastikan proses berjalan dengan normal. *User* dan *admin* dapat memasukkan kriteria seleksi data yang ingin ditampilkan, kemudian menekan tombol *find* untuk menampilkan hasil *enquiry*. Data yang ditampilkan menjadi dasar evaluasi keberhasilan proses *scramble* dan validasi integritas data dalam sistem.



Gambar 10. Sequence Diagram Service Scramble

Sequence Diagram Service Scramble pada Gambar 10 menjelaskan alur interaksi antara user, antarmuka aplikasi, dan sistem ketika proses *service scramble* dijalankan. Proses dimulai ketika *user* berada di beranda *TXX*, kemudian memilih menu *start service scramble* pada menu TXX sehingga sistem menampilkan *form service scramble*. Setelah *user* memilih dan mengeksekusi perintah *start service*, sistem mulai menjalankan proses *service scramble* secara otomatis. Selama proses berlangsung, sistem mengeksekusi mekanisme *scramble* data hingga seluruh tahapan selesai diproses. Setelah eksekusi berakhir, sistem mengirimkan status bahwa *service scramble* telah selesai, yang kemudian ditampilkan kembali ke antarmuka sebagai informasi bahwa proses telah berhasil dijalankan.



Gambar 11. Sequenece Diagram Enquiry Data CUSTOMER

Sequence Diagram Monitor Service Control pada gambar tersebut menjelaskan alur interaksi antara *user* atau *admin* dengan sistem ketika melakukan pemantauan data melalui menu *enquiry CUSTOMER*. Proses dimulai saat *user* atau *admin* masuk ke halaman beranda *TXX*, kemudian memilih menu *enquiry data CUSTOMER* pada menu *TXX*. Sistem selanjutnya menampilkan *form* untuk memasukkan kriteria atau *selection data* yang diinginkan. Setelah *user* atau *admin menginput selection data*, sistem memproses permintaan tersebut dan menampilkan hasil *enquiry data CUSTOMER* sesuai dengan kriteria yang dipilih. Hasil tampilan ini digunakan oleh *user* atau *admin* untuk memantau kondisi data serta melakukan verifikasi terhadap proses *service* yang sedang atau telah berjalan.

Implementasi Sistem

Implementasi sistem dilakukan dengan mengembangkan *service* sanitasi data berbasis metode *scramble* pada *core system TXX*. Sistem diintegrasikan dengan lingkungan pengembangan *PT XYZ*, yang terdiri dari *Oracle Database*, bahasa pemrograman *jBC*, dan kernel layanan berbasis *multithread* pada *TAFC*. Spesifikasi perangkat lunak yang digunakan meliputi *Windows OS*, *text editor Notepad++* dengan *plugin jBC*, dan *terminal emulator TeraTerm*. Adapun kebutuhan perangkat keras mencakup laptop pengembangan serta *server development internal PT XYZ* yang menjadi target eksekusi layanan. Spesifikasi perangkat keras yang digunakan ditunjukkan pada Tabel 1 dan telah memenuhi kebutuhan proses *development* dan *background processing*.

Tabel 1. Spesifikasi Perangkat Keras Implementasi Sistem

Komponen	Spesifikasi
Laptop	Asus Zenbook UX481FA
Prosesor	Intel® Core™ i5-10210U
RAM	8 GB
Storage	512 GB SSD NVMe
Server Development	Server PT XYZ

Implementasi Program

Implementasi program dilakukan dengan mengembangkan tiga bagian utama: *Parameter Service Scramble*, *Agent Service Scramble*, dan *Routine Service (LOAD, SELECT, MAIN, COMMON)*. Pada modul *parameter*, pengguna menetapkan daftar *field* *CUSTOMER* yang akan disanitasi. Tabel 2 berikut menunjukkan contoh enam *field* yang *diinput* sebagai parameter layanan berdasarkan halaman *ATI.TH.PARAM.SCRAMBLE*. Daftar *field* tersebut diringkas pada tabel berikut:

Tabel 2. Parameter Field Scramble CUSTOMER

Field Table.x	Keterangan
NAME.1	Nama lengkap tanpa singkatan
MOTH.MAIDEN	Nama gadis ibu kandung
ID.NAME.1	Nama pada identitas
ADDR.STREET	Alamat jalan
ADDR.SUBURB.TWN	Kelurahan
ADDR.MUNICIPAL	Kecamatan

