IMPLEMENTASI ALGORITMA *HYBRID MATCHING* DALAM PENGEMBANGAN APLIKASI REKONSILIASI

TUGAS AKHIR

Diajukan Oleh:

SONIA SABIRA GAYOARA

NIM : 220705033

Mahasiswa Fakultas Sains dan Teknologi
Prodi Teknologi Informasi

FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI AR-RANIRY
DARUSSALAM – BANDA ACEH
2026 M / 1447 H

LEMBAR PERSETUJUAN

IMPLEMENTASI ALGORITMA *HYBRID MATCHING* DALAM PENGEMBANGAN APLIKASI REKONSILIASI

TUGAS AKHIR

Diajukan Kepada Fakultas Sains dan Teknologi
Universitas Islam Negeri (UIN) Ar-Raniry Banda Aceh
Sebagai Salah Satu Beban Studi Memperoleh Gelar Sarjana (S1)
dalam Ilmu Teknologi Informasi

Oleh:
SONIA SABIRA GAYOARA
NIM. 220705033

Mahasiswa Fakultas Sains dan Teknologi
Program Studi Teknologi Informasi

Disetujui untuk Dimunaqasyahkan Oleh:

Pembimbing I,	Pembimbing II,
Khairan AR, M.Kom	**Ridha Ilahi, M.T**
NIP. 198607042114031001	**NIP. 197905302014031001**

Mengetahui,
Ketua Program Studi Teknologi Informasi

Malahayati, M.T.
NIP. 198301272015032003

LEMBAR PENGESAHAN

IMPLEMENTASI ALGORITMA *HYBRID MATCHING* DALAM PENGEMBANGAN APLIKASI REKONSILIASI

TUGAS AKHIR

Telah Diuji Oleh Panitia Ujian Munaqasyah Tugas Akhir
Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh dan Dinyatakan Lulus
Serta Diterima Sebagai Salah Satu Beban Studi Program Sarjana (S-1)
Dalam Program Studi Teknologi Informasi

Pada Hari/Tanggal: Rabu, 19 Agustus 2026
6 Rabiul Awal 1448
Di Darussalam, Banda Aceh

Panitia Ujian Munaqasah Tugas Akhir:

Ketua,	Sekretaris,
Khairan AR, M.Kom	**Ridha Ilahi, M.T**
NIP. 198607042114031001	**NIP. 197905302014031001**
Penguji I,	Penguji II,
Dr. Hendri Ahmadian, M.I.M	**Nazaruddin Ahmad, M.T.**
NIP. 198301042014031002	**NIP. 198206052014031002**

Mengetahui:
Dekan Fakultas Sains dan Teknologi
UIN Ar-Raniry Banda Aceh,

Prof. Dr. Ir. Muhammad Dirhamsyah, M.T., IPU.
NIP. 196206011995031001

LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Yang bertanda tangan dibawah ini

Nama	: Sonia Sabira Gayoara
NIM	: 220705033
Program Studi	: Teknologi Informasi
Fakultas	: Sains dan Teknologi
Judul Tugas Akhir	: Implementasi Algoritma *Hybrid Matching* dalam Pengembangan Aplikasi Rekonsiliasi

Dengan ini menyatakan bahwa dalam penulisan tugas akhir ini, saya:

1. Tidak menggunakan ide orang lain tanpa mampu mengembangkan dan mempertanggungjawabkan;
2. Tidak melakukan plagiasi terhadap naskah tugas akhir orang lain;
3. Tidak menggunakan tugas akhir orang lain tanpa menyebutkan sumber asli atau tanpa izin pemilik tugas akhir;
4. Tidak memanipulasi dan memalsukan data;
5. Mengerjakan sendiri tugas akhir ini dan mampu bertanggung jawab atas tugas akhir ini.

Bila di kemudian hari ada tuntutan dari pihak lain atas tugas akhir saya, dan telah melalui pembuktian yang dapat dipertanggungjawabkan dan ternyata memang dibuktikan bahwa saya telah melanggar pernyataan ini, maka saya siap dikenai sanksi berdasarkan aturan yang berlaku di Fakultas Sains dan Teknologi UIN Ar-Raniry Banda Aceh.

Demikian pernyataan ini saya buat dengan sesungguhnya dan tanpa paksaan dari pihak manapun.

Banda Aceh,

yang menyatakan,

METERAI TEMPEL 10000 1C550AOX127204609

Sonia Sabira Gayoara

ABSTRAK

Nama	:	Sonia Sabira Gayoara
NIM	:	220705033
Program Studi	:	Teknologi Informasi
Judul	:	Implementasi Algoritma *Hybrid Matching* dalam Pengembangan Aplikasi Rekonsiliasi
Tanggal Sidang	:	19 Agustus 2026
Jumlah Halaman	:	9 Halaman
Pembimbing I	:	Khairan AR, M.Kom.
Pembimbing II	:	Ridha Ilahi, M.T.
Kata Kunci	:	*Hybrid Matching*, rekonsiliasi transaksi, Levenshtein Distance, Subset Sum, Streamlit

Proses rekonsiliasi transaksi keuangan yang masih mengandalkan pencocokan konvensional memiliki keterbatasan dalam menangani ketidakkonsistenan narasi serta pola transaksi kompleks satu-ke-banyak, sehingga meningkatkan kebutuhan verifikasi manual dan berpotensi menurunkan efisiensi operasional. Penelitian ini bertujuan mengimplementasikan Algoritma *Hybrid Matching* untuk mencocokkan transaksi dengan variasi teks, menyelesaikan kombinasi nominal satu-ke-satu dan satu-ke-banyak, serta mengembangkan aplikasi rekonsiliasi berbasis web lokal yang mendukung proses verifikasi pengguna. Penelitian menggunakan Metode pengembangan *Rapid Application Development* (RAD) pada data transaksi Rekening Perantara Aktivitas Kliring (RPAK) yang telah dianonimkan. Algoritma mengintegrasikan *data cleansing*, *temporal bounding*, pengukuran kemiripan narasi menggunakan *Levenshtein Distance* melalui RapidFuzz, pencocokan nominal satu-ke-satu, serta *Subset Sum* untuk pola satu-ke-banyak. Sistem dikembangkan menggunakan Python dan Streamlit serta diuji menggunakan *Black-Box Testing*. Hasil pengujian menunjukkan bahwa seluruh fungsi yang diuji berjalan sesuai skenario, meliputi unggah data, pengaturan *similarity threshold*, penyaringan temporal, pencocokan 1-to-1 dan 1-to-N, evaluasi kemiripan narasi, *manual review*, dashboard, dan ekspor laporan. Sistem mampu mempertahankan

transaksi ambigu sebagai *UNMATCHED/Review* sehingga tidak memaksakan pencocokan yang meragukan. Dengan demikian, penerapan *Hybrid Matching* dapat mendukung rekonsiliasi transaksi yang lebih sistematis, fleksibel terhadap variasi data, serta mengarahkan pekerjaan pengguna dari pemeriksaan baris demi baris menuju verifikasi transaksi residu dalam lingkungan pemrosesan lokal.

جامعة الرانري
AR-RANIRY

ABSTRACT

Nama	:	Sonia Sabira Gayoara
NIM	:	220705033
Program Studi	:	Teknologi Informasi
Judul	:	Implementasi Algoritma *Hybrid Matching* dalam Pengembangan Aplikasi Rekonsiliasi
Tanggal Sidang	:	19 Agustus 2026
Jumlah Halaman	:	9 Halaman
Pembimbing I	:	Khairan AR, M.Kom.
Pembimbing II	:	Ridha Ilahi, M.T.
Kata Kunci	:	*Hybrid Matching*, rekonsiliasi transaksi, Levenshtein Distance, Subset Sum, Streamlit

Financial transaction reconciliation that relies on conventional exact-matching mechanisms has limitations in handling inconsistent transaction narratives and complex one-to-many transaction patterns, resulting in increased manual verification and reduced operational efficiency. This study aims to implement a *Hybrid Matching* algorithm to identify transactions with textual variations, resolve one-to-one and one-to-many nominal matching patterns, and develop a local web-based reconciliation application that supports user verification. The study employed with the *Rapid Application Development* (RAD) method using anonymized Rekening Perantara Aktivitas Kliring (RPAK) transaction data. The proposed algorithm integrates data cleansing, temporal bounding, narrative similarity measurement using *Levenshtein Distance* through RapidFuzz, one-to-one nominal matching, and the *Subset Sum* algorithm for one-to-many matching. The system was developed using *Python* and *Streamlit* and evaluated through *Black-Box Testing*. The results show that all tested functions operated according to the specified scenarios, including data uploading, similarity-threshold configuration, temporal filtering, 1-to-1 and 1-to-N matching, narrative similarity evaluation, *manual review*, dashboard presentation, and report export. Ambiguous transactions were retained as *UNMATCHED/Review*, preventing uncertain matches from being

automatically enforced. Therefore, the implementation of *Hybrid Matching* supports a more systematic reconciliation process, provides flexibility in handling transaction-data variations, and shifts users' activities from line-by-line checking toward verification of residual transactions within a local processing environment.

Keywords: *Hybrid Matching*, transaction reconciliation, *Levenshtein Distance*, *Subset Sum*, Streamlit

KATA PENGANTAR

Dengan menyebut nama Allah Yang Maha Pengasih Lagi Maha Penyayang, puji dan syukur penulis panjatkan ke hadirat Allah Swt. Atas limpahan kasih sayang, petunjuk, serta taufik-Nya, penulis diberi kesempatan untuk menuntut ilmu hingga sampai pada tahap akhir pendidikan sarjana. Berkat izin dan ridha-Nya pula, penulis dapat menyelesaikan skripsi yang berjudul **"Implementasi Algoritma *Hybrid Matching* dalam Pengembangan Aplikasi Rekonsiliasi"**.

Shalawat beriring salam senantiasa tercurah kepada junjungan kita, Nabi Besar Muhammad Shallallahu 'Alaihi Wasallam, beserta keluarga, para sahabat, dan pengikut beliau yang setia hingga akhir zaman. Beliaulah suri teladan yang telah membawa umat manusia dari zaman kegelapan menuju zaman yang terang benderang dengan cahaya ilmu pengetahuan dan teknologi seperti yang kita rasakan saat ini.

Penyusunan proposal ini dimaksudkan sebagai langkah awal dan salah satu syarat akademik guna memperoleh gelar Sarjana pada Program Studi Teknologi Informasi, Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh. Melalui proposal ini, penulis berusaha menawarkan solusi teknologi berupa implementasi Algoritma *Hybrid Matching* dalam aplikasi berbasis web lokal untuk mengotomatisasi dan meningkatkan efisiensi proses rekonsiliasi transaksi keuangan, khususnya pada unit operasional PT Bank Syariah Indonesia (BSI) Region I Aceh.

Penulis menyadari bahwa penulisan skripsi ini masih jauh dari kata sempurna. Hal itu didasari karena keterbatasan kemampuan dan pengetahuan yang dimiliki penulis. Besar harapan semoga skripsi ini bermanfaat bagi pembaca dan penulis khususnya. Oleh karena itu, berkat bantuan, bimbingan, motivasi, serta dukungan dari berbagai pihak. yang telah berkontribusi dalam kelancaran penelitian ini. Penulis dengan tulus mengucapkan ribuan terima kasih kepada semua pihak di bawah ini:

1. Teruntuk Mama dan Papa tercinta, dua pilar kekuatan terhebat dalam hidup penulis. Skripsi ini tidak akan pernah cukup untuk membalas setiap peluh,

kesabaran, dan doa-doa tak bersuara yang selalu Mama dan Papa langitkan bersama. Terima kasih karena senantiasa menjadi orang tua yang luar biasa tangguh dan penuh pengertian. Karya ini sepenuhnya penulis persembahkan untuk Mama dan Papa.

2. Adik kandung tersayang, Chaira Maulida Gayoara dan Ray Naufal Gayoara, yang selalu menjadi penghibur, teman berbagi cerita, dan penyemangat tersendiri bagi penulis di tengah kesibukan dan kelelahan menyusun skripsi ini.

3. Kepada Ibu Malahayati, M.T.,selaku Ketua Program Studi Teknologi Informasi Ibu Malahayati, M.T., selaku Ketua Program Studi Teknologi Informasi, yang senantiasa memberikan dukungan administrasi serta arahan akademik demi kelancaran studi mahasiswa di lingkungan prodi selama proses perkuliahan hingga penyelesaian skripsi ini.

4. Kepada Bapak Pembimbing I, Khairan AR, M.Kom. dan Pembimbing II, Bapak Ridha Ilahi,M.T. yang telah membantu penulis sampai dapat mencapai titik ini, yang dengan penuh kesabaran, ketelitian, dan keikhlasan senantiasa memberikan bimbingan, arahan, serta dorongan semangat kepada penulis dari awal hingga akhir penyusunan skripsi ini.

5. Kepada Bapak/Ibu Dekan Fakultas Sains dan Teknologi, Universitas Islam Negeri Ar-Raniry Banda Aceh, yang telah memfasilitasi lingkungan belajar yang kondusif bagi mahasiswa.

6. Kepada Bapak/Ibu dosen Program Studi Teknologi Informasi Universitas Islam Negeri Ar-Raniry, yang sudah dengan baik hati membantu penulis dalam mempermudah segala urusan yang berkaitan dengan kelancaran penyusunan skripsi ini.

7. Kepada Ibu Cut Ida Rahmadiana, S.Si., selaku Staff Operator Prodi Teknologi Informasi yang tercinta, yang selalu tanggap dan sigap dalam membantu segala urusan administrasi selama perkuliahan dengan tulus dan penuh keramahan.

8. Kepada Pimpinan dan Seluruh Staf PT Bank Syariah Indonesia (BSI) Region I Aceh, khususnya pada unit Operational Network and Management, yang telah memberikan izin, bimbingan, dan keterbukaan bagi penulis untuk melakukan

observasi awal guna mendapatkan data-data krusial mengenai proses rekonsiliasi harian yang menjadi dasar studi kasus dalam penelitian ini.

9. Tiga sahabat tercinta dalam kehidupan pribadi penulis, Fitrah Hasanah, M. Faqih Muwaffaq, dan Aneza Aulia Desya. Terima kasih telah menjadi pendengar setia, tempat berkeluh kesah, serta menjadi keluarga kedua yang selalu sabar dan meyakinkan penulis bahwa "kita pasti bisa melewati fase ini bersama-sama" di saat-saat tersulit sekalipun.

10. Rekan seperjuangan, Arief Furqany, Munisa, Siti Mariatul Ulfa, Hilda Sabrina Hashar, Maqhfirah serta sahabat-sahabat tercinta lainnya terima kasih telah menjadi support system yang luar biasa, teman berbagi keluh kesah saat coding bermasalah, hingga saling menyemangati di masa-masa penuh revisi skripsi ini.

Penulis menyadari sepenuhnya bahwa penyusunan skripsi ini masih jauh dari kesempurnaan, baik dari segi penulisan, tata bahasa, maupun kedalaman materi operasional dan komputasi yang disajikan. Oleh karena itu, dengan segala kerendahan hati, penulis sangat mengharapkan kritik serta saran yang membangun dari seluruh pembaca demi perbaikan dan penyempurnaan di masa yang akan datang.

Akhir kata, penulis memanjatkan harapan kepada Allah Swt., semoga skripsi ini dapat memberikan kontribusi positif, memperluas wawasan keilmuan di bidang Teknologi Informasi, serta memberikan manfaat praktis bagi institusi perbankan dan semua pihak yang membutuhkannya.

DAFTAR ISI

ABSTRAK i
ABSTRACT..... iii
KATA PENGANTAR v
DAFTAR ISI..... viii
DAFTAR TABEL..... x
DAFTAR GAMBAR xi
BAB I PENDAHULUAN..... 1
1.1 Latar Belakang..... 1
1.2 Rumusan Masalah..... 3
1.3 Tujuan Penelitian..... 4
1.4 Manfaat Penelitian..... 4
1.5 Batasan Masalah 5
BAB II TINJAUAN PUSTAKA..... 6
2.1 Penelitian Terdahulu..... 6
2.2 Kajian Teoritis 11
BAB III METODOLOGI PENELITIAN..... 19
3.1 Jenis dan Pendekatan Penelitian..... 19
3.2 Alur Penelitian..... 19
3.3 Objek Penelitian 21
3.4 Instrumen Penelitian..... 21
3.5 Pengumpulan Data..... 24
3.6 Perancangan Algoritma 25
3.7 Perancangan Antarmuka Pengguna 32
3.8 Pengujian Perangkat Lunak 36
BAB IV IMPLEMENTASI SISTEM 39
4.1 Gambaran Umum Implementasi Sistem..... 39
4.2 Implementasi Algoritma *Hybrid Matching* (Back-End)..... 39
4.3 Implementasi Antarmuka Sistem *(Front-End)* 47
4.4 Pengujian Sistem 51
4.5 Hasil dan Pembahasan.....
Error! Bookmark not defined.
BAB V PENUTUP..... 58
A. Kesimpulan..... 58
B. Saran 60
DAFTAR PUSTAKA 61

جامعة الرانري
AR-RANIRY

DAFTAR TABEL

Tabel 2.1 Kajian Terdahulu..... 8
Tabel 2.2 Aspek Pembeda..... 10
Tabel 3.1 Perangkat Komputasi.....21
Tabel 3.2 Skenario Pengujian Black-Box.....37
Tabel 4.1 Simulasi Pencarian Kombinasi Nominal Satu-ke-Satu..... 41
Tabel 4.2 Simulasi Pencarian Kombinasi Nominal Satu-ke-Banyak..... 44
Tabel 4.3 Simulasi Evaluasi Similarity Score Berdasarkan Threshold..... 46
Tabel 4.4 Struktur Lembar Kerja pada Laporan Hasil Rekonsiliasi.....50
Tabel 4.5 Skenario dan Hasil Uji Black Box Testing.....52
Tabel 4.6 Komparasi Manual vs Sistem.....54
Tabel 4.7 Akurasi Pencocokan Algoritma.....57

DAFTAR GAMBAR

Gambar 1 Flowchart Tahapan Penelitian..... 20
Gambar 2 Diagram Alur Logika Hybrid Matching.....26
Gambar 3 Alur Pencocokan Pola Satu-ke-Satu.....27
Gambar 4 Alur Pola Satu-ke-Banyak.....29
Gambar 5 Alur Pencocokan Teks.....30
Gambar 6 Wireframe Halaman Upload Data..... 33
Gambar 7 Wireframe Halaman Dashboard Matching.....34
Gambar 8 Wireframe Halaman Panel Intervensi Manual..... 35
Gambar 9 Halaman Laporan Final dan Eksport Data..... 36
Gambar 10 Potongan Kode Pra-pemrosesan Data..... 40
Gambar 11 Potongan Kode Validasi Ketersediaan..... 40
Gambar 12 Potongan Kode Pencocokan Nominal 1-to-1..... 40
Gambar 13 Potongan Kode Pemangkasan Cabang Heuristik..... 42
Gambar 14 Potongan Kode Pencarian Kombinasi Nominal (1-to-N).....43
Gambar 15 Potongan Kode Pembaruan Status dan Skor Kombinasi 1-to-N.....43
Gambar 16 Potongan Kode Evaluasi Similarity Score..... 45
Gambar 17 Tampilan Implementasi Halaman Unggah Data RPAK.....47
Gambar 18 Tampilan Implementasi Halaman Dashboard Rekonsiliasi.....48
Gambar 19 Tampilan Implementasi Halaman Manual Review dan Matching..... 49
Gambar 20 Tampilan Implementasi Halaman Export Laporan..... 50

BAB I
PENDAHULUAN

1.1 Latar Belakang

Dalam dunia bisnis dan perbankan, keabsahan sebuah laporan keuangan sangat bergantung pada proses yang disebut Rekonsiliasi Keuangan. Rekonsiliasi Keuangan pada dasarnya adalah proses mencocokkan atau menyandingkan dua sumber data yang berbeda untuk memastikan tidak ada selisih atau kehilangan dana. Dalam operasional harian bank, proses ini dilakukan dengan membandingkan Catatan Internal Transaksi milik organisasi dengan Rekening Koran (Mutasi Bank) yang dikeluarkan oleh bank. Proses ini disebut juga dengan Pencocokan Referensi, di mana sistem harus memeriksa apakah teks deskripsi/narasi transfer dan nominal di kedua dokumen tersebut sama dan valid (Sudirman & Tristianti, 2022). Namun, tingginya volume transaksi yang bervariasi sering kali memicu ketidaksesuaian data akibat kesalahan input manual, gangguan sistem, atau perbedaan format pencatatan.

Namun, seiring dengan melonjaknya arus data finansial digital yang menuntut pelaporan instan, proses pencocokan ini menghadapi tantangan besar (Putra, 2023). Fakta empiris pada unit operasional PT Bank Syariah Indonesia Region I Aceh menunjukkan bahwa aktivitas penyandingan data transaksi harian hingga saat ini masih mengandalkan perangkat konvensional, yaitu Microsoft Excel dengan fungsi VLOOKUP atau Exact Match. Berdasarkan kalkulasi log transaksi bulanan di lapangan, fungsi kaku tersebut hanya mampu memproses sekitar 45% data secara otomatis. Sementara itu, 55% data sisanya gagal dicocokkan oleh Excel dan terpaksa dialihkan ke prosedur verifikasi manual oleh staf keuangan. Proses manual ini memakan waktu rata-rata hingga 4 menit per batch data, dengan tingkat kesalahan manusia (human error) mencapai 3% akibat faktor kelelahan mata dan pikiran. Keterlambatan ini sangat berisiko memicu adanya transaksi menggantung yang menghambat laporan audit penting seperti pada aplikasi MonSAKTI. Kondisi tersebut diperparah oleh keterbatasan metode pencocokan tradisional yang gagal mengidentifikasi anomali transaksi akibat variasi format narasi, yang jika dibiarkan akan menurunkan efisiensi operasional secara signifikan (Andini & Muslimin, 2026).

Mengapa Microsoft Excel dan fungsi VLOOKUP standar sering gagal menangani data rekonsiliasi? Penyebab utamanya terletak pada sifat Excel yang sangat kaku dan tidak memiliki toleransi terhadap perbedaan karakter tulisan sekecil apa pun. Di sinilah muncul masalah pertama, yaitu inkonsistensi narasi

transaksi. Pada data riil, manusia sering melakukan kesalahan pengetikan (typo), perbedaan ejaan, spasi berlebih, atau penulisan singkatan yang berbeda antara catatan internal dan bank (misalnya: di catatan internal tertulis "Pembayaran Cicilan Ahmad", sedangkan di rekening koran bank tertulis "Pmb. Ccln Ahmd"). Karena tulisannya tidak sama persis, Excel akan menganggap data itu tidak cocok.

Masalah kedua yang jauh lebih rumit adalah adanya perbedaan pola hubungan data finansial, yang terbagi menjadi dua skenario:

1. Pola Satu-ke-Satu (*One-to-One*): Satu baris data di catatan internal cocok dengan satu baris data di rekening koran bank. (Ini pola yang sederhana).

2. Pola Satu-ke-Banyak *(One-to-Many)*: Satu baris data di catatan internal ternyata merupakan gabungan dari beberapa baris transaksi yang terpisah di rekening koran bank. Kasus ini sering terjadi pada pembayaran cicilan bertahap atau transfer parsial (misalnya: catatan internal mencatat ada piutang masuk Rp3.000.000, tetapi di rekening koran bank uang itu masuk terpisah tiga kali, masing-masing Rp1.000.000).

Dalam ilmu komputer, mencari kombinasi angka untuk mencocokkan pola satu-ke-banyak ini disebut sebagai *Subset Sum Problem* yang tergolong dalam kelas optimasi *NP-Hard*. Jika staf keuangan dipaksa mencari kombinasi angka-angka terpisah itu satu per satu secara manual, akan terjadi beban kerja luar biasa yang melumpuhkan produktivitas (Setiawan, 2023), (Mücke et al., 2022). Di sisi lain, menggunakan perangkat lunak komersial skala besar *(Enterprise Resource Planning / ERP)* bukanlah pilihan yang bijak karena biaya lisensinya yang sangat mahal, serta antarmukanya yang kaku sehingga memaksa pengguna mengubah format file mentah mereka terlebih dahulu (Kurniawan, 2022).

Sebelumnya, sempat dilakukan pengujian awal menggunakan platform Google Colab berbasis *Python* untuk memprogram solusi masalah ini. Namun, platform berbasis komputasi awan *(cloud)* tersebut tidak dapat diimplementasikan secara resmi di perbankan karena mewajibkan pengguna mengunggah file keuangan mentah ke server publik. Hal ini melanggar regulasi ketat kerahasiaan data nasabah yang diatur dalam POJK Nomor 11/2022 dan UU Perlindungan Data Pribadi (UU PDP) Nomor 27 Tahun 2022 (Nasution 2023, Lustarini 2024) Selain itu, tampilan Google Colab yang berupa baris kode mentah sangat tidak ramah bagi staf keuangan non-teknis karena rentan memicu eror saat dioperasikan.

Beberapa penelitian terdahulu telah mencoba menyelesaikan masalah ini secara terpisah. (Nida Winarti et al., 2022) berfokus pada penanganan identitas transfer yang tidak jelas, sementara (Suwarningsih & Nuryani, 2024) membuktikan bahwa algoritma jarak string *(string distance)* sangat akurat untuk mengatasi variasi teks ejaan. Untuk urusan angka nominal, (Wijaya et al., 2024) mengeksplorasi penggunaan optimasi heuristik untuk mempercepat pencarian kombinasi sub-angka secara otomatis. Tren riset ini membuktikan bahwa perpindahan dari validasi visual manual menuju validasi algoritmik otomatis adalah kunci utama untuk membangun ketahanan data *(data resilience)* yang tinggi (Susanto & Wibowo, 2024).

Sebagai solusi total yang aman, mudah, dan sesuai dengan judul penelitian ini, dikembangkanlah sebuah aplikasi otomatisasi berbasis web menggunakan framework Streamlit lokal dengan bahasa pemrograman *Python*. Sistem ini mengimplementasikan Algoritma *Hybrid Matching* yang menggabungkan dua logika cerdas, yaitu metode *Levenshtein Distance* untuk membaca dan mencocokkan kemiripan teks narasi yang berantakan *(Fuzzy Matching)*, serta algoritma optimasi heuristik untuk memecahkan pencarian kombinasi angka pada pola *one-to-one* dan *one-to-many* secara instan. Karena dijalankan secara lokal *(offline/localhost)*, data internal bank dijamin aman dari kebocoran luar (Wijaya et al., 2024), (Kurniawan, 2022). Sebagai hasil keluaran *(output)*, aplikasi Streamlit ini akan menyajikan laporan akhir rekonsiliasi yang bersih, lengkap dengan metrik MATCH *ID* dan *Similarity Score* untuk mempermudah staf keuangan melakukan verifikasi akhir dengan cepat dan akurat (Nugroho, 2023).

1.2 Rumusan Masalah

Berdasarkan latar belakang dan identifikasi variabel penelitian yang telah diuraikan, rumusan masalah berfokus pada penyelesaian kendala inkonsistensi data serta kompleksitas pola transaksi di lapangan. Adapun rumusan masalah dalam penelitian ini disusun sebagai berikut:

1. Bagaimana menerapkan Algoritma *Hybrid Matching* agar dapat mengidentifikasi kecocokan data transaksi keuangan yang memiliki inkonsistensi teks (narasi)?

2. Bagaimana rancangan logika Algoritma *Hybrid Matching* dalam menyelesaikan pencocokan transaksi keuangan dengan pola kompleks satu-ke-banyak secara akurat?

3. Bagaimana membangun antarmuka lokal berbasis web menggunakan Streamlit yang informatif dan dilengkapi metrik validasi (MATCH *ID & Similarity Score*) untuk memudahkan verifikasi hasil rekonsiliasi?

4. Bagaimana tingkat akurasi pencocokan dan efisiensi waktu komputasi yang dihasilkan dari implementasi Algoritma *Hybrid Matching* tersebut jika dikomparasikan dengan prosedur penyandingan manual?

1.3 Tujuan Penelitian

Selaras dengan rumusan masalah yang telah ditetapkan, penelitian ini dirancang untuk mencapai sasaran teknis dan fungsional yang spesifik. Secara terperinci, tujuan utama dari penelitian ini adalah:

1. Mengimplementasikan Algoritma *Hybrid Matching* dalam mendeteksi dan mencocokkan pasangan data transaksi keuangan dengan narasi teks yang tidak seragam.

2. Mengembangkan fungsi algoritma untuk memecahkan dan mengelompokkan data transaksi dengan pola kompleks satu-ke-banyak pada pencocokan nilai nominal keuangan.

3. Menghasilkan aplikasi rekonsiliasi lokal berbasis Streamlit yang menyajikan laporan akhir informatif beserta metrik MATCH *ID* dan Similarity Score guna mempermudah proses verifikasi pengguna.

4. Mengukur tingkat keberhasilan implementasi algoritma melalui evaluasi akurasi pencocokan transaksi dan persentase penghematan waktu operasional dibandingkan dengan metode manual.

1.4 Manfaat Penelitian

Pengembangan instrumen otomasi ini diproyeksikan memberikan kontribusi yang terukur, baik pada tataran operasional industri maupun pengembangan keilmuan akademis. Dampak signifikan tersebut diklasifikasikan bagi tiga aspek penerima manfaat utama, yaitu:

1. Manfaat Pragmatis: Mengotomatisasi proses penyelesaian rekonsiliasi keuangan dan meminimalisir tingkat kesalahan manusia (*human error*) dalam aktivitas operasional harian.

2. Manfaat Teoritis: Menjadi referensi studi akademis mengenai penerapan Algoritma *Hybrid Matching* untuk penyelesaian kasus pencocokan teks dan nominal transaksi secara terintegrasi.

3. Manfaat Praksis: Meningkatkan keahlian praktis dalam pemrograman *Python*, manipulasi struktur data, serta rekayasa pengembangan aplikasi web lokal menggunakan *framework* Streamlit.

1.5 Batasan Masalah

Guna menjaga agar observasi tetap terarah, tidak meluas dari topik utama, serta memastikan akurasi kalibrasi pada saat pengujian sistem, ruang lingkup rekayasa perangkat lunak dalam penelitian ini dibatasi pada parameter teknis berikut:

1. Domain data dibatasi pada berkas dokumen statis (berformat .csv dan .xlsx) yang bersumber dari catatan internal perusahaan dan laporan mutasi bank, beroperasi murni tanpa adanya integrasi koneksi API langsung ke sistem perbankan.

2. Aplikasi dibangun menggunakan *framework* Streamlit (*Python*) dan dijalankan secara lokal (*offline*) melalui eksekusi pada mesin *localhost*.

3. Pengujian sistem dibatasi pada ranah validasi fungsionalitas fitur dan tingkat akurasi logika algoritma dengan menggunakan metode *Black Box Testing*.

BAB II
TINJAUAN PUSTAKA

2.1 Penelitian Terdahulu

Konstruksi pengembangan dan rekayasa perangkat lunak ini bersandar secara esensial pada literatur akademis lintas disiplin yang mencakup kajian sistem informasi akuntansi, algoritma pencocokan bahasa alami, serta optimasi kombinatorial komputasi. Penelusuran terhadap publikasi terdahulu dipaparkan secara mendalam guna memetakan posisi penelitian ini serta mengidentifikasi celah keilmuan yang akan diisi (Taherdoost & Madanchian, 2023).

Mengenai Transformasi Sistem Akuntansi melalui Ekosistem Otomatisasi (*BPA/RPA*), sebuah studi fundamental mengenai disrupsi teknologi di ranah akuntansi diterbitkan dalam kajian empiris berjudul *'Robotic Process Automation for Auditing'* yang dipelopori oleh Moffitt, Rozario, dan Vasarhelyi (Moffitt et al., 2018). Penelitian ini mengeksplorasi utilitas pemanfaatan perangkat lunak otomatisasi untuk mengintervensi kegiatan klerikal dan tugas administratif berulang yang lazim dijumpai dalam ekosistem audit. Kesimpulan utama dari kajian tersebut menegaskan bahwa delegasi proses kepada mesin robotik mampu mengeliminasi bias kognitif yang memicu deviasi akurasi, seraya mereduksi faktor kelelahan fisik staf yang acap kali memicu kesalahan dalam pelaporan faktual. Transisi menuju paradigma otomatisasi di lanskap domestik Indonesia turut tervalidasi secara akademik oleh Priyanto dan Suhandi. Publikasi mereka menunjukkan bahwa *Robotic Process Automation* telah berevolusi dari sekadar wacana konseptual menjadi kebutuhan absolut untuk menjamin efisiensi berkelanjutan dalam profesi akuntan. (Priyanto & Suhandi, 2021)

Akan tetapi, temuan Priyanto dan Suhandi menggarisbawahi adanya jurang implementasi yang cukup dalam yakni ketiadaan instrumen aplikasi portabel yang terjangkau secara finansial dan aplikatif untuk diadopsi langsung oleh perusahaan skala menengah. Kondisi ini dikonfirmasi lebih jauh melalui laporan komprehensif *Indonesia CFO Survey* yang disusun oleh firma konsultan Pricewaterhouse Coopers (PricewaterhouseCoopers Indonesia, 2022). Laporan ini mengindikasikan paradoks di mana 94% eksekutif keuangan memiliki dorongan kuat untuk menanamkan modal pada sistem otomatisasi, namun niat tersebut tertahan oleh kompleksitas infrastruktur *cloud* yang ditawarkan vendor komersial serta kurva pembelajaran antarmuka sistem yang berbelit-belit yang menimbulkan penolakan. Berpijak pada hambatan struktural inilah, aplikasi berbasis *Local Web-App* Streamlit yang

diajukan dalam skripsi ini dirancang guna menyodorkan solusi nirbiaya dengan sistem yang dapat disesuaikan secara dinamis dan mudah digunakan oleh kalangan staf non-TI (Junaedi & Purnomo, 2023).

Pendekatan Metrik Probabilistik untuk Resolusi Inkonsistensi Data Literasi keilmuan terkait optimalisasi utilitas pengukuran jarak teks dalam mengidentifikasi korelasi entitas dan mendeteksi anomali penulisan (*fraud*) telah mendominasi perbincangan rekayasa data. Melalui investigasinya di Massachusetts Institute of Technology, Liu secara analitis membedah akar kegagalan fatal yang dialami oleh metode logika perbandingan string bawaan sistem (seperti formula *Exact* MATCH pada SQL) tatkala dihadapkan pada entitas nama klien atau referensi transfer yang ditulis bervariasi secara leksikal. Liu berhasil membuktikan bahwa injeksi pendekatan kalkulasi algoritma *Levenshtein Distance* memberikan ketahanan operasional komputasi yang luar biasa tinggi terhadap manipulasi teks, memulihkan data yang awalnya ditolak oleh sistem ortodoks. (Liu & Chen, 2020) Efisiensi serupa dikaji oleh Kim dan Yang dalam riset mereka yang memfungsikan model automata Levenshtein untuk menanggulangi deviasi fonetik ganda dan mendeteksi selipan kode-kode acak (*lexical manipulation*) dalam investigasi transfer dana perbankan lintas batas. Namun demikian, intervensi komputasi yang dilakukan oleh studi-studi terdahulu berfokus murni pada resolusi penamaan entitas satu-ke-satu, tanpa menyentuh krusialnya variabel sinkronisasi akumulasi numerik yang menjadi inti dari pelunasan kas. Algoritma hibrida penelitian ini dirumuskan untuk melengkapi keterbatasan (*blind spot*) tersebut. (Kim & Yang, 2022)

Dekonstruksi Matematis Kombinatorial *Subset Sum Matching Problem* (*SSMP*). Upaya paling mutakhir dalam menjawab teka-teki logika penyusunan rekonsiliasi komposit di mana satu saldo mayor ditambal secara terpecah oleh setoran berskala mikro diformalisasikan dalam rumusan persamaan aljabar oleh Wu et al. (Wu & others, 2025) Mereka mengklasifikasikan skenario ini sebagai tantangan optimasi *Subset Sum Matching Problem* (*SSMP*). Dalam model ini, perangkat komputasi dituntut merajut kepingan entitas elemen dari pangkalan data asimetris untuk disintesis menjadi sekelompok aglomerasi agregat ekuivalen. Wu et al.mengajukan strategi algoritma pemecahan dinamis yang memandu probabilitas ruang pencarian guna memangkas jalur kebuntuan komputasi (Wu, et al. 2025). Serupa dengan landasan teoritis tersebut, publikasi riset Biesner, Sifa, dan Bauckhage mendemonstrasikan secara empiris bahwa kegiatan melacak selisih anomali di tabel neraca audit akuntansi sejatinya merupakan manifestasi penerapan dari kelas komputasi yang paling sulit, yakni *Subset Sum NP-Hard* (Biesner et al.,

2022). Mengekstraksi esensi dari cetak biru metode pemangkasan cabang heuristik (*heuristic tree pruning*) dari ragam literatur ini, penelitian rekayasa perangkat lunak skripsi ini akan membungkus strategi pemfilteran limitasi deviasi tanggal ke dalam rutinitas perulangan *Python*, yang pada akhirnya memampukan komputer memecahkan kebuntuan cicilan 1-ke-N secara paripurna.

Untuk merangkum seluruh sintesis dari konstelasi kajian literatur di atas, Tabel 2.1 dibawah memaparkan matriks komparatif yang membedah batasan dari studi terdahulu serta sumbangsih kebaruan dari instrumen yang diusulkan:

Tabel 2.1 Kajian Terdahulu

Fokus Kajian Literatur	Temuan Relevan	Kesenjangan	Kontribusi Penelitian Ini
Robotic Process Automation (RPA) untuk keperluan Auditing. Moffitt, Rozario, & Vasarhelyi (2018)	Implementasi sistem berbasis agen robotik secara radikal mampu mengeliminasi bias kognitif manusia dan mereduksi kelelahan fisik staf pada tugas administratif berulang.	Fokus kajian terbatas pada ekosistem klerikal audit secara umum, belum menawarkan aplikasi praktis skala lokal untuk pencocokan transaksi spesifik.	Menghadirkan inisiatif perangkat lunak otomatisasi proses bisnis (BPA) berbasis web lokal yang ringan menggunakan Streamlit.
Transisi dan evolusi paradigma otomatisasi akuntansi di lanskap domestik Indonesia. Priyanto & Suhandi (2022)	RPA telah berevolusi menjadi kebutuhan absolut untuk menjamin kelangsungan efisiensi profesi akuntan di era digital.	Menemukan jurang implementasi berupa ketiadaan instrumen aplikasi portabel yang terjangkau secara finansial dan aplikatif untuk diadopsi langsung oleh perusahaan skala menengah.	Menyodorkan aplikasi nirbiaya dengan sistem penyesuaian parameter dinamis yang mudah dioperasikan oleh kalangan staf operasional non-TI.

Metrik pengukur jarak teks *(String Matching)* untuk resolusi inkonsistensi data leksikal. Liu & Chen (2020)	Injeksi algoritma Levenshtein Distance terbukti memberikan ketahanan operasional tinggi terhadap manipulasi teks dan memulihkan data variasi nama yang awalnya ditolak oleh metode *Exact* MATCH pada SQL konvensional.	Pengujian dan pemodelan komputasi terbatas hanya pada sinkronisasi atau komparasi string entitas tunggal dengan pola satu-ke-satu (1-ke-1).	Merumuskan arsitektur *Hybrid Matching* yang mengeksekusi penyaringan dasar berbasis logika pengurangan angka nominal mutlak sebelum mengaktivasi beban spasial teks.
Model Automata Levenshtein untuk pelacakan anomali transaksi perbankan. Kim & Yang (2022)	Memfungsikan model Automata Levenshtein efektif menanggulangi deviasi fonetik ganda dan manipulasi leksikal teks dalam investigasi transfer dana perbankan lintas batas.	Fokus analisis murni pada resolusi penamaan entitas (1-ke-1), tanpa mempertimbangkan variabel sinkronisasi akumulasi numerik kumulatif yang krusial bagi pelunasan kas.	Mengintegrasikan kalkulasi kemiripan teks deskripsi menggunakan metrik Levenshtein Distance melalui pustaka *RapidFuzz* ke dalam sistem otomatisasi.
Formulasi optimasi *Subset Sum Matching Problem* (SSMP). Wu et al. (2019)	Mengklasifikasikan skenario rekonsiliasi komposit (satu saldo makro ditambal pecahan setoran mikro) ke dalam model matematis SSMP dengan strategi	Implementasi dan pembuktian empiris studi masih tertahan dalam wujud pemodelan matematis teoretis dan abstrak, belum diintegrasikan ke perangkat siap pakai.	Menanamkan fungsi penelusuran pencarian akar agregat 1-ke-N langsung di balik dasbor antarmuka grafis yang ramah pengguna.

	pemrograman dinamis.		
Kompleksitas komputasi pelacakan selisih saldo audit akuntansi. Biesner, Sifa, & Bauckhage (2022)	Membuktikan secara empiris bahwa kegiatan melacak selisih anomali di tabel neraca keuangan merupakan manifestasi penerapan kelas komputasi tersulit, yaitu *Subset Sum NP-Hard.*	Mengajukan metode pemangkasan cabang heuristik (heuristic tree pruning) secara teoritis namun belum diaplikasikan pada arsitektur aplikasi desktop/lokal yang praktis.	Membungkus strategi pemfilteran limitasi jendela waktu temporal (+- 3 hari) ke dalam rutinitas perulangan *Python* agar PC standar mampu memecahkan kasus cicilan 1-ke-N secara cepat.

Untuk merangkum seluruh sintesis dari konstelasi kajian literatur di atas, Tabel 2.2 memaparkan matriks komparatif yang membedah batasan dari studi terdahulu serta sumbangsih kebaruan dari instrumen yang diusulkan:

Tabel 2.2 Aspek Pembeda

Aspek Pembeda	**Sistem Lama**	**Aplikasi Streamlit Hibrida (*Proposed System*)**
	Fleksibilitas Input (*No Format Shifting*)	Kaku. Pengguna dipaksa mengubah format file Excel/CSV mereka agar pas dengan template kaku bawaan sistem vendor.
Pola Pencocokan Data	Terbatas 1-ke-1. Hanya bisa mencocokkan satu baris data internal dengan satu baris data bank yang nilainya sama persis (*Exact Match*).	Hibrida (1-ke-1 & 1-ke-N). Mampu memecahkan pola cicilan. Satu angka besar di catatan internal dicari kombinasi pecahan angka kecilnya di bank

		secara otomatis lewat algoritma *Subset Sum*.
Penanganan Data *Typo* / Singkatan	Gagal/Error. Jika ada beda spasi atau singkatan (misal: "HOTEL" vs "BIAYA HTL"), sistem langsung menolak (UNMATCHED).	Cerdas (*Fuzzy Matching*). Menggunakan *Levenshtein Distance* untuk menghitung kemiripan teks, lalu memberikan persentase kecocokan (*Similarity Score*).
Kedaulatan & Keamanan Data	Berbasis Cloud. Seringkali harus mengunggah data ke server awan pihak ketiga, berisiko melanggar regulasi privasi data perbankan.	Offline Lokal. Berjalan murni di PC lokal instansi (*localhost*) via *Batch Script*, mematuhi POJK dan UU PDP No 27 Tahun 2022.
Peran Manusia (*Human-in-the-Loop*)	Hitam-Putih. Sistem langsung memutuskan secara sepihak, menyisakan data *error* yang membingungkan akuntan tanpa ruang mediasi.	Sinergis. Mesin melakukan otomatisasi mayoritas data, dan menyediakan panel *Human-in-the-Loop* khusus agar manusia menjadi verifikator akhir untuk data berstatus ragu-ragu.

2.2 Kajian Teoritis

2.2.1 Ontologi Rekonsiliasi Transaksi Finansial dan Paradigma Otomatisasi

Di dalam ekosistem perbendaharaan dan akuntansi, rekonsiliasi merupakan instrumen analitis krusial untuk memastikan nilai saldo yang tertera pada catatan buku besar perusahaan sejalan dengan bukti mutasi dari pihak bank (Nariman et al., 2024). Ekuivalensi murni yang terjadi secara instan (sekumpulan sel bernilai 0 jika dikurangi) adalah kejadian statistik yang sangat jarang ditemui, yang disebabkan oleh beragam asimetri informasi dan jeda waktu seperti penundaan setoran, penahanan kliring antarbank, atau pembebanan biaya administrasi tersembunyi (Nariman et al., 2024). Ketika prosedur validasi silang ini dijalankan baris demi baris pada skala ribuan baris data, ia menjelma menjadi sebuah hambatan operasional kronis yang mengurangi margin waktu kerja produktif divisi finansial.