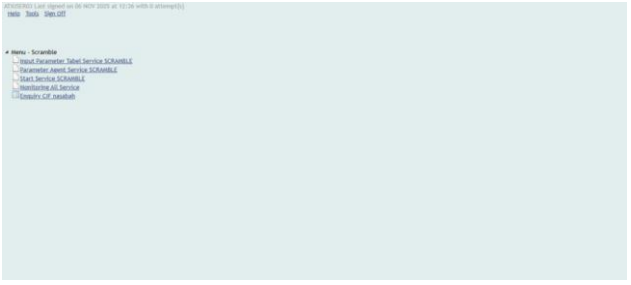
Setelah parameter ditetapkan, pengguna memasukkan jumlah *agent* yang akan dipakai oleh *service*. Konfigurasi *multithread* ini memungkinkan proses *scramble* dilakukan paralel sehingga kinerja sistem meningkat secara signifikan. Proses eksekusi dimulai melalui halaman *Start Service Scramble*, di mana pengguna mengubah kendali proses dari *STOP* menjadi *START*, sebelum sistem menjalankan *routine* secara otomatis pada *backend*. Hasil keluaran dapat diverifikasi melalui halaman *Enquiry Data CUSTOMER*. Contoh implementasi tampilan ringkasan *CUSTOMER* ditunjukkan pada Gambar 12.

Gambar 12. Halaman Enquiry Data Customer

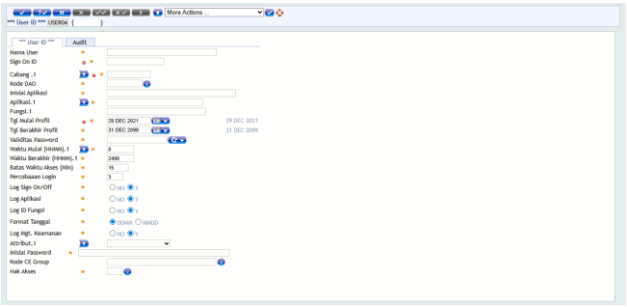
Dokumentasi Tampilan Sistem

Dokumentasi tampilan sistem berisi kumpulan tangkapan layar (*screenshot*) dari antarmuka aplikasi yang digunakan dalam penelitian ini. Dokumentasi ini bertujuan untuk memberikan gambaran visual mengenai bentuk antarmuka, struktur menu, serta alur penggunaan sistem dalam mendukung proses sanitasi data. Setiap tampilan yang disajikan mewakili fungsi utama sistem, mulai dari proses *login*, pengelolaan *user*, pengaturan *parameter scramble*, eksekusi *service*, hingga proses *monitoring* dan verifikasi hasil. Dengan adanya dokumentasi ini, pembaca diharapkan dapat memahami cara kerja sistem secara lebih jelas melalui ilustrasi visual yang ditampilkan.

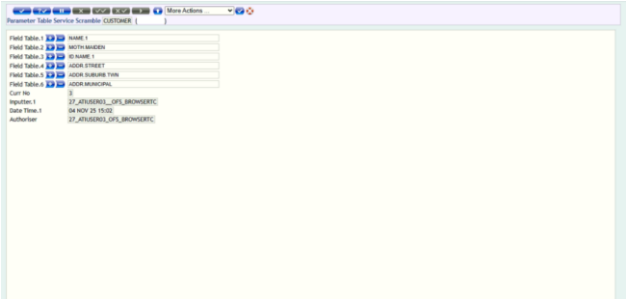
Gambar 13. Halaman Login



Gambar 14. Halaman Beranda User



Gambar 15. Halaman Create User

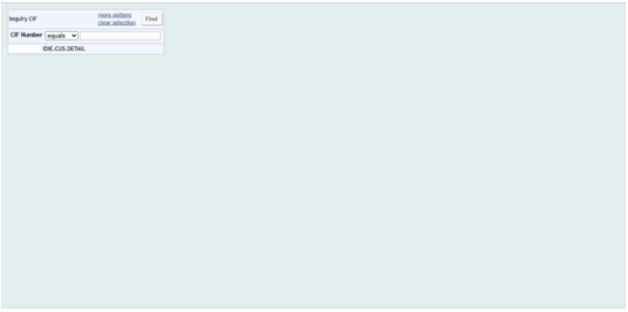


Gambar 16. Halaman Parameter ATI.TH.PARAM. SCRAMBLE

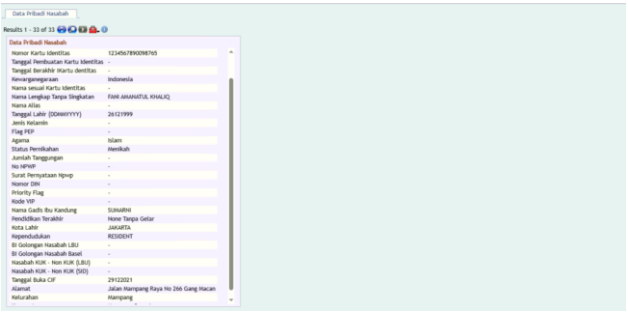
Gambar 17. Halaman Agent Service Scramble

Gambar 18. Halaman Start Service Scramble

Gambar 19. Halaman Monitor Service Control



Gambar 20. Halaman Enquiry Data Customer



Gambar 21. Halaman Enquiry Data Customer

Pengujian Sistem

Pengujian dilakukan untuk memastikan fungsionalitas antarmuka, logika program internal, kompatibilitas, dan aspek *usability*. Dua metode utama yang digunakan ialah *Black Box Testing* dan *White Box Testing*, dilengkapi dengan *compatibility test* melalui *System Usability Scale (SUS). Black box testing* dilakukan pada seluruh fungsi utama, termasuk *login*, menu *user/ admin*, *input parameter*, konfigurasi *agent*, start *service*, *monitoring*, serta verifikasi data melalui *Enquiry CUSTOMER*. Hasil pengujian menunjukkan seluruh skenario menghasilkan keluaran sesuai harapan. Hasil pengujian ditunjukkan pada Tabel 3.

Tabel 3. Contoh Hasil Pengujian Black Box pada Halaman Parameter Scramble

Skenario	Step Pengujian	Hasil yang Diharapkan	Hasil
Input parameter CUSTOMER	Input ID tabel dan field	Parameter tersimpan	Passed
Konfigurasi agent	Input jumlah agent = 5	Parameter tersimpan	Passed
Start service	Kendali START → commit	Service berjalan	Passed
Enquiry hasil	Tampilkan CUSTOMER tersanitasi	Data berubah sesuai aturan scramble	Passed

Gambar 22. Halaman Menu Scramble

Gambar di atas memperlihatkan halaman Parameter *SCRAMBLE* yang digunakan pada tahap pengujian.

Secara keseluruhan, seluruh fungsi inti *PASS* tanpa kegagalan. Hal ini menunjukkan bahwa interaksi pengguna dengan sistem telah berjalan stabil. Kemudian *white box testing* yang dilakukan pada empat routine utama: *.LOAD, .SELECT, MAIN, dan .COMMON*. Pengujian mencakup validasi logika internal, struktur kontrol, alur *multithreading*, serta pemeriksaan *error handling*. Pada *routine* .LOAD, bertanggung jawab memuat struktur tabel dan inisialisasi variabel.

Tabel 4. Hasil Pengujian Routine .LOAD

Kasus Uji	Deskripsi	Hasil
TC1	Validasi deklarasi variabel	Semua variabel benar
TC2	Inisialisasi nilai awal	Berhasil tanpa error
TC3	Tabel tidak ditemukan	Log error muncul

Kemudian pada pengujian *routine* .SELECT, memilih *record CUSTOMER* yang akan diproses serta membentuk *control list*.

Tabel 5. Hasil Pengujian Routine .SELECT

Kasus Uji	Deskripsi	Hasil
TC1	Data ditemukan	Array hasil terbentuk
TC2	Data kosong	Return array kosong
TC3	Pembentukan control.list	Control sesuai kriteria
TC4	control.list kosong	Program berhenti aman

Selanjutnya pada pengujian *routine* MAIN, menjalankan algoritma *scramble* dan memproses *multithread*.

Tabel 6. Hasil Pengujian Routine .MAIN

Kasus Uji	Deskripsi	Hasil
TC1	Scramble 1 thread	Record berubah sesuai algoritma
TC2	Scramble 5 thread	Tidak ada race condition
TC3	Simulasi 1 thread gagal	Sistem melanjutkan thread lain dan mencatat error
TC4	Scramble field tertentu	Hanya field terpilih yang berubah

Tabel 7. Hasil Pengujian Routine .COMMON

Kasus Uji	Deskripsi	Hasil
TC1	Akses variabel global	Semua routine dapat membaca
TC2	Modifikasi variabel	Nilai berubah di semua routine
TC3	Dependency check	Tidak ada konflik nilai

Terakhir pada *Usability Testing (SUS),* sebanyak 14 responden melakukan penilaian *usability*. Perhitungan *SUS* menghasilkan skor 77, sebagaimana ditunjukkan pada hasil berikut.