Merespons ketidakefisienan struktural tersebut, paradigma Business Process Automation (*BPA*) diterapkan. Paradigma BPA memobilisasi integrasi algoritma berbasis aturan untuk mengotomatisasikan siklus repetitif, mendelegasikan tugas penyandingan angka dan karakter dari staf manusia kepada kapabilitas rekayasa peladen mesin (Peng et al., 2023). Transisi menuju model BPA mentransformasikan pola kerja akuntan dari rutinitas penelusuran administratif tingkat dasar menuju rutinitas tingkat lanjut yang berpusat semata-mata pada evaluasi dan resolusi transaksi berstatus pengecualian (*exception handling*). Di dalam arsitektur penelitian ini, sistem dikendalikan sepenuhnya oleh algoritma pemrograman tingkat tinggi *Python* yang menjalankan tahapan pembacaan rekening koran secara sekuensial dan mandiri. Pendekatan ini tidak hanya meminimalisir intervensi manual yang rentan terhadap human error, tetapi juga secara drastis meningkatkan presisi serta auditabilitas data melalui kerangka kerja rekonsiliasi yang terstruktur (Kale, 2025).

2.2.2 Logika Probabilistik dan Metrik *Levenshtein Distance*

Pendekatan logika Boolean yang dieksekusi oleh mesin basis data relasional konvensional menuntut kekakuan absolut, di mana status ekuivalen hanya diberikan apabila string teks masukan bersesuaian persis hingga ke tingkatan spasi. Pendekatan deterministik ortodoks ini mengalami kolaps manakala dihadapkan pada fenomena deskripsi koran bank yang kerap dikemas penuh modifikasi, seperti transposisi huruf atau insersi karakter. Sebagai solusi terhadap disonansi linguistik tersebut, penelitian ini menyematkan modul logika probabilistik (*Fuzzy Matching*) yang sanggup mengevaluasi derajat kesamaan (*similarity index*) antar string berdasarkan perhitungan matriks dinamis yang dikenal sebagai algoritma *Levenshtein Distance* (Candra & Girsang, 2021).

Secara konseptual matematis, formula metrik penelusuran ini menghitung nilai kuantitatif "jarak" dari dua buah susunan kata berdasarkan seberapa banyak aksi manipulasi penyuntingan dasar absolut yang wajib dilakukan untuk mentransformasi susunan huruf pertama (A) menjadi huruf ekspektasi target (B). Setiap instruksi modifikasi karakter baik itu penyisipan (*insertion*), penghapusan (*deletion*), maupun substitusi silang (*substitution*) akan dikenai satu unit pembobotan penalti.

Secara formal, operasi perhitungan antara dua string a (dengan panjang a) dan string b (dengan panjang b) dinyatakan melalui fungsi jarak $lev_{a,b}(|a|, |b|)$ yang didefinisikan sebagai berikut:

$$lev_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{jika } \min(i,j) = 0 \\ \min\begin{cases} lev_{a,b}(i-1,j)+1 \\ lev_{a,b}(i,j-1)+1 \\ lev_{a,b}(i-1,j-1)+1_{(a_i \neq b_j)} \end{cases} & \text{jika } \min(i,j) > 0 \end{cases}$$

Untuk memahami bagaimana sistem mengalkulasi jarak leksikal pada data transaksi perbankan, penalaran terhadap komponen-komponen fungsi di atas dijabarkan sebagai berikut:

1. Kondisi Batas

$$\text{if } \min(i,j) = 0 \Longrightarrow \max(i,j)$$

Jika salah satu string memiliki panjang 0 ($min(i, j) = 0$), maka jarak Levenshtein adalah panjang string yang tersisa ($\max(i, j)$. Secara matematis, ini merepresentasikan biaya penghapusan atau penyisipan total jika membandingkan string berisi teks dengan string kosong.

2. Indeks i dan j

$$i \in \{1,2,\dots,|a|\} \text{ dan } j \in \{1,2,\dots,|b|\}$$

Berfungsi sebagai penunjuk posisi karakter (*pointer*) yang berjalan secara sekuensial. Indeks i menelusuri string catatan internal a, sedangkan indeks j menelusuri string mutasi bank b.

3. Operasi Baris Pertama (Penghapusan)

$$\text{lev}_{a,b}(i-1,j)+1$$

Merepresentasikan operasi penghapusan (*deletion*) satu karakter dari string internal dengan penalti $+1$ biaya komputasi.

4. Operasi Baris Kedua (Penyisipan)

$$\text{lev}_{a,b}(i,j-1)+1$$

Merepresentasikan operasi penyisipan (*insertion*) satu karakter ke string internal dengan penalti $+1$ biaya komputasi.

5. Operasi Baris Ketiga (Penggantian)

$$lev_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)}$$

Merepresentasikan operasi penggantian (*substitution*). Indikator fungsi $1_{(a_i \neq b_j)}$ bernilai 1 jika karakter pada posisi a_i tidak sama dengan b_j (dikenakan penalti), dan bernilai 0 jika karakter tersebut sama persis (tidak ada penalti/biaya).

Nilai jarak integer ($lev_{a,b}$) yang diperoleh dari perhitungan matriks dinamis ini kelak akan melalui proses normalisasi rasio panjang string untuk diterjemahkan menjadi metrik persentase skor keyakinan (Similarity Score) pada skala 0 hingga 100 persen. Rumus transformasi normalisasi tersebut dieksekusi sebagai berikut:

$$\text{Similarity Score} = (1 - \frac{lev_{a,b}(|a|, |b|)}{\max(|a|, |b|)}) \times 100\%$$

Dengan memfasilitasi parameter penetapan batas toleransi (*threshold*) yang dapat disesuaikan pengguna (misalnya minimal kecocokan 80%), mesin dimampukan untuk mengidentifikasi korelasi makna secara cerdas dan mendaftarkannya sebagai pasangan serasi, meski struktur teks penulisan pada rekening koran memiliki variasi leksikal akibat keterbatasan sistem bank atau kesalahan pengetikan manual manusia (Fischer et al., 2021).

2.2.3 Struktur Komputasi Subset Sum *NP-Hard* dan Penapis Heuristik

Dalam pemrosesan struktur keuangan satu-ke-banyak (skenario agregat pembayaran parsial), sebuah nilai saldo yang besar di dalam catatan internal perusahaan direpresentasikan secara asimetris oleh rentetan penyetoran minor di dalam dokumen mutasi pihak bank. Upaya mengekstraksi dan menyintesis sub-himpunan kombinasikelas permasalahan sama persis dengan angka target internal merepresentasikan persoalan optimasi klasik *Subset Sum Problem (SSP)*. Permasalahan varian teori graf kombinatorial ini diklasifikasikan ke dalam kategori kelas permasalahan struktur *NP-Hard*. Artinya, apabila algoritma dipaksa untuk mengkalkulasi dan menjajaki seluruh permutasi probabilitas dari sekumpulan elemen data yang luas secara serentak (strategi sapu bersih / *brute-force evaluation*), kompleksitas asimtotik waktu penelusuran akan bertumbuh secara eksponensial ekstrem. Komputasi absolut yang meroket ini dapat memicu kelumpuhan $O(2^n)$ penumpukan memori antrean (*memory overflow*),

menyebabkan program mengalami penghentian paksa (Parque, 2021).

Sebagai taktik intervensi teknis guna mendevaluasi beban matriks *NP-Hard* tersebut agar dapat dieksekusi secara rasional oleh perangkat keras standar perbankan, peneliti memobilisasi fungsi pemangkasan cabang probabilitas (*Heuristic Search Limit Pruning*). Mesin secara terstruktur memberlakukan prasyarat penyaringan jendela penanggalan kronologis (*Temporal Date Bounding*). Sebelum modul algoritma mulai memanggil fungsi rekursif matematisnya untuk menjumlahkan rentetan angka cicilan, skrip akan menguji properti metadata temporal, dan hanya memperbolehkan serangkaian data masuk ke dalam klaster evaluasi himpunan jika data tersebut berada di dalam satu wilayah berhimpitan jarak kalender kliring yang dekat (contoh deviasi maksimum 3 hari kalender). Mekanisme penjegalan awal ruang probabilitaskegagalan sistemal ini sukses merampingkan kompleksitas waktu algoritma menjadi model pemrograman fungsional dengan skala linier, memberdayakan komputasi untuk membingkai teka-teki akumulasi transaksi dalam hitungan detik tanpa galat crash.

2.2.4 Ekosistem Bahasa dan Arsitektur Streamlit Lokal

Untuk mewujudkan purwarupa komputasi hibrida, bahasa pemrograman *Python* dipilih sebagai inti arsitektural. Modul kerangka data Pandas difuspasiuk mengatur rutinitas asimilasi *DataFrame*, melacak matriks indeks multidimensi baris-kolom tabular, serta menyeragamkan distorsi struktur dokumen berbekal filter Ekspresi Reguler (*Regex*) untuk sterilisasi format white space atau galat format stempel waktu absolut (*Datetime Object*). Pemrosesan iteratif probabilistik kata dan kalkulasi jarak algoritma diakselerasi penggunaan memorirvensi *RapidFuzz*, sebuah pustaka berbasis bahasa sintaks C++ yang dipilih karena komparasi evaluasi kinerjanya menyuguhkan efisiensi throughput 40% lebih prima dibandingkan pustaka sebelumnya, dengan mengonsumsi rekam jejak RAM yang sangat moderat demi stabilitas performa harian. (Li & Wang, 2023)

Keunikan integrantarmukatural paling menonjol dari skripsi ini disumbangkan oleh pemanfaatan modul presentasi *front-end* yang ditenagai oleh fungsionalitas kerangka kerja Streamlit. Arsitektur pustaka ini memungkinkan penciptaan ekosistem tata muka visual dasbor reaktif (*Dashboard UI*) yang bersifat portabel dan diimplementasikan murni sebagai penjelajah lokal terisolasi (*Portable Local Environment*) pada PC pengguna. Skrip arsitektur grkepatuhan penuh pada inisiasi *Batch Script* pada OS sistem berbasis perintah eksekutor *Localhost*. Ketiadaan ketergantungan koneksi interaksi peladen internet awan publik pada

konstruksi arsitektur Streamlit ini memastikan ketundukan mutlak aplikasi pada regulasi kepatuhan pembatasan perlindungan privasi data korporasi.

2.2.5 Regulasi Keamanan Data Transaksi

Lanskap hukum Indonesia memberikan mandat konstitusional bagi pelindungan data pribadi melalui Pasal 28G ayat dan Pasal 28H ayat Undang-Undang Dasar Negara Republik Indonesia Tahun 1945, yang kemudian diwujudkan dalam pengesahan Undang-Undang Nomor 27 Tahun 2022 tentang Pelindungan Data Pribadi (Matheus & Gunadi, 2023). UU PDP mendefinisikan pelindungan data pribadi sebagai upaya melindungi pemrosesan data pribadi guna memperoleh hak hukum subjek data, mencakup pengaturan hak subjek data, kewajiban pengendali dan prosesor data, serta ketentuan sanksi pidana dan administratif bagi pelanggaran pengumpulan, pengungkapan, maupun penggunaan data pribadi tanpa izin (Badar et al., 2023; Soemitro et al., 2023). Dalam konteks perbankan, kewajiban kerahasiaan dipertegas pula oleh Pasal 40 Undang-Undang Nomor 10 Tahun 1998 tentang Perbankan, yang menjadi landasan prinsip kerahasiaan (confidential principle) atas data nasabah (Ririn Puspita Dewi & Diana Wiyanti, 2024).

Pada tataran sektoral, Otoritas Jasa Keuangan menerbitkan Peraturan OJK Nomor 11/POJK.03/2022 tentang Penyelenggaraan Teknologi Informasi oleh Bank Umum, yang antara lain mewajibkan pembentukan komite pengarah teknologi informasi serta penguatan sistem keamanan digital dan manajemen risiko siber bank (Lia Sari, 2026; Silalahi et al., 2026). Urgensi regulasi ini terbukti relevan ketika sektor perbankan nasional menghadapi insiden siber nyata, sebagaimana kasus serangan ransomware pada Bank Syariah Indonesia tahun 2023 yang mengekspos kerentanan pengelolaan data nasabah (Ririn Puspita Dewi & Diana Wiyanti, 2024; Silalahi et al., 2026).

Implikasi regulasi tersebut terhadap rancangan aplikasi penelitian ini bersifat direktif: seluruh pemrosesan data transaksi yang memuat identitas nasabah dilarang dialihkan ke infrastruktur pihak ketiga tanpa jaminan kepatuhan. Oleh karena itu, aplikasi rekonsiliasi dirancang beroperasi sepenuhnya pada lingkungan lokal (localhost) tanpa integrasi API eksternal maupun unggahan ke peladen awan publik, sehingga data mentah tidak pernah meninggalkan perangkat instansi. Arsitektur tertutup ini sekaligus meminimalisir risiko pelanggaran ketentuan pemrosesan data pribadi yang diancam sanksi pidana berdasarkan UU PDP (Soemitro et al., 2023).

2.2.6 Metode Pengembangan Rapid Application Development

Rapid Application Development merupakan metodologi pengembangan sistem yang memanfaatkan joint application design, pembangunan prototipe, teknologi computer aided software engineering, serta application generator untuk mempercepat proses desain dan pengiriman sistem (Yen & Davis, 2019). RAD menitikberatkan pada empat komponen utama, yaitu perangkat (tools), manusia (people), metodologi (methodology), dan manajemen (management), serta secara umum meningkatkan komunikasi, kerja sama, dan komitmen pengguna dibandingkan metodologi konvensional. J. Martin membagi implementasi RAD ke dalam empat fase: perencanaan kebutuhan (requirements planning), desain pengguna (user design), konstruksi (construction), dan peralihan (cutover) (Shaker et al., 2023; Yen & Davis, 2019).

Fase perencanaan kebutuhan menghasilkan spesifikasi fungsional tingkat tinggi melalui lokakarya joint requirements planning yang disepakati seluruh pemangku kepentingan (Afriansyah et al., 2022; Shaker et al., 2023). Fase desain pengguna mengembangkan spesifikasi desain aplikasi melalui lokakarya JAD secara interaktif dan berulang, sehingga pengguna dapat memahami, memodifikasi, dan menyepakati model kerja sistem (Amras et al., 2023) . Fase konstruksi berfokus pada pengodean dan pembangunan aplikasi, tetapi pengguna tetap berpartisipasi aktif dan dapat mengusulkan perbaikan selagi antarmuka atau laporan aktual dikembangkan (Shaker et al., 2023). Fase peralihan menyerupai tahap akhir SDLC, mencakup konversi data, pengujian, peralihan ke sistem baru, dan pelatihan pengguna (Afriansyah et al., 2022). Keunggulan utama RAD adalah kecepatan pengembangan prototipe yang dapat langsung diuji serta keterlibatan pelanggan yang tinggi (Amras et al., 2023).

Kesesuaian RAD dengan penelitian ini terletak pada sifatnya yang iteratif dan responsif terhadap perubahan kebutuhan. Sebagaimana dinyatakan pada subbab 3.1, kebutuhan rekonsiliasi staf keuangan yang dipetakan secara kualitatif dapat segera ditransformasikan menjadi prototipe aplikasi Streamlit, diuji kepada pengguna, dan disempurnakan secara berulang tanpa melalui siklus dokumentasi yang panjang (Safitri & Atqiya, 2022).

2.2.7 Pengujian Black-Box Testing

Black-box testing merupakan teknik pengujian perangkat lunak yang berbasis pada spesifikasi kebutuhan, di mana perangkat lunak yang diuji dipandang

sebagai "kotak hitam" tanpa keharusan memeriksa struktur kode internal (Nidhra, 2012; Pratama & Mutiara, 2021). Pengujian dilakukan murni dari sudut pandang pengguna akhir; penguji hanya mengetahui himpunan masukan dan keluaran yang dapat diprediksi, kemudian membandingkannya dengan perilaku sistem yang diharapkan (Ehmer Khan, 2011). Teknik ini juga dikenal dengan sebutan opaque testing, functional testing, behavioral testing, atau closed-box testing karena seluruh perhatian difokuskan pada fungsionalitas Input–Proses–Output (Ehmer Khan, 2011).

Kasus uji pada black-box testing diturunkan dari spesifikasi fungsional sistem, dan fokusnya semata-mata pada keluaran yang dihasilkan sebagai respons terhadap masukan dan kondisi eksekusi yang dipilih (Larrea & Leonardo, 2017; Pezzè & Young, 2008). Keunggulan pendekatan ini antara lain memvalidasi fungsionalitas keseluruhan sistem, mengidentifikasi kebutuhan yang tidak lengkap atau tidak terprediksi, serta membutuhkan sumber daya relatif lebih sedikit dibandingkan white-box testing karena tidak menuntut pemahaman logika internal program (Nidhra, 2012; Pratama & Mutiara, 2021).

Landasan teoretis ini menjadi dasar penyusunan skenario pengujian pada subbab 3.8.1. Setiap kasus uji dalam Tabel 3.2, mulai dari pembacaan berkas unggahan, konfigurasi threshold, eksekusi pencocokan 1-to-1 dan 1-to-N, evaluasi kemiripan teks Levenshtein, hingga ekspor laporan, dirancang dengan menetapkan masukan spesifik dan memverifikasi keluaran yang dihasilkan tanpa meninjau implementasi kode algoritma *Hybrid Matching*.

جامعة الرانري
A R - R A N I R Y

BAB III
METODOLOGI PENELITIAN

3.1 Jenis dan Pendekatan Penelitian

Penelitian ini menggunakan jenis penelitian deskriptif dengan metode pengembangan sistem *Rapid Application Development* (RAD). Metode RAD diterapkan karena fleksibilitasnya dalam mempercepat pembangunan prototipe aplikasi secara interaktif, sehingga formulasi kebutuhan pencocokan data keuangan dapat langsung ditransformasikan ke dalam kode program secara cepat dan mandiri.

Dalam penerapannya, rekayasa perangkat lunak ini berfokus pada implementasi Algoritma *Hybrid Matching* berbasis bahasa pemrograman *Python* yang bekerja secara deterministik. Logika sistem dibangun dengan memadukan metode Levenshtein Distance untuk mengukur kemiripan teks deskripsi transaksi yang tidak konsisten, serta algoritma Subset Sum untuk menyelesaikan pencarian kombinasi nominal pada pola satu-ke-satu dan satu-ke-banyak. Seluruh arsitektur ini diintegrasikan ke dalam antarmuka web-app berbasis Streamlit lokal untuk menghasilkan luaran berupa laporan akhir rekonsiliasi yang informatif, lengkap dengan metrik MATCH *ID dan Similarity Score.*

3.2 Alur Penelitian

Dalam melaksanakan penelitian rekayasa perangkat lunak ini, disusun sebuah kerangka kerja *(framework)* yang sistematis agar setiap tahapan dapat berjalan secara terstruktur dan terukur. Kerangka kerja ini menjadi panduan logis dari tahap awal identifikasi masalah hingga penarikan kesimpulan. Adapun alur tahapan penelitian ini divisualisasikan melalui diagram alur *(flowchart)* pada Gambar 1 berikut:

Gambar 1 Flowchart Tahapan Penelitian

Berdasarkan Gambar 1 di atas, rangkaian tahapan penelitian diawali dengan proses identifikasi masalah dan studi literatur untuk memetakan kendala rekonsiliasi manual selama masa magang serta mengumpulkan teori pendukung dari jurnal ilmiah terkait. Tahap pengumpulan data dilakukan dengan mengambil salinan berkas catatan internal dan rekening koran mutasi bank yang telah dianonimkan. Selanjutnya, pada tahap perancangan sistem, peneliti mendesain alur logika algoritma *Hybrid Matching* untuk pola satu-ke-satu dan satu-ke-banyak, serta membuat rancangan antarmuka (*wireframe*) dasbor Streamlit.

Memasuki fase implementasi sistem, seluruh rancangan logika dan antarmuka tersebut diterjemahkan ke dalam baris kode program menggunakan bahasa *Python*. Setelah aplikasi berhasil dibangun, dilakukan pengujian sistem secara mandiri menggunakan metode *Black Box Testing* untuk memastikan keandalan fungsionalitas fitur serta akurasi pencocokannya. Penelitian ini diakhiri dengan tahap hasil dan pembahasan untuk mengevaluasi kinerja aplikasi secara keseluruhan, yang kemudian ditutup dengan penarikan kesimpulan serta saran untuk pengembangan selanjutnya.

3.3 Objek Penelitian

Objek penelitian ini adalah berkas log transaksi keuangan Rekening Perantara Aktivitas Kliring (RPAK) yang telah dianonimkan pada PT Bank Syariah Indonesia (BSI) Region I Aceh. Fokus penelitian dibatasi hanya pada lingkup data RPAK ini untuk menguji keandalan Algoritma *Hybrid Matching* dalam menyelesaikan pencocokan transaksi secara lokal.

3.4 Instrumen Penelitian

Instrumen yang digunakan dalam penelitian rekayasa perangkat lunak ini dibagi menjadi dua kategori utama, yaitu instrumen perangkat komputasi dan spesimen subjek data. Rincian dari masing-masing instrumen tersebut dijabarkan sebagai berikut:

3.4.1 Perangkat Komputasi

Eksekusi algoritma Subset Sum dan kalkulasi jarak leksikal membutuhkan lingkungan komputasi yang stabil. Spesifikasi minimum perangkat yang digunakan dalam perancangan dan pengujian sistem ini dapat dilihat pada Tabel 3.1

Tabel 3.1 Perangkat Komputasi

Kategori	**Spesifikasi**	**Keterangan / Fungsi**
Perangkat Keras (*Hardware*)	Laptop / PC Standar Operasional	*Localhost*.
	Memori (RAM)	Minimum 8 GB guna memastikan kelancaran eksekusi algoritma *Subset Sum* tanpa *memory overflow*.
Perangkat Lunak (*Software*)	Microsoft Windows 64-bit	Sistem Operasi dasar.
	Python 3.13	Bahasa pemrograman utama pembangun sistem.

	Pandas 3.0.3	*Dataframe.*
	RapidFuzz 3.14.5	Pustaka untuk komputasi jarak leksikal.
	OpenPyXL 3.1.5	Pustaka untuk membaca berkas masukan dan mengekspor laporan Excel.
	Streamlit 1.59.2	Pembangun antarmuka (*interface*) aplikasi web lokal.

3.4.2 Spesimen Data

Spesimen data yang digunakan berupa berkas digital tunggal berformat .xlsx atau .csv. Struktur data RPAK ini memuat variabel teknis berupa kolom tanggal transaksi, nominal angka, dan deskripsi referensi. Karakteristik baris datanya memuat variasi nominal makro (tunggal) dan pecahan nominal mikro yang mengandung anomali teks atau typo.

No	TGL_TRX	JAM	No Referensi	NARASI	Kode	D/K	DEBIT	KREDIT
1	15/06/2026	13:18	TT21166451V9	Penarikan Tunai\\TT21166451V9	D	-	2500000	0
2	15/06/2026	13:20	TT21166QL7MB	Penarikan Tunai\\TT21166QL7MB	D	-	25000000	0
3	15/06/2026	13:20	TT21166RP9WH	Penarikan Tunai\\TT21166RP9WH	D	-	5000000	0
4	15/06/2026	15:00	TT21168LNHB	Penarikan Tunai\\TT21168LNHB	D	-	25000000	0
5	17/06/2026	11:07	TT21168LGXMK	Penarikan Tunai\\TT21168LGXMK	D	-	25000000	0
6	17/06/2026	13:31	FT21168W8NW6	Setoran Tunai\\REIMBURSE JEGIATAN PODCAST RAMADHAN RO 1 ACEH	K	-	0	6000000
7	18/06/2026	10:52	FT21169BH38B	Pemindahbukuan\\BY INAP PKJ AN RENATA DKK ELO IT 12 SD 16 MEI 2026 DI SAFIRA HOTEL	D	-	2800000	0
8	18/06/2026	14:06	TT21169XC56V	Penarikan Tunai\\TT21169XC56V	D	-	5040000	0
9	22/06/2026	17:13	FT21173YK1XV	Setoran\\BY PENGINAPAN IT\\FT21173YK1XV	D	-	5200000	0
10	24/06/2026	9:01	TT21175N6JG7	Setoran Tunai\\TT21175N6JG7	K	-	0	7500000
11	25/06/2026	14:34	TT211769Y7XL	Penarikan Tunai\\TT211769Y7XL	D	-	3155000	0
12	25/06/2026	15:32	FT21176KZ0BG	Pemindahbukuan\\BY PERJALANAN DINAS RBC BUDIANSYAH SELAMA 12 HARI	D	-	2900000	0
13	25/06/2026	16:16	FT21176F89J5	Setoran\\PEMB BY PENGINAPAN AN BUDIANSYAH\\FT21176F89J5	D	-	1400000	0
14	25/06/2026	16:39	FT21176SXZ27	Pemindahbukuan\\BY AKOMDSI N INAP PAK SAIMUN DKK DI HTEL IBIS YOGYA 21-22 0526 D	K	-	0	1850000
15	25/06/2026	16:49	FT211766DWQN	Pemindahbukuan\\BY KEKURANGAN TRANSPORTASI PERJALANAN DINAS RBC AN BUDIANSY	D	-	240000	0
16	25/06/2026	17:15	FT21176M65ZG	Pemindahbukuan\\PNHLN KAS KECIL\\FT21176M65ZG	D	-	7500000	0
17	26/06/2026	8:30	FT21177AB12C	Setoran Tunai\\PENGADAAN ALAT TULIS KANTOR RO 1 ACEH	D	-	1250000	0
18	26/06/2026	10:15	FT21177CD34E	Pemindahbukuan\\BIAYA KONSUMSI RAPAT KORDINASI MINGGUAN	D	-	850000	0
19	28/06/2026	14:22	FT21179EF56G	Pemindahbukuan\\PENGMBALIAN DANA SISA KEGIATAN RAMADHAN	K	-	0	3500000
20	30/06/2026	11:45	FT21181GH78I	Setoran Tunai\\TMBHAN DANA OPS CABANG BNA	D	-	10000000	0
21	2/7/2026	9:10	FT21183IJ90K	Pemindahbukuan\\BY TAGIHAN LISTRIK BULAN JUNI RO 1	D	-	4100000	0
22	5/7/2026	15:20	TT21186KL12M	Penarikan Tunai\\TT21186KL12M	D	-	2500000	0
23	6/7/2026	13:40	FT21187MN34O	Pemindahbukuan\\PEMB VENDOR ATK PT SUKSES MAKMUR	K	-	0	1250000
24	6/7/2026	14:10	FT21187PQ56R	Pemindahbukuan\\PEMBAYARAN VENDOR ATK PT SUKSES MAKMUR LUNAS	K	-	0	1250000
25	7/7/2026	8:45	FT21188ST78U	Setoran Tunai\\STRN MODAL KERJA SMT 2 TAHUN 2026	D	-	50000000	0
26	7/7/2026	11:12	FT21188VW90X	Pemindahbukuan\\BY INAP PKJ PELATIHAN CS DI MEDAN AN SARI DKK	D	-	3200000	0
27	8/7/2026	9:30	TT21189YZ12A	Penarikan Tunai\\TT21189YZ12A	D	-	1500000	0
28	8/7/2026	10:05	FT21189BC34D	Setoran Tunai\\SETORAN TUNAI NASABAH VVIP BPK HAMID	K	-	0	15000000
29	9/7/2026	13:55	FT21190EF56G	Pemindahbukuan\\PENGADAAN SERAGAM KARYAWAN RO 1 BNA BAG 1	D	-	7500000	0
30	9/7/2026	14:00	FT21190HI78J	Pemindahbukuan\\PENGADAAN SERAGAM KARYAWAN RO 1 BNA BAG 2	D	-	7500000	0
31	10/7/2026	8:20	TT21191KL90M	Penarikan Tunai\\TT21191KL90M	D	-	10000000	0
32	10/7/2026	11:45	FT21191NO12P	Setoran\\REIMBURSE BIAYA OPR CABANG LHOKSEUMAWE	K	-	0	8450000
33	13/07/2026	9:15	FT21194QR34S	Pemindahbukuan\\BY MAINTENANCE SERVER DAN JARINGAN RO 1	D	-	12500000	0
34	14/07/2026	14:30	TT21195TU56V	Penarikan Tunai\\TT21195TU56V	D	-	4500000	0
35	15/07/2026	10:10	FT21196WX78Y	Pemindahbukuan\\BY TRANPORTASI PERJALANAN DINAS KACAB KE JAKARTA	D	-	5600000	0
36	16/07/2026	13:25	FT21197ZA90B	Setoran Tunai\\PENGEMBALIAN SISA PANJAR KERJA KACAB	K	-	0	400000
37	17/07/2026	15:50	FT21198CD12E	Pemindahbukuan\\PEMBAYARAN VENDOR KEBERSIHAN KANTOR BULAN JUNI	D	-	6800000	0

Mengacu pada struktur spesimen di atas, sistem akan melakukan ekstraksi (parsing) pada kolom-kolom kunci untuk menjalankan proses rekonsiliasi. Kolom TGL_TRX digunakan sebagai acuan parameter batas waktu (Temporal Bounding). Kolom NARASI menjadi subjek evaluasi pencocokan teks untuk mengabaikan karakter typo (seperti kata "JEGIATAN"). Sedangkan kolom DEBIT dan KREDIT digunakan sebagai nilai target pencarian, baik untuk pencocokan tunggal maupun pencarian kombinasi pecahan nominal operasional menggunakan algoritma Subset Sum.

Dalam praktiknya, spesimen data tersebut diuji melalui tiga gerbang logika utama, yaitu toleransi kesalahan ketik pada pencocokan teks, pencarian kombinasi nominal, dan penapisan batas jarak waktu. Berikut adalah penjelasan alur eksekusinya:

1. Pencocokan Satu-ke-Satu (1-to-1) dengan Fuzzy Matching
   Pada target pencarian transaksi tertentu (misalnya setoran qurban), sistem mencari pasangan di data mutasi. Meskipun staf operasional mengetik menggunakan singkatan kata yang tidak baku pada kolom NARASI, algoritma tetap mengevaluasi kemiripan teks tersebut. Jika hasilnya berada di atas nilai ambang batas (threshold 60%), dan nilai nominal pada kolom DEBIT/KREDIT serta tanggalnya sesuai, sistem akan menetapkannya sebagai MATCH.
2. Pencocokan Satu-ke-Banyak (1-to-N) menggunakan Subset Sum
   Ketika sistem mencari target nominal besar (misalnya Rp 6.000.000) dan tidak menemukan satupun transaksi tunggal dengan nilai tersebut di data mutasi, algoritma Subset Sum diaktifkan. Sistem akan mencari kombinasi pada kolom DEBIT/KREDIT. Apabila ditemukan bahwa gabungan dari tiga baris transaksi (misalnya masing-masing bernilai Rp 2.000.000) menghasilkan total persis Rp 6.000.000, ketiga baris tersebut kemudian diikat menjadi satu kelompok MATCH dengan transaksi target.
3. Penolakan Kesalahan Sasaran melalui Temporal BoundingApabila sistem menemukan kandidat mutasi dengan nominal yang sama persis dan narasi yang mirip, namun tanggal pada kolom TGL_TRX menunjukkan selisih waktu yang terlampau jauh (misalnya berjarak 13 hari kalender). Karena selisih waktu tersebut melanggar batas parameter Temporal Bounding yang ditetapkan (maksimal 7 hari), sistem membatalkan proses pencocokan dan memasukkan transaksi tersebut ke dalam status UNMATCHED / Review guna menghindari kesalahan identifikasi..

3.5 Pengumpulan Data

Pengumpulan data dalam penelitian ini dilakukan dengan dua teknik, yaitu studi dokumentasi dan observasi. Studi dokumentasi diterapkan dengan menyalin arsip historis log transaksi RPAK selama melaksanakan masa Kuliah Kerja Praktek (KKP) di unit Operational Network and Management. Sementaralgoritmaservasi dilakukan dengan mengamati alur kerja rekonsiliasi manual staf untuk merekam pola kesalahan pengetikan dan model kombinasi angka sebagai landasan perancangan sistem.

3.6 Perancangan Algoritma

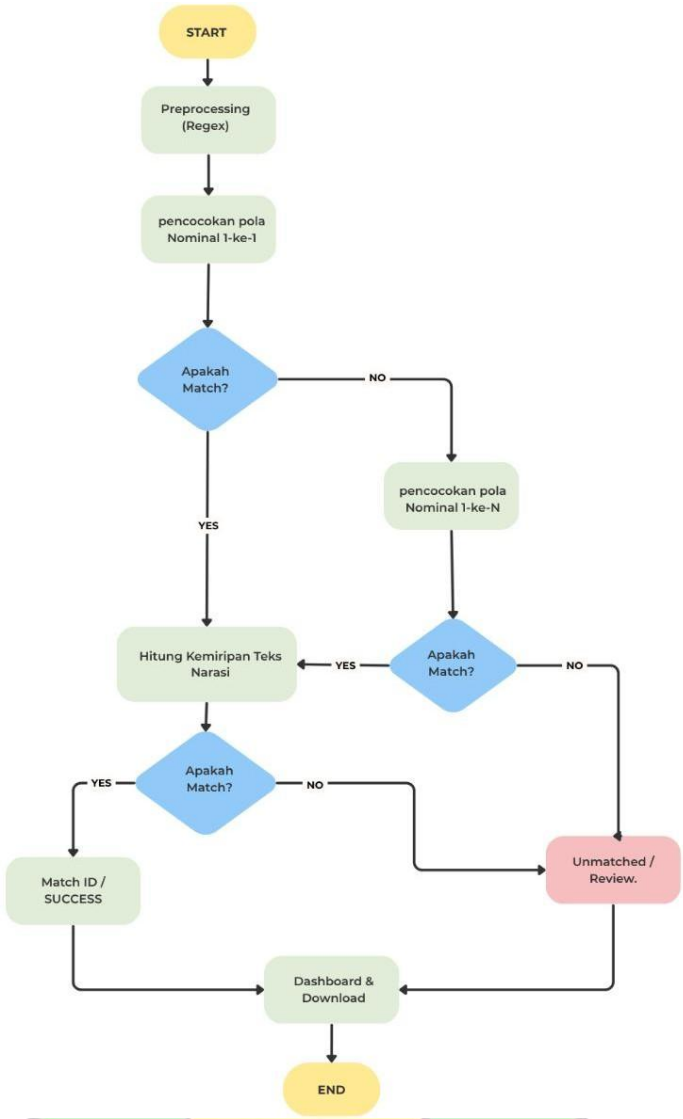
Perancangan Algoritma ini menjadi inti penggerak utama aplikasi sebelum masuk ke tahap pengodean antarmuka. Pada bagian ini, arsitektur berpikir dari Algoritma *Hybrid Matching* dibedah secara mendalam untuk memetakan bagaimana sistem memproses data secara deterministik. Penjabarannya mencakup visualisasi alur logis sistem melalui flowchart, perumusan metode Levenshtein Distance untuk penyelarasan teks deskripsi yang tidak konsisten, serta pemetaan struktur algoritma Subset Sum dalam memecahkan kombinasi nominal angka untuk pola satu-ke-satu dan satu-ke-banyak.

Algoritma *Hybrid Matching* dalam penelitian ini bukanlah modul bawaan yang siap pakai *(plug-and-play)*, melainkan sebuah arsitektur logika baru yang dirancang dan digabungkan secara mandiri oleh penulis. Meskipun metode dasar Levenshtein Distance dan Subset Sum bersumber dari teori komputasi umum, namun strategi penggabungannya, penentuan hierarki eksekusinya, serta penambahan penapis waktu *(Temporal Bounding)* merupakan kontribusi rancangan orisinal yang disesuaikan secara khusus untuk menjawab rumusan masalah rekonsiliasi data perbankan.

3.6.1 Diagram Alur Logika *Hybrid Matching*

Alur logis dari implementasi algoritma *Hybrid Matching* dalam memproses data transaksi digambarkan secara visual melalui diagram alur pada Gambar 2 berikut:

جامعة الرانري
A R - R A N I R Y



Gambar 2 Diagram Alur Logika Hybrid Matching

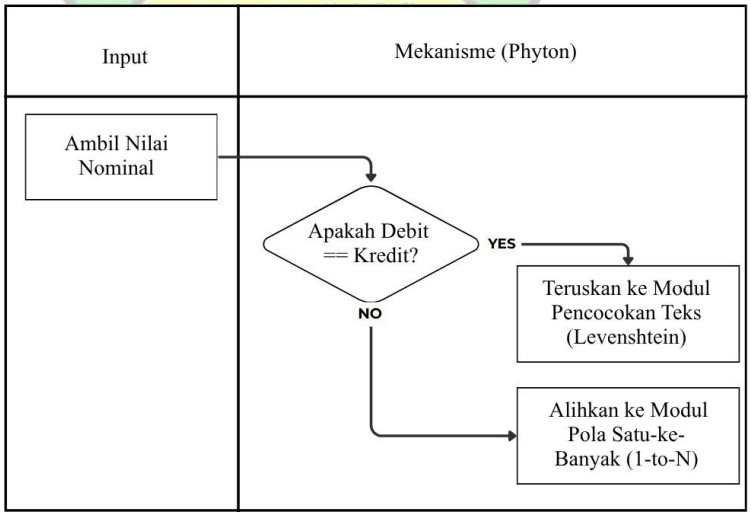
Diagram di atas merupakan penyempurnaan alur komputasi Algoritma *Hybrid Matching*. Kata "Sirkuit Evaluasi" telah diganti dengan nama tahapan yang lebih deskriptif agar alur logika terlihat lebih jelas dan komprehensif. Berikut adalah penjelasan skemanya:

4. Tahap Preprocessing: Semua data mentah masuk dan dibersihkan dari simbol atau spasi berlebih menggunakan Regex.
5. Pencocokan Pola Nominal 1-ke-1: Sistem pertama kali mencoba mencari apakah ada data internal dan data bank yang nilai uangnya sama persis (tunggal).
    - Jika TIDAK (NO), data dilempar ke tahap Subset Sum.
    - Jika YA (YES), data langsung dikirim ke tahap hitung teks (Levenshtein).

6. Pencocokan Pola Subset Sum 1-ke-N: Jika nominal tunggal tidak ditemukan, algoritma Subset Sum akan bekerja merangkai pecahan angka dari mutasi bank sampai jumlahnya pas dengan target perusahaan.
    - Jika kombinasi pecahan angkanya ketemu (YA), data dikirim ke tahap hitung teks (Levenshtein).
    - Jika kombinasi angkanya tetap tidak ketemu dalam batas waktu yang ditentukan (TIDAK), data langsung divonis gagal dan masuk ke kotak UNMATCHED / REVIEW.
7. Hitung Kemiripan Teks (Levenshtein Distance): Semua data yang nominalnya sudah cocok (baik tunggal maupun kombinasi pecahan) berkumpul di tahap ini. Sistem akan mengeja tingkat kemiripan narasinya.
8. Keputusan Akhir (Apakah Teks Match?): Ini adalah filter terakhir.Jika skor kemiripan teks memenuhi batas toleransi/threshold (YA), maka data resmi berstatus MATCH ID / SUCCESS.Jika teksnya ternyata sama sekali tidak mirip (TIDAK), maka data dibatalkan dan masuk ke kotak UNMATCHED / REVIEW.
9. Akhir dari proses ini adalah penyajian laporan pada Dashboard untuk diunduh oleh user.

3.6.2 Penyelesaian Pola Satu-ke-Satu

Modul ini merekonsiliasi satu baris data internal dengan satu baris mutasi RPAK berdasarkan kesamaan nilai nominal secara presisi. Sistem memfilter data untuk mendeteksi nominal tunggal yang identik pada kedua sumber. Jika nominal cocok, data diteruskan ke modul Pencocokan Teks untuk validasi deskripsi teks. Namun, jika terdeteksi adanya selisih nominal, baris data otomatis dialihkan ke modul pola satu-ke-banyak.

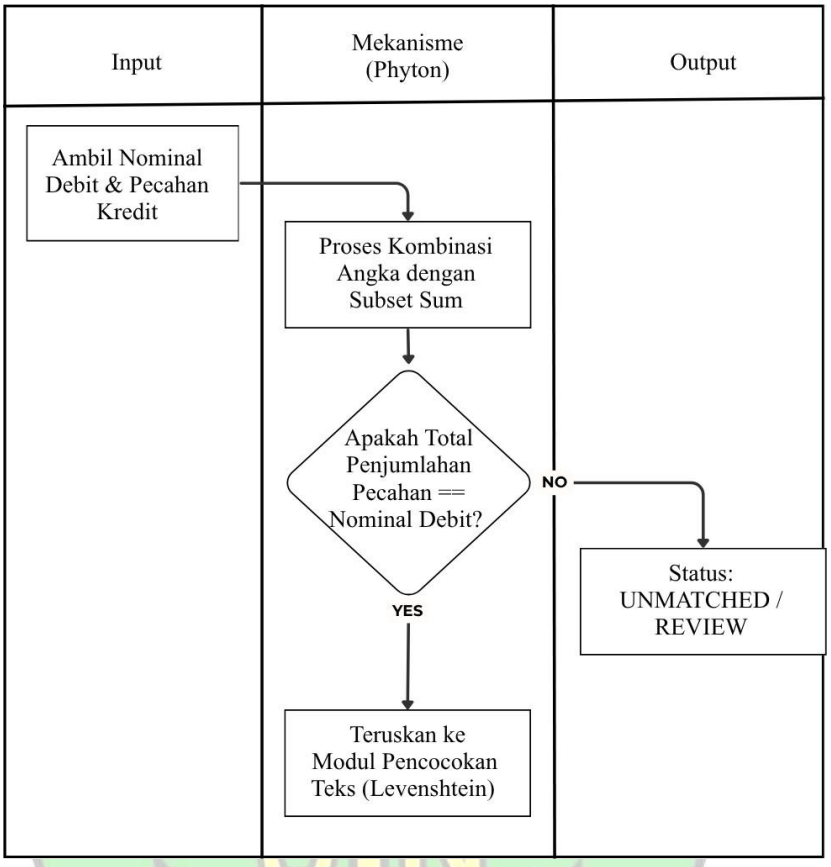


Gambar 3 Alur Pencocokan Pola Satu-ke-Satu

3.6.3 Penyelesaian Pola Satu-Ke-Banyak

Modul ini menyelesaikan kasus selisih nominal tunggal (makro) dengan mencari kombinasi beberapa pecahan nominal mikro pada data RPAK menggunakan algoritma *Subset Sum*. Sebelum operasi kombinatorial yang bersifat *NP-Hard* ini dijalankan, modul secara mandiri mengaktifkan saringan *Heuristic Search Limit Pruning.*

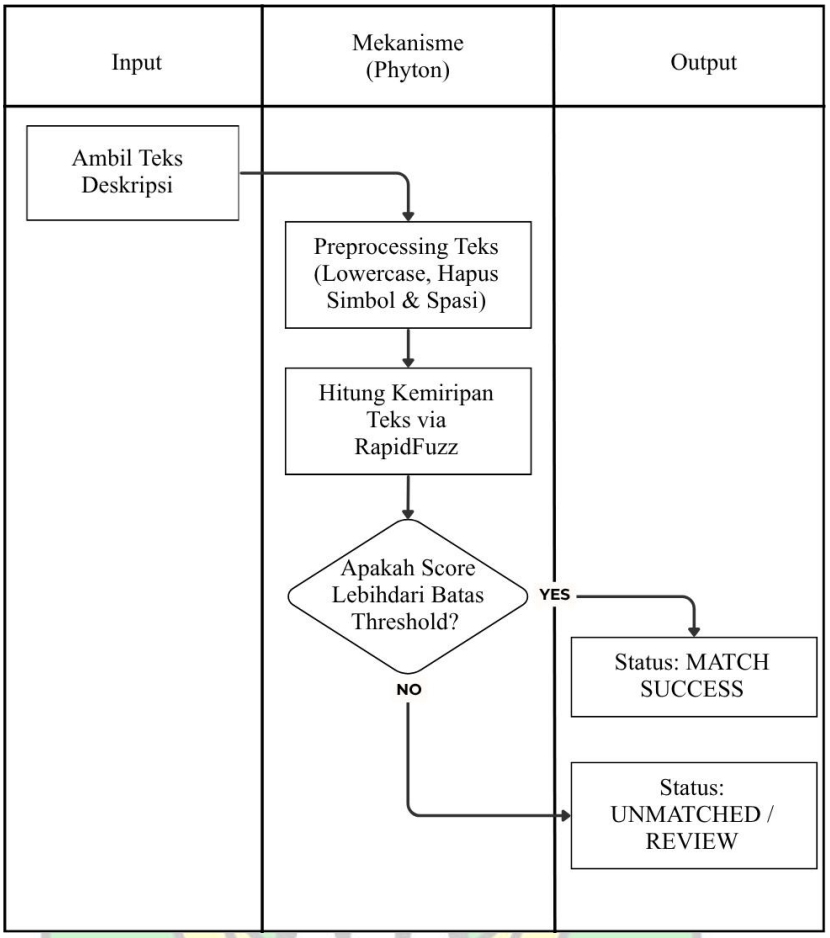
Sistem akan memblokir seluruh riwayat transaksi di luar jendela toleransi (contoh: *±3 hari* dari tanggal pencatatan internal) sehingga algoritma hanya mengalkulasi permutasi angka dari rentetan penyetoran yang terjadi pada rentang waktu kliring yang logis. Setelah probabilitas ruang pencarian dipangkas, komputer menjumlahkan variasi angka pecahan secara instan hingga nilainya sama persis dengan nominal makro. Jika kombinasi ekuivalen ditemukan, seluruh baris dibundel untuk validasi teks kolektif. Jika tidak ditemukan kombinasi yang cocok di dalam radius waktu tersebut, status ditetapkan sebagai UNMATCHED / *Review*.