Tabel 8. Ringkasan Skor SUS

Jumlah Responden	Skor Rata-Rata SUS	Kategori
14	77	Good (Acceptable)

Untuk penelitian ini masuk dalam kategori *GOOD*. Artinya secara *usability* termasuk layak atau dapat diterima.

Pembahasan

Penelitian ini berfokus pada pengembangan layanan sanitasi data menggunakan metode *scramble* dalam *Core System TXX* berbasis *TAFC*. Dengan meningkatnya kebutuhan akan keamanan data, terutama di lembaga keuangan, penting untuk memastikan bahwa data pelanggan yang sensitif dapat dikelola dengan aman. Melalui analisis sistem yang berjalan, ditemukan bahwa penggunaan data asli dalam lingkungan produksi tanpa proses sanitasi dapat meningkatkan risiko kebocoran informasi pribadi. Hal ini sejalan dengan temuan Almasri dan Mahmoud (2008), yang menekankan perlunya praktik pengembangan sistem yang aman dalam pengelolaan data pelanggan. Sistem usulan yang dikembangkan dalam penelitian ini dirancang untuk memberikan mekanisme sanitasi data yang efisien. Pengguna dapat mendefinisikan parameter yang mencakup data pelanggan yang akan disanitasi, sehingga proses sanitasi dapat dilakukan dengan baik tanpa mengubah struktur dan format asli data. Ini sejalan dengan konsep yang dijelaskan oleh Gupta *et al.* (2022) mengenai pentingnya teknik sanitasi seperti *scrambling* yang dapat melindungi data sensitif tanpa mengurangi fungsionalitasnya. Lebih lanjut, penggunaan *Entity Relationship Diagram* (ERD) dalam perancangan sistem memudahkan pemodelan hubungan antar tabel dalam basis data. Hal ini mendukung pemahaman yang lebih baik tentang organisasi data, sebagaimana diungkapkan oleh Pulungan *et al.* (2023) yang menunjukkan bahwa ERD berfungsi sebagai alat bantu yang efektif dalam pembuatan database. Dengan demikian, penelitian ini tidak hanya memberikan solusi praktis untuk masalah sanitasi data, tetapi juga menyajikan pendekatan yang sistematis dalam perancangan dan pengelolaan basis data yang aman dan efisien, sesuai dengan prinsip-prinsip yang telah ada dalam literatur sebelumnya.

4. Kesimpulan

Berdasarkan penelitian dan implementasi pengembangan layanan sanitasi data dengan metode *scramble* pada sistem TXX studi kasus PT XYZ, dapat disimpulkan bahwa layanan sanitasi data dapat terintegrasi dengan *core system* TXX untuk menjaga kerahasiaan data nasabah dalam lingkungan pengembangan dan UAT. Metode *scramble* dapat diterapkan sebagai teknik utama dalam proses sanitasi data, yang memungkinkan pengacakan data sensitif tanpa mengubah struktur dan format aslinya. Proses dan hasil dari layanan sanitasi data yang dikembangkan tidak berpengaruh pada pengujian sistem secara fungsional, tanpa mengganggu skenario bisnis yang ada. Hasil pengujian black box menunjukkan bahwa seluruh fungsi antarmuka dan proses layanan berjalan sesuai yang diharapkan, sementara pengujian white box pada struktur program memastikan bahwa logika internal, kontrol alur, dan mekanisme multithread bekerja stabil tanpa kondisi

balapan. Selain itu, pengujian kegunaan menggunakan System Usability Scale terhadap 14 responden menunjukkan skor 77, yang termasuk dalam kategori "Baik" dan dapat diterima. Meskipun demikian, layanan yang dikembangkan dalam penelitian ini masih memiliki beberapa kekurangan dan keterbatasan. Oleh karena itu, ada beberapa hal yang perlu dikembangkan agar menjadi lebih baik, antara lain menambahkan parameter field "formula rnd" agar formula dapat terus diperbarui secara berkala dan menjaga algoritma pengacakan tetap terupdate, menambahkan parameter field "flag processed" agar pengguna dapat memilah parameter tabel yang akan diproses, serta menambahkan beberapa validasi dasar pada parameter untuk mengurangi kesalahan manusia saat pengisian nilai parameter. Selain itu, juga perlu dibuat dokumen panduan penggunaan layanan *scramble* untuk mempercepat proses pelatihan bagi pengguna.