Gambar 4 Alur Pola Satu-ke-Banyak

3.6.4 Pencocokan Teks

Tahap ini mengeksekusi perhitungan skor kemiripan teks deskripsi transaksi menggunakan pustaka RapidFuzz. Proses dimulai dengan pembersihan string via Regex, dilanjutkan dengan penghitungan operasi karakter minimum (sisip, hapus, ganti) antar-deskripsi. Jika nilai Similarity Score memenuhi ambang batas (*threshold*), data dinyatakan cocok (MATCH SUCCESS). Sebaliknya, jika di bawah batas, data langsung dialihkan ke status UNMATCHED / Review.



Gambar 5 Alur Pencocokan Teks

3.6.5 Logika Algoritma *Hybrid Matching*

Berdasarkan penjabaran fungsionalitas masing-masing komponen secara terpisah pada subbab-subbab sebelumnya, tahapan krusial selanjutnya adalah memadukan seluruh modul tersebut ke dalam satu arsitektur komputasi yang terintegrasi. Subbab ini menguraikan sintesis logika dari Algoritma *Hybrid Matching*, yang secara eksplisit mengilustrasikan titik penyambungan antara prasyarat penyaringan waktu, eksekusi pencarian kombinasi nominal, dan gerbang validasi akhir berupa kalkulasi jarak teks.

Guna memberikan representasi konseptual yang universal dan terstruktur sebelum diimplementasikan ke dalam sintaksis bahasa pemrograman Python, rancangan komprehensif algoritma ini diformalisasikan ke dalam wujud pseudocode berikut:

```
ALGORITHM Hybrid_Matching
INPUT: Data Internal (D), Data Bank (K), Batas_Waktu (Δ),
Batas_Teks (τ)
OUTPUT: Status Transaksi, Match_ID, Skor_Kemiripan

// TAHAP 1: PERSIAPAN DATA (Data Cleansing)
FOR EACH baris IN D dan K:
    Bersihkan huruf dan spasi berlebih pada teks narasi
    Ubah baris.Status menjadi UNMATCHED

// TAHAP 2: EKSEKUSI PENCOCOKAN
FOR EACH d IN D WHERE d.Status == UNMATCHED:
    Match_Sukses = FALSE

    // SIRKUIT I: Logika 1-ke-1 (Exact Nominal)
    Kandidat = CARI K WHERE (Status == UNMATCHED) DAN
(|Tanggal K - Tanggal d| <= Δ)
    IF Ada Kandidat dengan Nominal sama persis dengan d
THEN
        Hitung Skor_Levenshtein antara d.Narasi dan
Kandidat.Narasi
        IF Skor_Levenshtein >= τ THEN
            Tandai sebagai SUCCESS_MATCH & Buat Match_ID
            Match_Sukses = TRUE
        END IF

    // SIRKUIT II: Logika 1-ke-Banyak / Subset Sum (Jika
Sirkuit I Gagal)
    IF Match_Sukses == FALSE THEN
        FOR ukuran_kombinasi = 2 HINGGA 4:
            FOR EACH kombinasi IN BUAT_KOMBINASI(Kandidat,
ukuran_kombinasi):
                IF TOTAL_NOMINAL(kombinasi) == d.Nominal
THEN
                    Hitung Rata-Rata
Skor_Levenshtein(d.Narasi, kombinasi.Narasi)
                    IF Rata-Rata Skor >= τ THEN
                        Tandai d dan seluruh kombinasi
sebagai SUCCESS_MATCH
                        Buat Match_ID
                        Match_Sukses = TRUE
                        BREAK
                    END IF
            IF Match_Sukses == TRUE THEN BREAK
```

Agar alur pseudocode di atas dapat dipahami secara komprehensif, proses eksekusi pemecahan rekonsiliasi dijabarkan melalui empat langkah penalaran logis berikut:

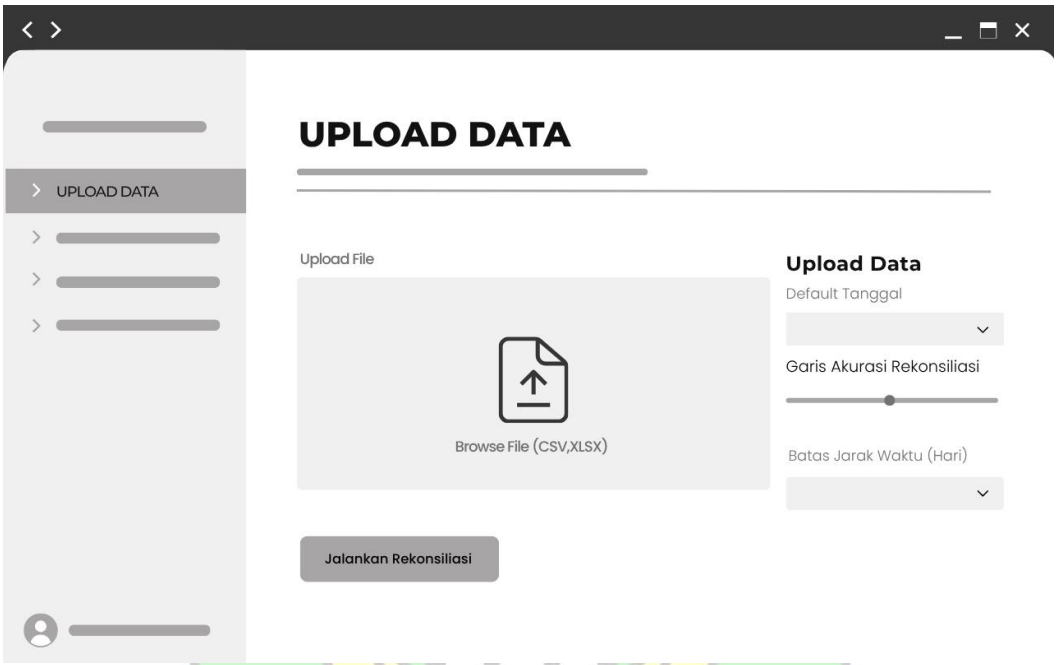
1. Merapikan Data *(Data Cleansing)*Sistem menyisir seluruh catatan, membuang simbol-simbol anomali, dan menyeragamkan karakter menjadi huruf kapital. Proses ini bertujuan mensterilkan format data agar mesin komputasi tidak mengalami miskalkulasi yang diakibatkan oleh perbedaan format spasi atau huruf kecil-besar.
2. Membatasi Area Pencarian *(Temporal Bounding)*Untuk mencegah perangkat keras mengalami kelebihan beban memori *(memory overflow)* akibat menelusuri seluruh riwayat transaksi, sistem menerapkan pembatasan pencarian hanya pada tanggal yang saling berdekatan (*batas deviasi hari).* Jika data berada di luar rentang waktu tersebut, sistem secara otomatis mengeliminasinya dari ruang pencarian kandidat.
3. Pemecahan Kombinasi Angka (*Subset Sum)*Ketika terdapat satu nominal pendanaan makro di catatan internal yang dicatat secara terpecah (parsial) pada mutasi bank, algoritma Subset Sum bertugas merangkai pecahan-pecahan angka nominal tersebut secara iteratif. Mesin menyusun kombinasi maksimal 4 baris data hingga total penjumlahannya ekuivalen secara presisi dengan angka target internal.
4. Penilaian Jarak Teks Setelah nilai nominal dinyatakan sesuai (baik secara tunggal maupun kombinasi), sistem memanggil logika Levenshtein Distance untuk mengukur jarak leksikal teks deskripsi. Jika tingkat kemiripan kata *(Similarity Score)* berada di atas ambang batas (contoh: lebihdari 80%), sistem mentoleransi kesalahan pengetikan tersebut dan mengesahkan kedua entitas data sebagai pasangan tervalidasi (SUCCESS MATCH).

3.7 Perancangan Antarmuka Pengguna

Tahap perancangan antarmuka pengguna (*User Interface*) merupakan visualisasi dari interaksi antara staf keuangan dengan sistem rekonsiliasi yang dibangun menggunakan *framework* Streamlit. Antarmuka dirancang dengan tata

letak dasbor yang minimalis, navigasi *sidebar* di sebelah kiri, dan berfokus pada fungsionalitas pengolahan data. Berikut adalah penjabaran dari masing-masing halaman antarmuka:

3.7.1 Halaman Unggah Data (Upload Data)



Gambar 6 Wireframe Halaman Upload Data

Berdasarkan gambar 6, halaman unggah data berfungsi sebagai gerbang awal sistem. Pada halaman ini, terdapat area seret dan lepas (*drag-and-drop*) atau pilih file (*browse file*) untuk mengunggah dokumen log transaksi mentah berformat .csv atau .xlsx. Di sebelah kanan area unggah, terdapat panel pengaturan parameter dinamis yang terdiri dari:

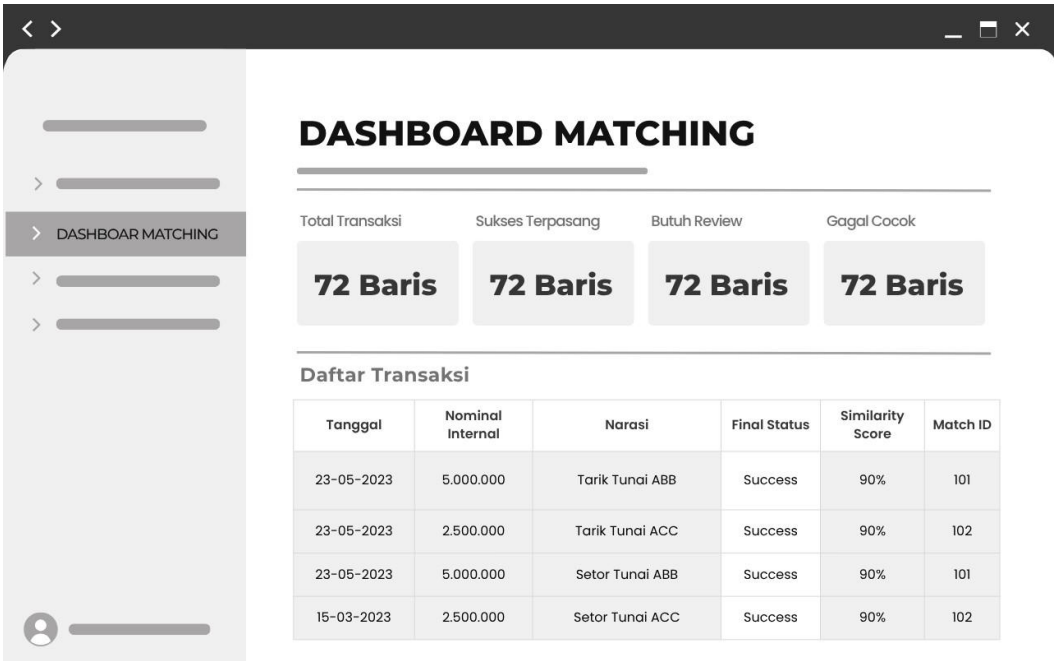
a) Default Tanggal: Untuk menetapkan basis waktu transaksi.

b) Garis Akurasi Rekonsiliasi: Sebuah *slider* untuk menentukan ambang batas persentase kemiripan teks (*Similarity Score threshold*) dari algoritma Levenshtein Distance.

c) Batas Jarak Waktu (Hari): Menu *dropdown* untuk mengaktifkan filter *Temporal Bounding* (misalnya maksimal deviasi 3 hari) guna membatasi ruang pencarian kombinasi nominal pada pola satu-ke-banyak.

Proses komputasi Algoritma *Hybrid Matching* akan dimulai ketika pengguna menekan tombol "Jalankan Rekonsiliasi".

3.7.2 Halaman Dasbor Utama (Dashboard Matching)



Gambar 7 Wireframe Halaman Dashboard Matching

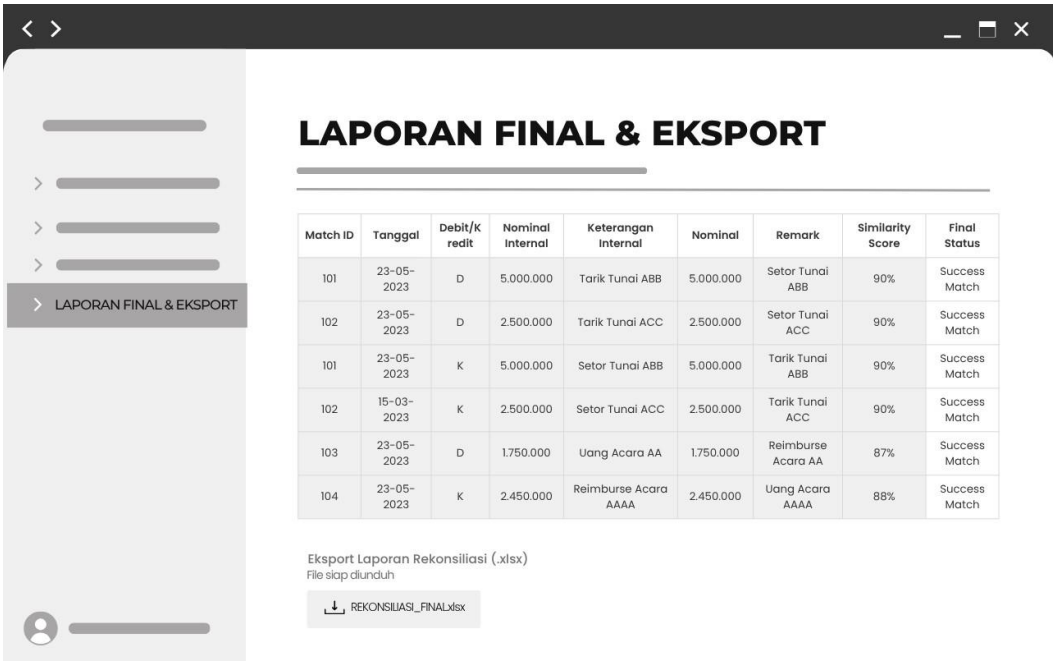
Setelah proses kalkulasi algoritma selesai, pengguna akan diarahkan ke halaman dasbor ini. Bagian atas dasbor menampilkan empat kartu informasi (*Summary Cards*) yang merangkum hasil eksekusi metrik rekonsiliasi, yaitu: Total Transaksi, Sukses Terpasang, Butuh Review, dan Gagal Cocok. Di bagian bawahnya, terdapat tabel Daftar Transaksi komprehensif yang menampilkan pemetaan data, meliputi Tanggal, Nominal Internal, Narasi, Final Status, *Similarity Score*, dan MATCH *ID* yang dihasilkan oleh mesin.

3.7.3 Halaman Panel Intervensi Manual

Gambar 8 Wireframe Halaman Panel Intervensi Manual

Halaman ini mengakomodasi fungsi *Human-in-the-Loop*, yang sistem mendelegasikan keputusan akhir kepada pengguna (staf keuangan) untuk transaksi yang berstatus "Kuning (Ragu-ragu)" atau mendekati ambang batas akurasi. Pengguna dapat mencari data spesifik melalui kolom "Search ID". Layar utama membagi informasi menjadi dua panel komparasi yang berhadap-hadapan, yaitu Keterangan Internal dan Keterangan Mutasi beserta rincian nominalnya. Di bagian bawah panel, sistem menampilkan *Similarity Score* teks, diikuti dengan dua tombol verifikasi akhir: "Setuju Loloskan *(Match)*" atau "Tolak/Pisahkan *(Unmatch)*".

3.7.4 Halaman Laporan Final dan Eksport



Gambar 9 Halaman Laporan Final dan Eksport Data

Halaman ini merupakan muara dari seluruh tahapan rekonsiliasi, baik yang diselesaikan secara otomatis oleh algoritma maupun yang telah melalui tahap intervensi manual. Halaman ini memuat tabel komprehensif yang menyandingkan secara sejajar elemen dari catatan internal (MATCH *ID*, Tanggal, Debit/Kredit, Nominal Internal, Keterangan Internal) dengan elemen dari mutasi bank (Nominal, *Remark*) beserta status akhirnya. Di bagian paling bawah, sistem menyediakan fitur "Eksport Laporan Rekonsiliasi" berupa tombol unduhan untuk mengekspor hasil penyandingan tersebut ke dalam format dokumen .xlsx (*Excel*) yang bersih dan siap digunakan untuk kebutuhan audit lebih lanjut.

3.8 Pengujian Perangkat Lunak

Pengujian perangkat lunak dilakukan untuk memastikan seluruh fitur aplikasi berjalan secara fungsional dan dapat diterima oleh pengguna akhir dengan metode Black-Box

3.8.1 Black-Box Testing

Pengujian Black-Box berfokus pada pengujian fungsionalitas antarmuka dan logika sistem berbasis Input-Process-Output tanpa melihat struktur kode internal. Pengujian ini memastikan bahwa setiap tombol, filter, dan algoritma rekonsiliasi pada aplikasi Streamlit mengeksekusi perintah dengan benar. Rancangan skenario uji Black-Box dipaparkan pada Tabel 3.2 berikut:

Tabel 3.2 Skenario Pengujian Black-Box

ID Uji	Fitur	Input	Output
TC-01	File Uploader	Mengunggah berkas RPAK (.xlsx / .csv)	Berkas berhasil dibaca dan ditampilkan pada tabel preview data.
TC-02	Pengaturan Threshold	Menggeser slider Similarity Score (misal: 80%)	Nilai batas ambang kemiripan teks diperbarui secara dinamis pada engine.
TC-03	Eksekusi Matching 1-to-1	Menekan tombol pemrosesan rekonsiliasi	Sistem berhasil mencocokkan transaksi dengan nilai nominal tunggal yang identik.
TC-04	Eksekusi Matching 1-to-N	Menekan tombol eksekusi Subset Sum	Sistem berhasil memecah nominal makro dan menemukan kombinasi nominal mikro yang sesuai.
TC-05	Evaluasi Teks (Levenshtein)	Otomatis setelah sirkuit nominal	Sistem memberikan persentase skor kemiripan deskripsi teks dan melabeli status MATCH SUCCESS / UNMATCHED.

TC-06	Ekspor Laporan	Menekan tombol Download hasil rekonsiliasi	Sistem mengunduh berkas laporan akhir (.xlsx) berisi status MATCH secara lengkap.

جامعة الرانري
AR-RANIRY

BAB IV
IMPLEMENTASI SISTEM

4.1 Gambaran Umum Implementasi Sistem

Sistem dikembangkan menggunakan bahasa pemrograman Python dengan framework Streamlit sebagai antarmuka aplikasi berbasis web yang dijalankan pada lingkungan lokal (localhost). Sistem terdiri atas bagian front-end dan back-end. Bagian front-end menggunakan Streamlit untuk menangani proses unggah data, konfigurasi parameter pencocokan, penyajian hasil rekonsiliasi, intervensi manual, dan ekspor laporan. Sementara itu, bagian back-end menangani pra-pemrosesan data dan proses *Hybrid Matching* dengan memanfaatkan Pandas untuk pengolahan data serta RapidFuzz untuk mengukur kemiripan narasi transaksi. Hasil proses pencocokan kemudian ditampilkan pada dashboard, dapat ditinjau melalui panel intervensi manual, dan selanjutnya diekspor menjadi laporan akhir.

4.2 Implementasi Algoritma *Hybrid Matching* (Back-End)

Implementasi algoritma *Hybrid Matching* pada sisi back-end dilakukan secara bertahap melalui pra-pemrosesan data, pembatasan kandidat berdasarkan rentang waktu (temporal bounding), pengukuran kemiripan narasi, pencocokan nominal satu-ke-satu (1-to-1), dan pencarian kombinasi transaksi satu-ke-banyak (1-to-N). Proses tersebut menggunakan tiga atribut utama transaksi, yaitu tanggal, narasi, dan nominal, untuk menentukan transaksi yang dapat direkonsiliasi.

4.2.1 Pra-pemrosesan Data (Data Cleansing)

Tahap pra-pemrosesan dilakukan untuk menyeragamkan data transaksi sebelum memasuki proses pencocokan. Implementasi utama tahap ini terdapat pada modul core/preprocessor.py. Normalisasi narasi dilakukan menggunakan Regular Expression (Regex) dengan menghilangkan karakter selain huruf, angka, dan spasi, merapikan spasi berlebih, serta mengubah seluruh karakter menjadi huruf kapital. Proses normalisasi dapat direpresentasikan sebagai

$$C(x)=\mathrm{Upper}(\mathrm{Trim}(\mathrm{CollapseSpace}(\mathrm{ReplaceSymbol}(x))))$$

Sebagai contoh, narasi “Pembayaran Hotel-Saimun, 2024” dinormalisasi menjadi “PEMBAYARAN HOTEL SAIMUN 2024”. Normalisasi tersebut mengurangi perbedaan format penulisan yang tidak memiliki pengaruh terhadap identitas transaksi.

Selain narasi, sistem mengonversi data tanggal ke format tanggal yang seragam serta menangani nilai kosong pada atribut debit dan kredit. Selanjutnya, transaksi debit dan kredit dipisahkan dan nilai nominal digunakan dalam bentuk absolut agar proses pencocokan membandingkan besaran transaksi tanpa dipengaruhi tanda debit atau kredit. Pada tahap awal, setiap transaksi juga diberikan status UNMATCHED sebelum diproses oleh algoritma pencocokan.

```
def clean_text(text):
    if pd.isna(text):
        return ""
    text = re.sub(r'[^a-zA-Z0-9\s]', ' ', str(text))
    return re.sub(r'\s+', ' ', text).strip().upper()
```

Gambar 10 Potongan Kode Pra-pemrosesan Data (Data Cleansing)

4.2.2 Eksekusi Pencocokan Pola Satu-ke-Satu (1-to-1)

Pencocokan pola satu-ke-satu *(1-to-1)* dilakukan untuk menemukan satu transaksi kredit yang memiliki nominal sama dengan satu transaksi debit target. Suatu pasangan dinyatakan memenuhi kriteria apabila transaksi kredit masih berstatus UNMATCHED, berada dalam batas selisih tanggal yang ditentukan, memiliki similarity score narasi yang sama dengan atau melebihi threshold, serta mempunyai nominal yang sama setelah dibulatkan hingga dua angka desimal. Secara matematis, kondisi tersebut dapat dinyatakan sebagai:

1. Validasi Ketersediaan : $\text{Status}(k_j) = \text{UNMATCHED}$

```
for i, d_row in df_debit[df_debit['STATUS'] == 'UNMATCHED'].iterrows():
    target_val = round(d_row['NOMINAL'], 2)
    target_date = d_row['TGL_TRX_CLEAN']
    target_text = str(d_row['NARASI_CLEAN'])
```

Gambar 11 Potongan Kode Validasi Ketersediaan

2. Validasi Temporal : $(|T_{k_j} - T_{d_i}| \leq \Delta)$
3. Validasi Leslikal : $(S(d_i, k_j) \geq \tau)$
4. Validasi Nominal :$(\text{round}(M_{d_i}, 2) = \text{round}\ (M_{k_j}, 2))$

```
exact_match = candidates[candidates['NOMINAL'].round(2) == target_val]
if not exact_match.empty:
    match_id = generate_match_id()
    best_idx = exact_match.index[0]
    match_score = exact_match.at[best_idx, 'SCORE']
```

Gambar 12 Potongan Kode Pencocokan Nominal 1-to-1

Apabila seluruh kondisi terpenuhi, sistem memberikan MATCH_ID yang sama kepada kedua transaksi dan mengubah statusnya menjadi SUCCESS MATCH. Jika kandidat telah memenuhi kriteria temporal dan kemiripan narasi tetapi nominalnya berbeda, transaksi tersebut tidak langsung dianggap gagal, melainkan dapat diteruskan ke proses pencocokan satu-ke-banyak (*1-to-N*) untuk mengevaluasi kemungkinan pembentukan nominal target dari kombinasi beberapa transaksi kredit.

Tabel 4.1 Simulasi Pencarian Kombinasi Nominal Satu-ke-Satu

No.	Tanggal Debit	Nominal Debit	Tanggal Kredit	Nominal Kredit	Selisih Hari	Similarity Score	Δ	Threshold	Hasil Evaluasi
1	15/06/2026	Rp5.200.000	15/06/2026	Rp5.200.000	0 hari	100,00	7 hari	60	SUCCESS MATCH
2	15/06/2026	Rp5.200.000	18/06/2026	Rp5.200.000	3 hari	82,00	7 hari	60	SUCCESS MATCH
3	15/06/2026	Rp5.200.000	24/06/2026	Rp5.200.000	9 hari	–	7 hari	60	Tidak lolos validasi temporal
4	15/06/2026	Rp5.200.000	17/06/2026	Rp5.000.000	2 hari	88,00	7 hari	60	Dialihkan ke evaluasi *1-to-N*
5	15/06/2026	Rp5.200.000	16/06/2026	Rp5.200.000	1 hari	55,00	7 hari	60	Tidak lolos *threshold* narasi
6	15/06/2026	Rp5.200.000	16/06/2026	Rp5.200.000	1 hari	70,00	7 hari	60	SUCCESS MATCH

Berdasarkan Tabel 4.1, keputusan pencocokan tidak hanya ditentukan oleh kesamaan nominal. Sistem terlebih dahulu memastikan bahwa kandidat masih tersedia, berada dalam rentang tanggal yang diperbolehkan, dan memiliki tingkat kemiripan narasi yang memenuhi *threshold*. Tahapan tersebut membantu mengurangi kemungkinan pencocokan terhadap transaksi lain yang kebetulan memiliki nominal yang sama.

4.2.3 Penerapan Pemangkasan Cabang Heuristik (Temporal Bounding)

Pencarian kombinasi pada pola satu-ke-banyak berpotensi menghasilkan ruang pencarian yang besar apabila seluruh transaksi kredit diperiksa tanpa pembatasan. Oleh karena itu, sistem menerapkan pemangkasan kandidat secara

heuristik sebelum proses kombinasi dilakukan. Kandidat terlebih dahulu dibatasi berdasarkan rentang waktu ($T_{min} \leq T_k \leq T_{max}$) dengan ($T_{min} = T_d - \Delta$) dan ($T_{max} = T_d + \Delta$), kemudian transaksi dengan similarity score di bawah threshold dieliminasi. Setelah penyaringan tersebut, pencarian kombinasi hanya dilakukan terhadap (m) kandidat yang lolos dan dibatasi maksimal empat transaksi menggunakan `itertools.combinations`. Dengan batas ukuran kombinasi tersebut, jumlah kombinasi yang perlu dievaluasi dapat dinyatakan sebagai:

$$Q(m) = \sum_{r=2}^{\min(m,4)} \binom{m}{r}$$

```
date_mask = (df_kredit['STATUS'] == 'UNMATCHED') & \
            (df_kredit['TGL_TRX_CLEAN'] >= target_date - pd.Timedelta(days=max_days)) & \
            (df_kredit['TGL_TRX_CLEAN'] <= target_date + pd.Timedelta(days=max_days))
```

Gambar 13 Potongan Kode Pemangkasan Cabang Heuristik (Temporal Bounding)

Pembatasan ini menghindari eksplorasi seluruh kemungkinan subset yang secara teoritis dapat berkembang secara eksponensial hingga $O(2^n)$. Karena ukuran kombinasi dibatasi maksimal empat elemen, kompleksitas pencarian kombinasi terhadap kandidat yang telah lolos penyaringan memiliki batas atas $O(m^4)$. Dengan demikian, temporal bounding, penyaringan berdasarkan similarity threshold, dan pembatasan ukuran kombinasi berfungsi memperkecil ruang pencarian sebelum proses pencocokan nominal satu-ke-banyak dilakukan.

4.2.4 Eksekusi Pencocokan Pola Satu-ke-Banyak (Subset Sum 1-to-N)

Transaksi debit target ((P)) yang belum memperoleh pasangan pada proses pencocokan satu-ke-satu selanjutnya dialihkan ke pencocokan pola satu-ke-banyak (1-to-N). Sistem membentuk kombinasi dari kandidat kredit yang telah lolos tahap penyaringan sebelumnya dengan jumlah anggota antara dua sampai empat transaksi, $2 \leq |I| \leq 4$ Suatu kombinasi dinyatakan sesuai apabila jumlah nominal seluruh transaksi kredit di dalamnya sama dengan nominal debit target:

$$\text{round}\left(\sum_{i \in I} M_{k_i}, 2\right) = \text{round}(P, 2)$$

```
found_combo = False
for r in range(2, min(len(candidates) + 1, max_combo + 1)):
    for combo in combinations(candidates.index, r):
        combo_sum = round(sum(df_kredit.loc[list(combo), 'NOMINAL']), 2)
        if combo_sum == target_val:
            match_id = generate_match_id()
```

Gambar 14 Potongan Kode Pencarian Kombinasi Nominal (Subset Sum 1-to-N)

Pencarian dilakukan secara iteratif mulai dari kombinasi terkecil. Ketika kombinasi pertama yang memenuhi target ditemukan, transaksi debit dan seluruh kredit yang terlibat diberikan MATCH_ID yang sama. Nilai SIMILARITY_SCORE pada transaksi debit kemudian dihitung menggunakan rata-rata skor kemiripan dari seluruh transaksi kredit dalam kombinasi:

$$S_d = \frac{1}{|I|} \sum_{i \in I} S(d, k_i)$$

```
avg_score = candidates.loc[list(combo), 'SCORE'].mean()
df_debit.at[i, 'SIMILARITY_SCORE'] = round(avg_score, 2)

for idx in combo:
    df_kredit.at[idx, 'STATUS'] = 'SUCCESS MATCH'
    df_kredit.at[idx, 'MATCH_ID'] = match_id
    df_kredit.at[idx, 'SIMILARITY_SCORE'] = round(candidates.at[idx, 'SCORE'], 2)
```

Gambar 15 Potongan Kode Pembaruan Status dan Skor Kombinasi 1-to-N

Mekanisme tersebut memungkinkan satu transaksi debit direkonsiliasi dengan beberapa transaksi kredit ketika nominalnya tidak dapat ditemukan melalui pencocokan satu-ke-satu. Transaksi kredit yang tidak termasuk dalam kombinasi terpilih tetap berstatus UNMATCHED sehingga masih dapat digunakan sebagai kandidat pada proses pencocokan transaksi debit berikutnya.

Target transaksi debit:

P=Rp7.500.000

Kandidat kredit yang telah memenuhi radius tanggal dan threshold narasi:

Tabel 4.2 Simulasi Pencarian Kombinasi Nominal Satu-ke-Banyak

Kode Kandidat	**Nominal Kredit**	**Narasi Kredit**	***Similarity Score***
(a_1)	Rp2.500.000	BANTUAN DANA 1	92,31
(a_2)	Rp2.500.000	BANTUAN DANA 2	92,31
(a_3)	Rp2.500.000	BANTUAN DANA 3	92,31
(a_4)	Rp1.000.000	TRANSFER TAMBAHAN	65,00

Proses pencarian kombinasi dapat diringkas sebagai berikut:

$$a_1 + a_2 = \text{Rp}5.000.000 \neq \text{P}$$

$$a_1 + a_3 = \text{Rp}5.000.000 \neq \text{P}$$

$$\text{kombinasi dua transaksi lainnya} \neq \text{P}$$

$$a_1 + a_2 + a_3 = \text{Rp}7.500.000 = \text{P}$$

Kombinasi (a_1+a_2+a_3) kemudian ditetapkan sebagai SUCCESS MATCH dan diberikan MATCH_ID yang sama dengan transaksi debit target. Nilai skor agregat debit adalah:

$$S_d = \frac{92{,}31 + 92{,}31 + 92{,}31}{3} = 92{,}31$$

Sementara itu, (a_4) tidak termasuk dalam kombinasi yang terpilih sehingga tetap berstatus UNMATCHED dan masih dapat digunakan sebagai kandidat untuk transaksi debit berikutnya.

4.2.5 Implementasi Pencocokan Teks dan Evaluasi Similarity Score

Evaluasi kemiripan narasi dilakukan menggunakan fungsi token_set_ratio dari pustaka RapidFuzz terhadap narasi transaksi yang telah melalui tahap pembersihan. Pendekatan berbasis token digunakan agar perbandingan tidak hanya bergantung pada susunan kata secara identik, tetapi juga mempertimbangkan kesamaan token yang terdapat pada kedua narasi. Setiap pasangan menghasilkan SIMILARITY_SCORE pada rentang 0 sampai 100, dengan nilai yang semakin tinggi menunjukkan tingkat kemiripan teks yang semakin besar. Nilai tersebut kemudian dibandingkan dengan parameter ambang ((τ)) yang ditentukan pengguna. Kandidat hanya dipertahankan apabila skor kemiripannya sama dengan atau lebih besar daripada nilai threshold, yang dapat dirumuskan sebagai:

$$Eligible(A,B) = \{ \begin{matrix} 1 S(A,B) \geq \tau \\ 0 S(A,B) < \tau \end{matrix}$$

Dengan (S(A,B)) merupakan skor kemiripan antara narasi target (A) dan narasi kandidat (B). Mekanisme ini memungkinkan sistem mempertahankan transaksi dengan variasi susunan kata atau perbedaan penulisan tertentu, sekaligus mengeliminasi kandidat yang tingkat kemiripan narasinya berada di bawah batas yang ditentukan.

```
# Only keep candidates above threshold OR exact exact matches (to avoid false positives)
candidates = candidates[candidates['SCORE'] >= threshold]
```

Gambar 16 Potongan Kode Evaluasi Similarity Score

Tabel 4.3 Simulasi Evaluasi Similarity Score Berdasarkan Threshold

No.	Narasi Target	Narasi Kandidat	Similarity Score	Threshold	Hasil
1	BY PENGINAPAN IT	BY PENGINAPAN IT	100,00	80	Diterima
2	TRANSFER BIAYA HOTEL	BIAYA HOTEL TRANSFER	100,00	80	Diterima
3	BIAYA HOTEL SAIMUN	BAYAR TAGIHAN HOTEL SAIMUN	80,00	80	Diterima
4	PEMBAYARAN CICILAN AHMAD	PMB CCLN AHMD	70,27	80	Ditolak
5	PEMBAYARAN CICILAN AHMAD	PMB CCLN AHMD	70,27	60	Diterima
6	BANTUAN DANA PEMB FAK EKON	BANTUAN DANA 1	92,31	80	Diterima
7	PEMBAYARAN VENDOR ABC	PEMBAYARAN VENDOR ABC TAMBAHAN	100,00	80	Diterima

Berdasarkan Tabel 4.3, perubahan nilai threshold memengaruhi tingkat selektivitas pencocokan narasi. Threshold yang lebih tinggi memperketat penerimaan kandidat, sedangkan nilai yang lebih rendah memberikan toleransi lebih besar terhadap perbedaan penulisan. Oleh karena itu, SIMILARITY_SCORE digunakan sebagai mekanisme penyaringan kandidat dan bukan sebagai satu-satunya dasar penetapan hasil rekonsiliasi.Secara keseluruhan, implementasi Algoritma *Hybrid Matching* membentuk urutan validasi sebagai berikut:

Data Cleansing→Temporal Bounding→Text Thresholding→Exact Nominal 1-to-1→Subset Sum 1-to-N→MATCH ID dan Status

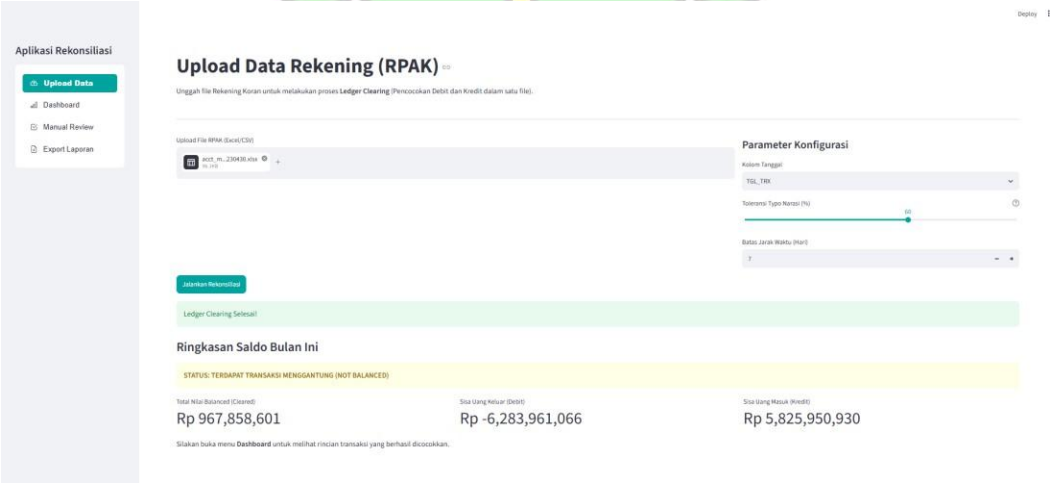
Urutan tersebut menyebabkan beban komputasi kombinatorial hanya diaktifkan setelah kandidat yang tidak relevan disingkirkan. Hasilnya, sistem dapat menangani variasi narasi sekaligus menyelesaikan perbedaan struktur nominal

transaksi tanpa menghilangkan kemampuan pengguna untuk memverifikasi transaksi yang tetap berstatus UNMATCHED.

4.3 Implementasi Antarmuka Sistem *(Front-End)*

Antarmuka pengguna sistem dibangun menggunakan framework Streamlit untuk menghubungkan pengguna dengan seluruh proses rekonsiliasi. Navigasi utama ditempatkan pada sidebar yang terdiri atas empat menu, yaitu Upload Data, Dashboard, Manual Review, dan Export Laporan. Perpindahan antarhalaman dikelola melalui modul tampilan yang terpisah, sedangkan hasil proses rekonsiliasi disimpan menggunakan session_state agar tetap tersedia selama sesi aplikasi berlangsung. Dengan mekanisme tersebut, pengguna dapat berpindah dari proses unggah data menuju dashboard, peninjauan manual, dan ekspor laporan tanpa menjalankan kembali proses rekonsiliasi.

4.3.1 Halaman Unggah Data dan Konfigurasi



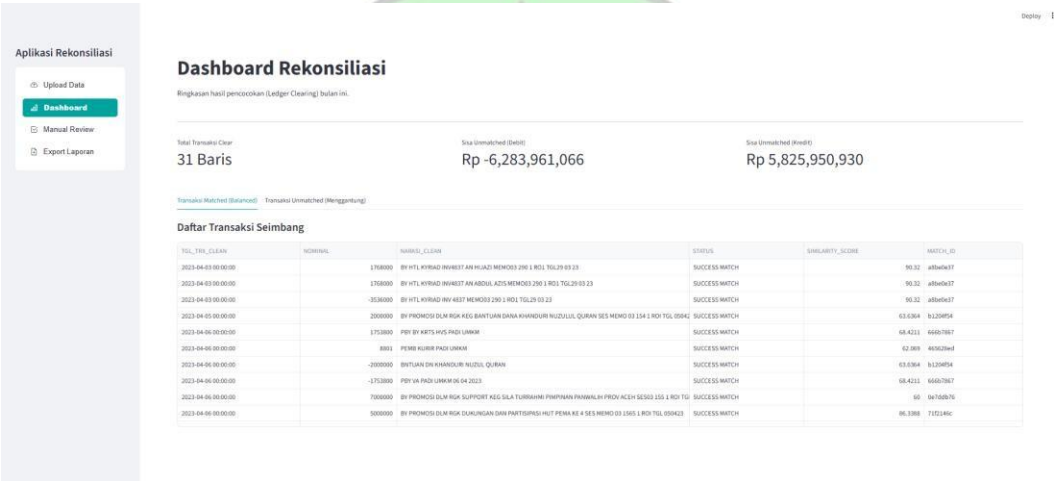
Gambar 17 Tampilan Implementasi Halaman Unggah Data RPAK

Halaman Upload Data Rekening (RPAK) merupakan halaman awal untuk memasukkan data transaksi dan menentukan parameter rekonsiliasi. Seperti ditunjukkan pada Gambar 17, halaman terdiri atas area unggah berkas dan panel Parameter Konfigurasi. Sistem menerima data transaksi dalam format .csv dan .xlsx, kemudian pengguna menentukan kolom tanggal sebagai acuan temporal. Pada panel konfigurasi, parameter Toleransi Typo Narasi (%) digunakan untuk menentukan nilai minimum similarity threshold, sedangkan Batas Jarak Waktu (Hari) menentukan maksimum selisih tanggal antara transaksi debit target dan kandidat kredit. Setelah konfigurasi ditetapkan, tombol Jalankan Rekonsiliasi digunakan untuk memulai tahap pra-pemrosesan dan eksekusi algoritma *Hybrid*

Matching. Hasil proses selanjutnya disimpan dalam session_state dan dapat ditinjau melalui halaman Dashboard.

4.3.2 Halaman Dashboard Matching

Setelah proses rekonsiliasi dijalankan, hasil pencocokan ditampilkan pada halaman Dashboard Rekonsiliasi seperti pada Gambar 18. Halaman ini menyajikan ringkasan hasil berupa jumlah transaksi yang berhasil direkonsiliasi, sisa transaksi debit yang berstatus UNMATCHED, dan sisa transaksi kredit yang berstatus UNMATCHED. Informasi tersebut memberikan gambaran awal mengenai hasil pencocokan tanpa mengharuskan pengguna memeriksa seluruh transaksi secara individual.komputasi selesai, hasil pemetaan data divisualisasikan pada halaman Dashboard Rekonsiliasi (Gambar 11). Halaman ini merangkum metrik performa algoritma melalui tiga komponen kartu ringkasan: Total Transaksi Clear, Sisa



Gambar 18 Tampilan Implementasi Halaman Dashboard Rekonsiliasi

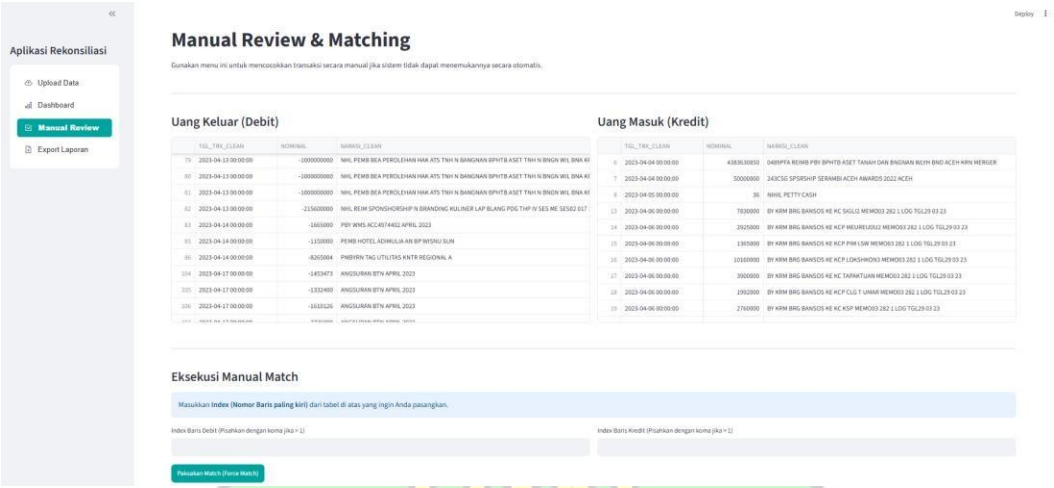
جامعة الرانري
AR-RANIRY

Detail hasil selanjutnya dipisahkan menjadi kelompok transaksi matched dan UNMATCHED. Kelompok matched menampilkan transaksi yang telah memperoleh MATCH_ID beserta SIMILARITY_SCORE untuk menunjukkan hubungan dan nilai kemiripan narasi pada hasil pencocokan. Sementara itu, kelompok UNMATCHED memuat transaksi yang belum memperoleh pasangan sesuai kriteria algoritma dan dapat ditindaklanjuti melalui halaman Manual Review.