5. Daftar Pustaka

Almasri, E., & Mahmoud, Q. H. (2008). Investigating Web Services on Mobile Devices. *Journal of Systems and Software, 81*(3), 420–431.

Aziz, M., & Haron, S. (2020). Internet Banking of Islamic Banks: Issues of Security and Privacy. *Journal of Islamic Finance, 5*(1), 67–75.

Belgian Data Protection Authority. (2020). Recommendation on data sanitisation and data medium destruction techniques: Information media erasure and destruction guide (Version 1.01).

Direktorat Jenderal Perbendaharaan. (2025). Mengenal ISO/IEC 27001: Standar emas keamanan informasi. Kementerian Keuangan Republik Indonesia.

Duan, X., Shao, Z., Wang, W., Zhang, E., Yue, D., Qin, C., & Nam, H. (2022). A Steganography Model Data Protection Method Based on Scrambling Encryption. *Computers, Materials & Continua, 72*(3), 5363–5375. https://doi.org/10.32604/cmc.2022.027807.

Fitria, A. D. (2021). Implementasi API Sanitasi Data pada Sistem Perbankan Syariah. *Jurnal Teknologi Informasi dan Komputer, 9*(2), 33–40.

Gupta, A., Kumar, R., & Singh, M. (2022). Data Protection Using Scrambling Technique. *International Journal of Computer Applications, 183*(3), 10–15.

Hutapea, S. (2020). Penerapan Software Testing Pada Aplikasi Properti PT. X dengan Metode Regresi & Black Box.

Koç, H., Erdoğan, A. M., Barjakly, Y., & Peker, S. (2021). UML Diagrams in Software Engineering Research: A Systematic Literature Review. *Proceedings, 74*(1), 13. https://doi.org/10.3390/proceedings2021074013.

Nababan, V. L. Y. B., Wulandari, A., & Karlina, L. (2024). PERLINDUNGAN DATA PRIBADI NASABAH DI INDUSTRI PERBANKAN. *SYARIAH: Jurnal Ilmu Hukum, 1*(4), 234-240.

Neagu, M.-I., & Miclea, L. (2014). Data scrambling in memories: A security measure. In *2014 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)* (pp. 1–6).

Pooja Badgujar. (2021). Implementing Data Masking Techniques for Privacy Protection. *Journal of Technological Innovations, 2*(4). https://doi.org/10.93153/5yysvh44.

Pulungan, S. M., Febrianti, R., Lestari, T., Gurning, N., & Fitriana, N. (2023). Analisis Teknik Entity-Relationship Diagram Dalam Perancangan Database. *Jurnal Ekonomi Manajemen Dan Bisnis (JEMB), 1*(2), 143–147. https://doi.org/10.47233/jemb.v1i2.533.

Sa'adah, N., Astawa, I. G. P., & Sudarsono, A. (2018). Trusted Data Transmission Using Data Scrambling Security Method with Asymmetric Key Algorithm for Synchronization. *EMITTER International Journal of Engineering Technology, 6*(2), 217–235. https://doi.org/10.24003/emitter.v6i2.267.

Setiawan, G. W. (2021). PENGUJIAN PERANGKAT LUNAK MENGGUNAKAN METODE BLACK BOX: Studi Kasus EXELSA Universitas Sanata Dharma.

Singh, S., & Rai, R. K. (2014). A review report on security threats on database. *International Journal of Computer Science and Information Technologies, 5*(3), 3215–3219.

Subashini, R., & Kavitha, V. (2019). Data Sanitization Techniques: Protecting Against Data Leakage. *Journal of Cyber Security and Information Systems, 7*(4), 55–63.

Vijay Sai, R., & Saravanan, S. (2018). A Review on Security in Cache Memories. *Indian Journal of Science and Technology, 9*(48), 1–6. https://doi.org/10.17485/ijst/2016/v9i48/96037.

Vijay Sai, R., Saravanan, S., & Anandkumar, V. (2015). Implementation of a Novel Data Scrambling based Security Measure in Memories for VLSI Circuits. *Indian Journal of Science and Technology, 8*(35), 1–6. https://doi.org/10.17485/ijst/2015/v8i35/86798.

Wali, R. R., Permana, A. A., & Prasetyo, D. H. (2020). Logical Record Structure (LRS) dalam Perancangan Basis Data. *JISICOM (Journal of Information System, Informatics and Computing), 4*(1), 45–52.