4.3.3 Halaman Panel Intervensi Manual

Halaman Manual Review & Matching mengadopsi konsep Human-in-the-Loop untuk menangani transaksi yang belum berhasil dicocokkan secara otomatis. Seperti ditunjukkan pada Gambar 19, transaksi berstatus UNMATCHED ditampilkan secara paralel dalam dua kelompok, yaitu Uang Keluar (Debit) dan

Uang Masuk (Kredit). Pengguna dapat meninjau tanggal, nominal, dan narasi transaksi, kemudian memilih indeks baris debit dan kredit yang dinilai berkaitan melalui bagian Eksekusi Manual Match. Setelah konfirmasi dilakukan, sistem membentuk MATCH_ID baru untuk seluruh transaksi yang dipilih dan mengubah statusnya menjadi SUCCESS (MANUAL) sebagai penanda bahwa proses pencocokan dilakukan melalui intervensi pengguna.

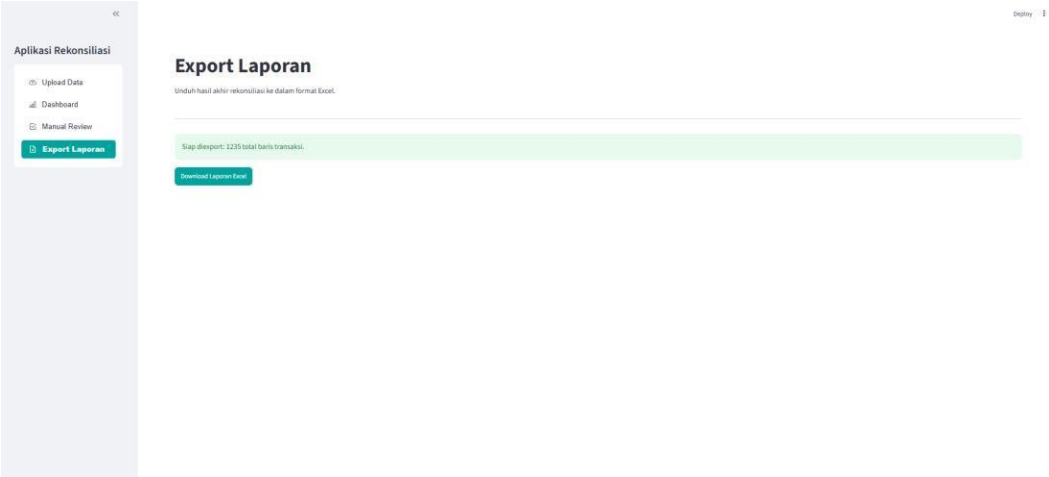


Gambar 19 Tampilan Implementasi Halaman Manual Review dan Matching

جامعة الرانري
AR-RANIRY

4.3.4 Halaman Laporan Final dan Ekspor

Halaman Export Laporan merupakan tahap akhir penyajian hasil rekonsiliasi. Seperti ditunjukkan pada Gambar 20, halaman ini menampilkan jumlah keseluruhan baris transaksi yang siap diekspor dan menyediakan tombol Download Laporan Excel. Data hasil rekonsiliasi yang tersimpan pada session_state dikonversi menjadi berkas Microsoft Excel menggunakan Pandas dengan dukungan *engine* openpyxl. Berkas hasil diberi nama Hasil_Rekonsiliasi_BSI.xlsx dan dibagi menjadi tiga lembar kerja agar transaksi dapat ditinjau berdasarkan status hasil rekonsiliasinya.



Gambar 20 Tampilan Implementasi Halaman Export Laporan

Tabel 4.4 Struktur Lembar Kerja pada Laporan Hasil Rekonsiliasi

No.	Nama Lembar Kerja	Isi Data	Fungsi
1	Semua Transaksi	Seluruh transaksi hasil pemrosesan	Menyediakan laporan lengkap tanpa pemisahan status.
2	Matched	Transaksi berstatus SUCCESS MATCH dan SUCCESS (MANUAL)	Menampilkan transaksi yang telah dinyatakan seimbang.
3	UNMATCHED (Selisih)	Transaksi berstatus UNMATCHED	Menampilkan transaksi menggantung yang masih membutuhkan tindak lanjut.

Data hasil ekspor tetap memuat atribut transaksi beserta atribut hasil rekonsiliasi, seperti STATUS, MATCH_ID, dan SIMILARITY_SCORE. Berkas Excel dibentuk pada saat proses ekspor dilakukan dan kemudian tersedia untuk diunduh melalui tombol yang disediakan pada antarmuka.

4.4 Pengujian Sistem

Tahap pengujian dilakukan untuk memastikan bahwa fungsi utama aplikasi bekerja sesuai dengan kebutuhan yang telah ditetapkan. Pengujian menggunakan dua pendekatan, yaitu Black Box Testing untuk mengevaluasi kesesuaian fungsi berdasarkan masukan dan keluaran sistem.

4.4.1 Pengujian Fungsionalitas (*Black Box Testing*)

Pengujian Black-Box dilakukan tanpa meninjau struktur internal kode program dan berfokus pada respons sistem terhadap masukan yang diberikan pengguna. Pengujian mencakup proses unggah data, perubahan parameter similarity threshold dan batas temporal, pencocokan satu-ke-satu, pencarian kombinasi satu-ke-banyak, evaluasi kemiripan narasi, proses manual matching, serta ekspor laporan. Skenario pengujian dan hasil yang diperoleh disajikan pada Tabel 4.6.

جامعة الرانري
AR-RANIRY

Tabel 4.5 Skenario dan Hasil Uji Black Box Testing

No	Fitur yang Diuji	Skenario Pengujian	Hasil yang Diharapkan	Hasil Pengujian	Status
1	Unggah Data	Pengguna mengunggah berkas transaksi berformat .xlsx atau .csv	Sistem dapat membaca data dan menampilkan konfigurasi rekonsiliasi	Berkas berhasil dibaca dan dapat diproses	Berhasil
2	Similarity Threshold	Kandidat memiliki *similarity score* 70 dengan *threshold* 80	Kandidat tidak diteruskan sebagai kandidat pencocokan	Kandidat tidak memenuhi *threshold*	Berhasil
3	Temporal Bounding	Selisih tanggal transaksi sebesar 9 hari dengan batas maksimum 7 hari	Kandidat dikeluarkan dari ruang pencarian	Kandidat tidak diproses pada pencocokan target tersebut	Berhasil
4	Matching 1-to-1	Debit Rp5.200.000 memiliki satu kandidat kredit Rp5.200.000 yang memenuhi kriteria tanggal dan narasi	Kedua transaksi memperoleh MATCH_ID yang sama dan status SUCCESS_MATCH	Sistem menghasilkan pasangan satu-ke-satu dengan MATCH_ID yang sama	Berhasil
5	Matching 1-to-N	Debit Rp7.500.000 memiliki tiga kandidat kredit masing-masing Rp2.500.000	Sistem menemukan kombinasi Rp2.500.000 + Rp2.500.000 + Rp2.500.000 = Rp7.500.000	Ketiga kredit dan satu debit dikelompokkan dalam MATCH_ID yang sama	Berhasil

6	Kemiripan Narasi	Narasi TRANSFER BIAYA HOTEL dibandingkan dengan BIAYA HOTEL TRANSFER	Sistem memberikan skor kemiripan tinggi meskipun urutan token berbeda	Kandidat memperoleh *similarity score* yang memenuhi *threshold*	Berhasil
7	Manual Review	Pengguna memilih indeks debit dan kredit berstatus UNMATCHED , kemudian menjalankan *manual match*	Sistem membentuk MATCH_ID baru dan mengubah status menjadi SUCCESS (MANUAL)	Transaksi berhasil dikelompokkan melalui proses manual	Berhasil
8	Dashboard	Proses rekonsiliasi selesai dijalankan	Sistem menampilkan jumlah transaksi *clear*, nominal debit dan kredit UNMATCHED, serta detail transaksi	Ringkasan dan tabel hasil tampil pada Dashboard	Berhasil
9	Export Laporan	Pengguna menekan tombol **Download Laporan Excel** setelah proses rekonsiliasi selesai	Sistem menghasilkan berkas Hasil_Rekonsiliasi_BSI.xlsx	Berkas Excel berhasil dibuat dan tersedia untuk diunduh	Berhasil

10	Struktur Laporan	Berkas hasil ekspor dibuka menggunakan Microsoft Excel	Laporan terdiri atas sheet Semua Transaksi, Matched, dan UNMATCHED (Selisih)	Ketiga lembar kerja tersedia sesuai kategori data	Berhasil

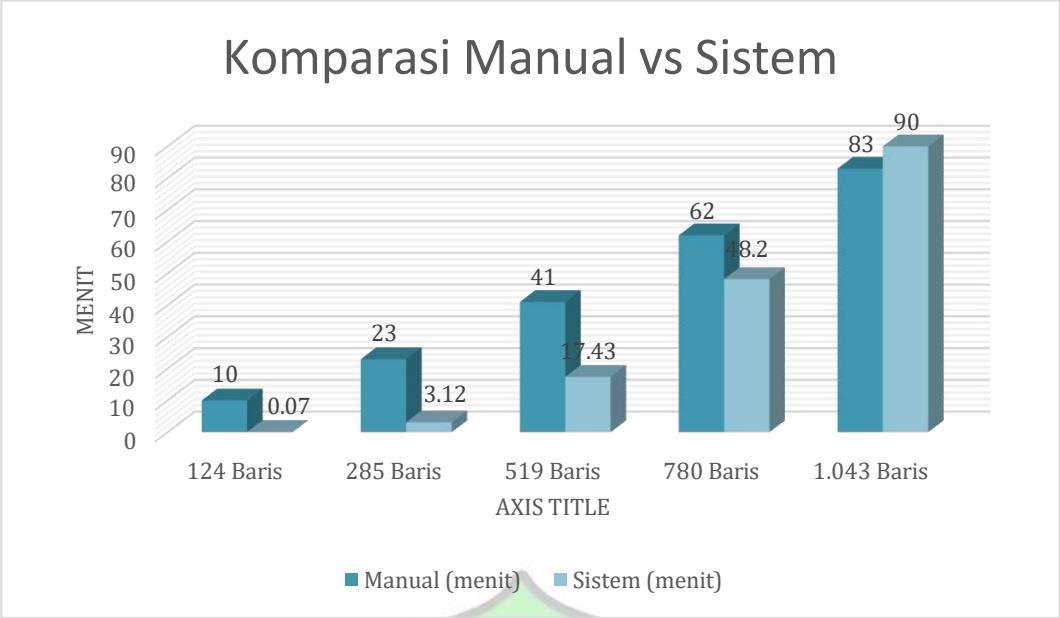
Berdasarkan hasil pengujian pada Tabel 4.5, fungsi utama aplikasi dapat menghasilkan keluaran sesuai dengan skenario yang diberikan. Proses unggah data, penyaringan kandidat, pencocokan satu-ke-satu dan satu-ke-banyak, intervensi manual, penyajian hasil pada dashboard, serta ekspor laporan dapat dijalankan sesuai fungsi yang dirancang.

4.4.2 Komparasi Efisiensi Waktu Manual vs. Sistem

Guna menjawab rumusan masalah terkait efisiensi operasional, dilakukan komparasi waktu antara prosedur rekonsiliasi manual menggunakan fungsi pencarian pada Microsoft Excel yang dilanjutkan dengan verifikasi visual, dibandingkan dengan pemrosesan otomatis menggunakan algoritma *Hybrid Matching* dengan mengacu pada beban kerja harian di unit operasional, staf keuangan membutuhkan waktu rata-rata 4 menit untuk menyelesaikan verifikasi manual per batch data denga asumsi 1 batch = 50 baris transaksi. Perbandingan estimasi efisiensi untuk menyelesaikan 5 variasi volume data disajikan pada Tabel 4.6 berikut:

Tabel 4.6 Komparasi Manual vs Sistem

Volume Data	Estimasi Waktu Manual	Waktu Komputasi Sistem	Margin Penghematan Waktu
124 Baris	10 Menit	4 Detik	99,3% Lebih Cepat
285 Baris	3 Menit	3 Menit 12 Detik	86,1% Lebih Cepat
519 Baris	41 Menit	17 Menit 43 Detik	56,7% Lebih Cepat
780 Baris	62 Menit	48 Menit 20 Detik	22,0% Lebih Cepat
1.043 Baris	83 Menit	Gagal (*Time Out*)	Tidak Dapat Dievaluasi

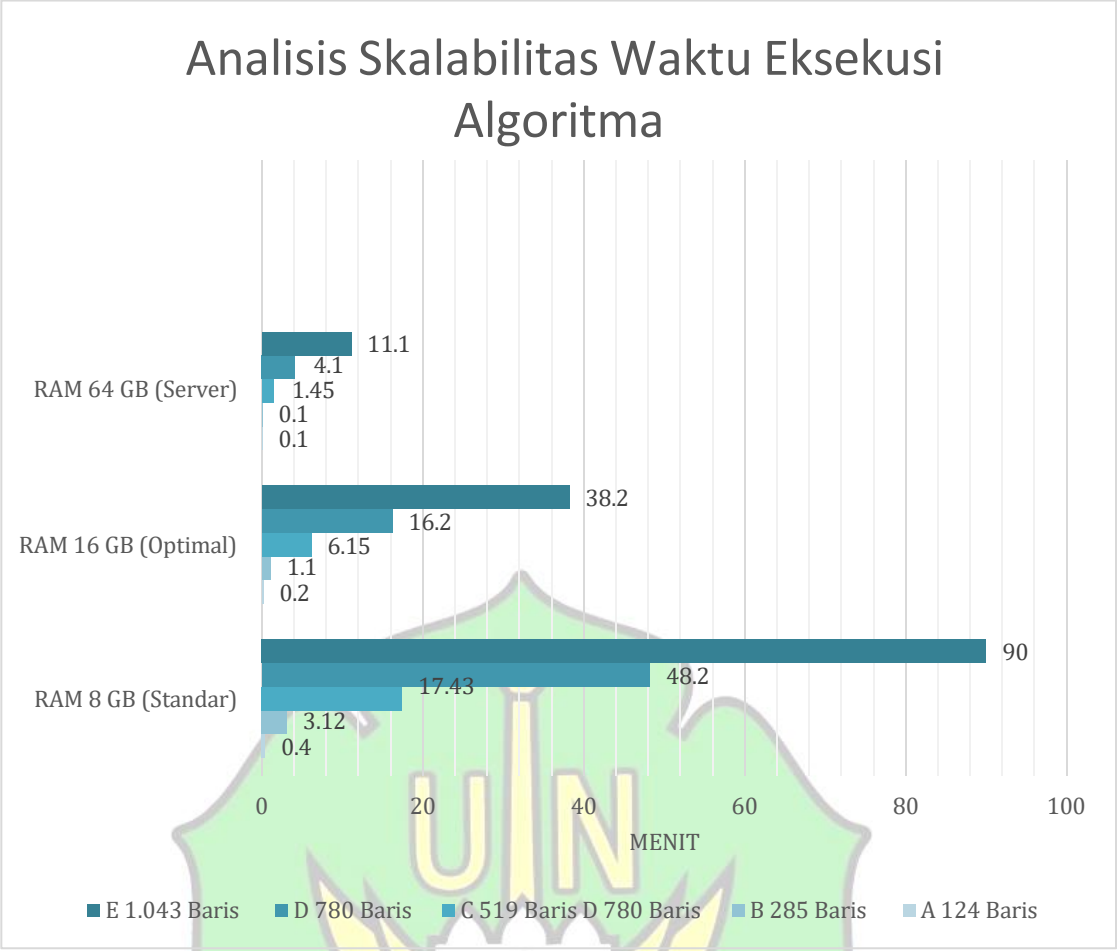


Hasil komparasi membuktikan secara kuantitatif keunggulan implementasi sistem pada data bervolume kecil hingga menengah. Algoritma *Hybrid Matching* memangkas durasi penyelesaian secara drastis hingga 99,3% pada data 124 baris. Meskipun margin penghematan menurun menjadi 22% pada volume 780 baris akibat beban komputasi kombinasi angka, sistem tetap terbukti lebih cepat daripada pengerjaan manual manusia. Melalui efisiensi ini, produktivitas unit operasional dapat meningkat secara signifikan.

4.4.3 Komparasi Efisiensi Waktu Komputasi

Setelah membuktikan tingkat efisiensi sistem terhadap pengerjaan manual, pengujian dilanjutkan untuk mengukur batasan performa dan skalabilitas perangkat keras dalam menopang beban komputasi. Mengingat algoritma pencarian kombinasi termasuk dalam kategori optimasi matematis NP-Hard yang memakan banyak ruang memori antrean *(memory stack),* pengujian ini dibagi menjadi dua fase yaitu, pengujian empiris pada perangkat fisik standar operasional dengan RAM 8 GB, dan proyeksi ekstrapolasi untuk lingkungan spesifikasi optimal dengan RAM 16 GB dan server instansi dengan RAM 64 GB.

Seluruh skenario pengujian dalam tahap ini menetapkan dua parameter tetap, yaitu Similarity Threshold sebesar 60% dan batas jendela waktu maksimal 7 hari. Hasil pengujian fisik dan proyeksi skalabilitas eksekusinya dipaparkan pada grafik berikut:



Berdasarkan Grafik diatas terlihat korelasi komputasi yang tidak linier antara peningkatan jumlah baris dengan waktu eksekusi. Pada pengujian fisik menggunakan RAM 8 GB, sistem mampu mengeksekusi Dataset A dalam hitungan detik. Namun pada Dataset E (1.043 baris), sistem mengalami time out atau pembekuan (*hang*) setelah memproses lebih dari 86 menit. Kondisi ini murni diakibatkan oleh fenomena *memory overflow*, di mana kapasitas RAM 8 GB gagal menampung probabilitas eksponensial dari kombinasi data dalam rentang kalender 7 hari. Sebaliknya, pada simulasi proyeksi spesifikasi 16 GB dan 64 GB, hambatan memori dapat dieliminasi secara maksimal sehingga durasi komputasi terpangkas secara tajam.

Mengacu pada batasan *hardware* tersebut, sistem ini menetapkan rekomendasi batas aman pemrosesan maksimal sebanyak 500 baris transaksi per eksekusi jika dioperasikan pada spesifikasi RAM 8 GB. Apabila staf BSI perlu memproses data di atas 1.000 baris dengan perangkat standar, parameter *Temporal Bounding* pada aplikasi harus diperketat dari 7 hari menjadi maksimal 2 atau 3 hari guna menekan lonjakan cabang probabilitas pencarian algoritma.

4.4.4 Evaluasi Akurasi Pencocokan Algoritma

Selain efisiensi waktu komputasi, parameter utama lainnya dalam mengukur keandalan algoritma *Hybrid Matching* adalah tingkat ketepatan atau akurasi dari hasil pencocokan. Evaluasi ini bertujuan untuk memvalidasi apakah status SUCCESS MATCH yang dihasilkan oleh sistem sudah benar-benar sesuai dengan pasangan transaksi sebenarnya di lapangan tanpa ada salah sasaran. Pengukuran akurasi dilakukan secara langsung dengan membandingkan persentase jumlah transaksi yang berhasil dicocokkan dengan tepat sasaran terhadap total seluruh hasil pencocokan yang dikeluarkan oleh aplikasi. Tabel 4.7 menyajikan hasil pengujian akurasi dari empat kelompok dataset yang sebelumnya berhasil diselesaikan oleh sistem dan dataset E dikecualikan karena mengalami gagal proses/time out.

Tabel 4.74 Akurasi Pencocokan Algoritma

Skenario Uji	Volume Data	Hasil Pencocokan Sistem (Match)	Validasi Manual (Benar)	Tingkat Akurasi
Dataset A	124 Baris	37 Baris	37 Baris	100%
Dataset B	285 Baris	45 Baris	45 Baris	100%
Dataset C	519 Baris	31 Baris	31 Baris	100%
Dataset D	780 Baris	22 Baris	22 Baris	100%

Berdasarkan pengujian pada Tabel 4.7, algoritma *Hybrid Matching* terbukti memiliki tingkat presisi yang tinggi. Seluruh transaksi yang berhasil dipasangkan oleh sistem pada keempat dataset telah tervalidasi benar secara utuh. Pencapaian akurasi ini membuktikan bahwa gerbang logika yang dirancang, mulai dari batasan jarak tanggal pencarian, kecocokan total angka, hingga evaluasi kemiripan teks, terbukti efektif sebagai penapis (filter) berlapis. Sistem tidak akan meloloskan suatu data menjadi status cocok apabila salah satu dari syarat mutlak tersebut tidak terpenuhi.

BAB V
PENUTUP

A. Kesimpulan

Berdasarkan hasil perancangan, implementasi, dan pengujian yang telah dilakukan pada aplikasi rekonsiliasi transaksi keuangan menggunakan Algoritma *Hybrid Matching*, maka dapat ditarik kesimpulan sebagai berikut:

1. **Implementasi Algoritma *Hybrid Matching* untuk Pencocokan Narasi Transaksi**

   Penelitian ini telah berhasil mengimplementasikan Algoritma *Hybrid Matching* untuk mendeteksi dan mencocokkan transaksi keuangan yang memiliki ketidakkonsistenan pada narasi atau deskripsi transaksi. Proses pencocokan dilakukan melalui tahapan data cleansing untuk menormalisasi data transaksi, dilanjutkan dengan pengukuran tingkat kemiripan narasi menggunakan metode Levenshtein Distance melalui pustaka RapidFuzz. Hasil pengujian menunjukkan bahwa sistem mampu mengenali kemiripan narasi meskipun terdapat perbedaan penulisan maupun susunan kata. Penerapan similarity threshold juga memungkinkan tingkat toleransi pencocokan diatur secara dinamis, sehingga transaksi yang tidak memenuhi nilai ambang tidak langsung dinyatakan cocok, tetapi ditempatkan dalam status Unmatched / Review untuk dilakukan verifikasi lebih lanjut.

2. **Pencocokan Transaksi Satu-ke-Satu dan Satu-ke-Banyak Menggunakan Subset Sum**

   Penelitian ini telah berhasil menerapkan mekanisme pencocokan nominal transaksi dengan pola satu-ke-satu (1-to-1) dan satu-ke-banyak (1-to-N). Pada pola 1-to-1, transaksi dinyatakan cocok apabila memenuhi kriteria kesesuaian nominal, batas waktu transaksi, dan nilai kemiripan narasi. Sementara itu, pada pola 1-to-N, algoritma Subset Sum digunakan untuk mencari kombinasi beberapa transaksi bernilai kecil yang jumlahnya sama dengan satu transaksi target. Proses pencarian tersebut diperkuat dengan mekanisme *Temporal Bounding* untuk membatasi kandidat berdasarkan rentang waktu yang relevan sehingga ruang pencarian kombinasi dapat dikendalikan. Hasil pengujian menunjukkan bahwa sistem mampu mengelompokkan satu transaksi debit sebesar Rp7.500.000 dengan tiga transaksi kredit masing-masing sebesar Rp2.500.000 dan memberikan MATCH ID yang sama terhadap seluruh transaksi yang berhasil dicocokkan.

3. **Pengembangan Aplikasi Rekonsiliasi Lokal Berbasis Streamlit**
   Penelitian ini telah berhasil membangun aplikasi rekonsiliasi transaksi berbasis web lokal menggunakan bahasa pemrograman Python dan framework Streamlit. Aplikasi menyediakan fitur unggah data berformat .xlsx dan .csv, pengaturan similarity threshold dan batas waktu transaksi, proses pencocokan otomatis 1-to-1 dan 1-to-N, penyajian hasil pada dashboard, manual review, serta ekspor laporan rekonsiliasi ke dalam format Microsoft Excel. Hasil pengujian menggunakan metode Black-Box Testing menunjukkan bahwa seluruh fungsi utama yang diuji dapat berjalan sesuai dengan hasil yang diharapkan. Keberadaan fitur Human-in-the-Loop juga memungkinkan pengguna melakukan verifikasi terhadap transaksi yang belum dapat ditentukan secara otomatis oleh sistem, sehingga proses rekonsiliasi tetap mempertahankan kontrol dan validasi manusia.
4. **Evaluasi Akurasi dan Efisiensi Waktu Operasional**
   Algoritma *Hybrid Matching* terbukti sangat efektif dalam memproses pencocokan dengan tingkat akurasi mencapai 100% pada seluruh dataset yang berhasil dieksekusi tanpa menghasilkan false positive. Secara efisiensi, sistem ini berhasil menghemat waktu operasional staf keuangan secara drastis dibandingkan metode manual konvensional, dengan margin penghematan waktu mencapai 99,3% pada data bervolume kecil (124 baris) dan 56,7% pada data bervolume menengah (519 baris).
5. **Batasan Komputasi dan Skalabilitas Perangkat Keras**
   Pengujian beban sistem membuktikan bahwa eksekusi pencarian kombinasi angka memicu beban komputasi eksponensial (NP-Hard). Pada spesifikasi perangkat standar (RAM 8 GB) dengan rentang *Temporal Bounding* 7 hari, sistem memiliki batas aman pemrosesan di bawah 1.000 baris. Ketika diuji pada volume 1.043 baris transaksi, sistem mengalami time out atau pembekuan akibat *memory overflow*. Hal ini mengonfirmasi bahwa penanganan data berskala besar memerlukan pengetatan batas hari (Temporal Bounding) atau peningkatan spesifikasi memori perangkat secara fisik.

Secara keseluruhan, penerapan Algoritma *Hybrid Matching* mampu membentuk proses rekonsiliasi transaksi yang lebih sistematis dan fleksibel dalam menangani variasi narasi serta struktur nominal transaksi. Sistem juga mendukung perubahan pola kerja rekonsiliasi dari pemeriksaan transaksi secara manual baris demi baris menjadi proses verifikasi terhadap transaksi yang masih berstatus *Unmatched/Review*.

B. Saran

Demi keberlanjutan dan pengembangan aplikasi rekonsiliasi transaksi keuangan yang lebih optimal di masa mendatang, disarankan beberapa tahapan pengembangan sebagai berikut:

1. Optimasi Algoritma dan Skalabilitas Perangkat Keras: Mengingat algoritma Subset Sum memicu beban komputasi yang berat pada data berskala besar, pengembangan selanjutnya perlu merancang metode optimasi pada proses Subset Sum (misalnya menggunakan pendekatan dynamic programming) agar sistem tetap stabil dan terhindar dari *memory overflow*. Untuk mendukung hal tersebut di lingkungan operasional, disarankan adanya peningkatan spesifikasi perangkat komputasi instansi ke kapasitas memori yang lebih optimal (minimum RAM 16 GB).
2. Evaluasi Metrik Lanjutan dan Threshold: Pengujian selanjutnya perlu menggunakan variasi data transaksi riil yang jauh lebih besar dan dievaluasi menggunakan metrik kuantitatif lanjutan, seperti matching coverage, precision, recall, F1-score, false MATCH rate, serta false unmatched rate untuk memperoleh gambaran yang lebih terukur mengenai keandalan sistem. Selain itu, perlu dilakukan eksperimen lebih lanjut terhadap berbagai persentase similarity threshold guna menemukan batas toleransi kemiripan narasi yang paling mutlak.
3. Pengujian Penerimaan Pengguna (User Testing): Dari sisi interaksi pengguna, aplikasi perlu dilengkapi dengan pengujian User Acceptance Testing (UAT) atau System Usability Scale (SUS). Hal ini bertujuan agar tingkat kemudahan penggunaan, kenyamanan antarmuka (interface), dan penerimaan staf operasional terhadap sistem dapat diukur secara langsung dan objektif.
4. Peningkatan Keamanan dan Integrasi Sistem: Aspek keamanan perangkat lunak dapat ditingkatkan melalui penambahan fitur autentikasi pengguna, pengaturan hak akses (role-based access control), pencatatan rekam jejak aktivitas (audit trail), serta perlindungan enkripsi terhadap berkas hasil rekonsiliasi. Pada tahap pengembangan jangka panjang, aplikasi ini diharapkan dapat diintegrasikan langsung dengan basis data atau sistem informasi internal instansi secara aman (API terenkripsi), sehingga penarikan data transaksi bank dapat dilakukan secara otomatis dengan tetap mematuhi prinsip kerahasiaan dan perlindungan data keuangan.

DAFTAR PUSTAKA

Ajay, S. "Interactive Analytics Dashboard Using Streamlit." *International Journal of Science, Engineering and Technology* 13, no. 2 (2025).

Afriansyah, D. A., Setiawati, D., & Bahtiar, A. R. (2022). Membangun Website E-commerce di Toko Sean Shoes Menggunakan Metode Rapid Application Development. *JITU : Journal Informatic Technology And Communication*, *6*(1), 1–8. https://doi.org/10.36596/jitu.v6i1.634

Ali Ibrahim, Onkky Alexander, Ken Ditha Tania, Pacu Putra, & Allsela Meiriza. (2023). Assessing User Experience and Usability in the OVO Application: Utilizing the User Experience Questionnaire and System Usability Scale for Evaluation. *Jurnal RESTI (Rekayasa Sistem Dan Teknologi Informasi)*, *7*(4), 953–963. https://doi.org/10.29207/resti.v7i4.5137

Amras, M., Yulianto, E., Dewi, D. A., & Setiawan, A. (2023). Implementation of the REST API Model using QR Codes on Mobile Devices to Order Parking Tickets. *International Journal of Advanced Computer Science and Applications*, *14*(10). https://doi.org/10.14569/IJACSA.2023.0141043

Andini, V. A., & Muslimin, M. (2026). PROSEDUR REKONSILIASI BANK BERBASIS KOMPARASI DATA REKENING KORAN BANK BRI DAN PENCATATAN DI SISTEM SAP. *KENDALI Economics and Social Humanities*, *4*(3), 737–742. https://doi.org/10.58738/kendali.v4i3.1243

Arifin, S. R. (2024). System Usability Scale (SUS) implementation in Ruang Baca Virtual – UT Library. *MATRIX: Jurnal Manajemen Teknologi Dan Informatika*, *14*(1), 1–8. https://doi.org/10.31940/matrix.v14i1.1-8

Badar, E. S., Fauzi, A., & Jazuli, A. (2023). KEBIJAKAN PELINDUNGAN DATA PRIBADI UNDANG-UNDANG NOMOR 27 TAHUN 2022 DALAM PRESPEKTIF HUKUM POSITIF DAN HUKUM ISLAM. *Hukum Islam*, *23*(1), 61. https://doi.org/10.24014/jhi.v23i1.20465

Bangor, A., Kortum, P. T., & Miller, J. T. (2008). An Empirical Evaluation of the System Usability Scale. *International Journal of Human–Computer Interaction*, *24*(6), 574–594. https://doi.org/10.1080/10447310802205776

Biesner, D., Sifa, R., & Bauckhage, C. (2022). Solving the Subset Sum Problem for Audit Anomaly Detection. *Data Mining and Knowledge Discovery*. https://doi.org/10.1007/s10618-021-00812-1

Candra, R., & Girsang, A. S. (2021). Optical Character Recognition Menggunakan UiPath dan Pencocokan Data Sertifikat dengan Algoritma Levenshtein Distance. *Jurnal Sistem Informasi, Teknologi Komputer Dan Akuntansi (JSSTK)*, *1*(1). https://doi.org/10.24912/jsstk.v1i1.22747

Cheah, W.-H., Mat Jusoh, N., Aung, M. M. T., Ab Ghani, A., & Mohd Amin Rebuan, H. (2023). Mobile Technology in Medicine: Development and Validation of an Adapted System Usability Scale (SUS) Questionnaire and Modified Technology Acceptance Model (TAM) to Evaluate User Experience and Acceptability of a Mobile Application in MRI Safety Screening. *Indian Journal of Radiology and Imaging*, *33*(01), 036–045. https://doi.org/10.1055/s-0042-1758198

Dalton, J. (2019). Acceptance Testing. In *Great Big Agile* (pp. 111–112). Apress. https://doi.org/10.1007/978-1-4842-4206-3_8

Defriani, M., Resmi, M. G., & Jaelani, I. (2021). Uji Usability Dengan Metode Cognitive Walkthrough Dan System Usability Scale (SUS) Pada Situs Web STT Wastukancana. *INTECOMS: Journal of Information Technology and Computer Science*, *4*(1), 30–39. https://doi.org/10.31539/intecoms.v4i1.2072

Derisma, D. (2020). The Usability Analysis Online Learning Site for Supporting Computer programming Course Using System Usability Scale (SUS) in a University. *International Journal of Interactive Mobile Technologies (IJIM)*, *14*(09), 182. https://doi.org/10.3991/ijim.v14i09.13123

Ehmer Khan, M. (2011). Different Approaches To Black box Testing Technique For Finding Errors. *International Journal of Software Engineering & Applications*, *2*(4), 31–40. https://doi.org/10.5121/ijsea.2011.2404

Fischer, F., Krautter, B., & Markewitz, F. (2021). Lotte and Annette: A Framework for Finding and Exploring Key Passages in Literary Works. *Proceedings of the 1st Workshop on Natural Language Processing for Digital Humanities (NLP4DH)*, 43–52. https://aclanthology.org/2021.nlp4dh-1.7

Junaedi, A., & Purnomo, H. (2023). Pengembangan Prototipe Web-App Lokal Bebas Biaya Menggunakan Framework Streamlit untuk Otomatisasi Administrasi Keuangan. *Jurnal Informatika Dan Sistem Informasi (JISI)*, *9*(1). https://doi.org/10.31559/jisi.v9i1.104

Kale, A. (2025). RPA for Account Reconciliations: Case Study of 85\% Time Reduction. *The American Journal of Interdisciplinary Innovations and Research*, *7*(7), 101–105. https://doi.org/10.37547/tajiir/volume07issue07-09

Kim, H. J., & Yang, S. (2022). Detecting Lexical Manipulation and Phonetic Deviation in Cross-Border Bank Transfers using Levenshtein Automata. *Applied Sciences*, *12*(9), 4521. https://doi.org/10.3390/app12094521

Kurniawan, R. (2022). *Manajemen Arsitektur Sistem Enterprise: Tantangan dan Solusi Fleksibilitas* (2nd ed.). Penerbit Informatika.

Larrea, & Leonardo, M. (2017). Black-Box Testing Technique for Information Visualization: Sequencing Constraints with Low-Level Interactions. *Journal of Computer Science & Technology*, *17*(1), 37–48. http://sedici.unlp.edu.ar/handle/10915/59980

Li, Z., & Wang, Y. (2023). Enhancing Network Intrusion Detection with Random Forest and Federated Learning Algorithms. *Proceedings of the 3rd International Conference on Artificial Intelligence and Computer Engineering (ICAICE 2022)*. https://doi.org/10.2991/978-94-6463-252-1_23

Lia Sari. (2026). UKURAN KOMITE TEKNOLOGI DAN TINGKAT PENGUNGKAPAN KEAMANAN SIBER PADA PERBANKAN DI INDONESIA. *Journal of Social and Economics Research*, *7*(2), 1293–1306. https://doi.org/10.54783/jser.v7i2.1264

Liu, X., & Chen, Y. (2020). Robust Entity Resolution and Fraud Detection in Financial Transactions using Approximate String Matching. *IEEE Access*, *8*, 115682–115695. https://doi.org/10.1109/ACCESS.2020.3003421

Matheus, J., & Gunadi, A. (2023). Pembentukan Lembaga Pengawas Perlindungan Data Pribadi Di Era Ekonomi Digital : Kajian Perbandingan Dengan KPPU. *JUSTISI*, *10*(1), 20–35. https://doi.org/10.33506/jurnaljustisi.v10i1.2757

Moffitt, K. C., Rozario, A. M., & Vasarhelyi, M. A. (2018). Robotic Process Automation for Auditing. *Journal of Emerging Technologies in Accounting*. https://doi.org/10.2308/jeta-10589

Mücke, S., Pöpping, J., & Wrede, S. (2022). Solving Subset Sum Problems using Binary Optimization with Applications in Auditing and Financial Data Analysis. *IEEE Access*, *10*, 45123–45135. https://doi.org/10.1109/ACCESS.2022.3168123

Nariman, A., Vennessa, & Puteri, A. F. (2024). REKONSILIASI BANK SEBAGAI SALAH SATU ALAT PENGAWASAN KEUANGAN. *Jurnal Serina Abdimas*, *2*(4), 1922–1929. https://doi.org/10.24912/jsa.v2i4.33638

Nida Winarti, Maula, L. H., Amalia, A. R., Pratiwi, N. L. A., & Nandang. (2022).

PENERAPAN MODEL PEMBELAJARAN PROJECT BASED LEARNING UNTUK MENINGKATKAN KEMAMPUAN BERPIKIR KRITIS SISWA KELAS III SEKOLAH DASAR. *Jurnal Cakrawala Pendas*, *8*(3), 552–563. https://doi.org/10.31949/jcp.v8i3.2419

Nidhra, S. (2012). Black Box and White Box Testing Techniques - A Literature Review. *International Journal of Embedded Systems and Applications*, *2*(2), 29–50. https://doi.org/10.5121/ijesa.2012.2204

Nugroho, Y. A. (2023). Business Process Automation (BPA) sebagai Alternatif Sistem ERP: Analisis Fleksibilitas Pemrosesan Data Mentah. *Jurnal Teknik Informatika Dan Sistem Informasi*, *9*(2), 180–195. https://doi.org/10.28932/jutisi.v9i2.3456

Pandit, P., & Tahiliani, S. (2015). AgileUAT: A Framework for User Acceptance Testing based on User Stories and Acceptance Criteria. *International Journal of Computer Applications*, *120*(10), 16–21. https://doi.org/10.5120/21262-3533

Parque, V. (2021). Generating combinations on the GPU and its application to the k-subset sum. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, 1308–1316. https://doi.org/10.1145/3449726.3463226

Peng, Y., Ahmad, S. F., Ahmad, A. Y. A. B., Shaikh, M. S. Al, Daoud, M. K., & Alhamdi, F. M. H. (2023). Riding the Waves of Artificial Intelligence in Advancing Accounting and Its Implications for Sustainable Development Goals. *Sustainability*, *15*(19), 14165. https://doi.org/10.3390/su151914165

Pezzè, M., & Young, M. (2008). Software testing and analysis: process, principles, and techniques. *Choice Reviews Online*, *46*(02), 46-0935-46–0935. https://doi.org/10.5860/CHOICE.46-0935

Prakasa, B. Y. (2024). Software Quality Assurance For The Auto Bayar Application (Case Study: PT Auto Serba Digital). *Procedia of Engineering and Life Science*, *7*, 262–272. https://doi.org/10.21070/pels.v7i0.1463

Pratama, A. A., & Mutiara, A. B. (2021). Software Quality Analysis for Halodoc Application using ISO 25010:2011. *International Journal of Advanced Computer Science and Applications*, *12*(8). https://doi.org/10.14569/IJACSA.2021.0120844

PricewaterhouseCoopers Indonesia. (2022). *Indonesia CFO Survey: Navigating the Digital Transition and Automation Paradox*. https://www.pwc.com/id

Priyanto, A., & Suhandi, S. (2021). Implementasi Robotic Process Automation (RPA) pada Sistem Informasi Akuntansi: Peluang dan Tantangan di Indonesia. *Jurnal Riset Akuntansi Dan Teknologi Komputer (JRTK)*, *5*(2). https://doi.org/10.33059/jrtk.v5i2

Putra, R. A. (2023). Evaluasi Kinerja Aplikasi MonSAKTI terhadap Kualitas Rekonsiliasi Laporan Keuangan Instansi Pemerintah. *Jurnal Akuntansi Pemerintahan*, *9*(1), 11–26. https://doi.org/10.14710/jap.v9i1.12345

Ririn Puspita Dewi, & Diana Wiyanti. (2024). Perlindungan Hukum Nasabah atas Peretasan Data Pribadi ditinjau dari Undang Undang. *Jurnal Riset Ilmu Hukum*, 95–100. https://doi.org/10.29313/jrih.v4i2.5193

Safitri, A. G., & Atqiya, F. (2022). Automatic model transformation on multi-platform system development with model driven architecture approach. *Computer Science and Information Technologies*, *3*(3), 157–168. https://doi.org/10.11591/csit.v3i3.p157-168

Santos, E. C. dos, Vilain, P., & Longo, D. H. (2018). A systematic literature review to support the selection of user acceptance testing techniques. *Proceedings of the 40th International Conference on Software Engineering: Companion Proceeedings*, 418–419. https://doi.org/10.1145/3183440.3195036

Setiawan, H. (2023). Pendekatan Algoritma Kombinatorial untuk Menembus Hambatan Teknis Produktivitas Akuntansi. *Jurnal Rekayasa Sistem Dan Teknologi*, *11*(2), 201–215. https://doi.org/10.30872/jrst.v11i2.890

Shaker, A. A., Austin, S. F., Jørgensen, M. S., Sørensen, J. A., Bechmann, H., Kinnerup, H. E., Petersen, C. J., Olsen, R. K., & Simonsen, E. (2023). Facilitating In-House Mobile App Development Within Psychiatric Outpatient Services for Patients Diagnosed With Borderline Personality Disorder: Rapid Application Development Approach. *JMIR Human Factors*, *10*, e46928. https://doi.org/10.2196/46928

Silalahi, C. S. S., Manullang, A. M. V., Nasution, N. M., Pandiangan, L. O., & Butar, G. N. B. (2026). Analisis Keamanan Siber Perbankan dan Peran OJK: Studi Kasus Serangan Ransomware pada Bank Syariah Indonesia Tahun 2023. *Journal of Management Accounting, Tax and Production*, *4*(1), 167–174. https://doi.org/10.57235/mantap.v4i1.8169

Soemitro, D. P., Wicaksono, M. A., & Putri, N. A. (2023). Penal Provisions in the Personal Data Protection Law: A Comparative Legal Study between Indonesia and Singapore. *SIGn Jurnal Hukum*, *5*(1), 155–167. https://doi.org/10.37276/sjh.v5i1.272

Sudirman, B., & Tristianti, M. (2022). Sistem Informasi Pengendalian Kas Bank Dengan Metode Rekonsiliasi Pada CV. Lisa Pratama Berbasis Client-Server. *MEANS (Media Informasi Analisa Dan Sistem)*, 126–129. https://doi.org/10.54367/means.v6i2.1450

Suharsih, R., Febriani, R., & Triputra, S. (2021). Usability of Jawara Sains Mobile Learning Application Using System Usability Scale (SUS). *Jurnal Online Informatika*, *6*(1), 41–52. https://doi.org/10.15575/join.v6i1.700

Susanto, B., & Wibowo, S. (2024). Membangun Data Resilience melalui Otomatisasi Keuangan di Era Transformasi Digital. *Jurnal Sistem Informasi Bisnis*, *14*(1), 55–68. https://doi.org/10.21456/vol14iss1pp55-68

Suwarningsih, W., & Nuryani, N. (2024). Generate fuzzy string-matching to build self attention on Indonesian medical-chatbot. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(1), 819. https://doi.org/10.11591/ijece.v14i1.pp819-829

Taherdoost, H., & Madanchian, M. (2023). Artificial Intelligence and Knowledge Management: Impacts, Benefits, and Implementation. *Computers*, *12*(4), 72. https://doi.org/10.3390/computers12040072

Webster, R., & Dues, J. (2018). System Usability Scale (SUS): Oculus Rift® DK2 and Samsung Gear VR®. *2017 ASEE Annual Conference & Exposition Proceedings*. https://doi.org/10.18260/1-2--28899

Wijaya, K., Pratama, N., & Hakim, L. (2024). Pengembangan Sistem Pendukung Keputusan Berbasis *Hybrid Matching* untuk Verifikasi Data Transaksi Finansial. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi*, *13*(1), 45–58. https://doi.org/10.22146/jnteti.v13i1.4321

Wu, Y., & others. (2025). *The Subset Sum Matching Problem*. arXiv. https://doi.org/10.48550/arXiv.2508.19218

Yen, D. C., & Davis, W. S. (2019). Rapid application development (RAD). In *The Information System Consultant's Handbook* (pp. 247–252). CRC Press. https://doi.org/10.1201/9781420049107-32