

## **IMPLEMENTATION OF LOW-CODE PROGRAMMING TECHNOLOGY WITH AGILE METHOD IN DEVELOPING A PETTY CASH TRANSACTION MANAGEMENT APPLICATION (CASE STUDY: PT BANK CENTRAL ASIA TBK)**

Anin Ambya Soulani<sup>\*1</sup>, Nofiyati<sup>2</sup>, Nur Alfi Ekowati<sup>3</sup>

<sup>1,2,3</sup>Informatics, Faculty of Engineering, Universitas Jenderal Soedirman, Indonesia  
Email: <sup>1</sup>[anin.soulani@mhs.unsoed.ac.id](mailto:anin.soulani@mhs.unsoed.ac.id), <sup>2</sup>[nofiyati@unsoed.ac.id](mailto:nofiyati@unsoed.ac.id), <sup>3</sup>[nuralfi.ekowati@unsoed.ac.id](mailto:nuralfi.ekowati@unsoed.ac.id)

(Article received: June 15, 2024; Revision: June 26, 2024; published: July 02, 2024)

### **Abstract**

*Conventional application development often faces challenges like extensive code writing, long development times, high costs, and difficulties in maintenance and customization. Low-code programming offers an innovative solution by minimizing manual coding and enabling application creation through visual interfaces and drag-and-drop logic. This research explains the application of low-code programming technology in developing a petty cash transaction recording application at PT Bank Central Asia Tbk, specifically in the Corporate Communication & Social Responsibility Division. The low-code approach allows for faster, more efficient, and easier-to-maintain application development. The research uses the agile method, covering plan, design, develop, test, deploy, review, and launch stages. This case study, using the OutSystems platform, shows significant benefits such as increased development time efficiency, ease of maintenance, and flexibility in meeting dynamic business needs. The developed application can be integrated into the company's existing IT environment, improving the accuracy of petty cash transaction recording and reporting, and providing easy user access. In conclusion, low-code programming technology proves to be an effective solution for developing complex business applications efficiently in terms of time and cost.*

**Keywords:** Application, Low-Code Programming, OutSystems, Petty Cash, PT Bank Central Asia Tbk.

## **IMPLEMENTASI TEKNOLOGI LOW-CODE PROGRAMMING DENGAN METODE AGILE DALAM PENGEMBANGAN APLIKASI PENCATATAN TRANSAKSI PETTY CASH (STUDI KASUS: PT BANK CENTRAL ASIA TBK)**

### **Abstrak**

Proses pengembangan aplikasi konvensional sering menghadapi tantangan seperti penulisan kode yang ekstensif, waktu pengembangan yang panjang, biaya tinggi, dan kesulitan pemeliharaan serta penyesuaian dengan kebutuhan bisnis. *Low-code programming* hadir sebagai solusi dengan meminimalkan penulisan kode manual dan memungkinkan pembuatan aplikasi melalui antarmuka visual serta logika *drag-and-drop*. Penelitian ini menjelaskan penerapan *low-code programming* dalam pengembangan aplikasi pencatatan transaksi *petty cash* di PT Bank Central Asia Tbk, khususnya pada Divisi *Corporate Communication & Social Responsibility*. Metode *agile* yang digunakan meliputi tahapan *plan, design, develop, test, deploy, review, dan launch*. Studi kasus menggunakan platform OutSystems menunjukkan bahwa *low-code programming* meningkatkan efisiensi waktu pengembangan, kemudahan pemeliharaan, serta fleksibilitas dalam menyesuaikan kebutuhan bisnis. Aplikasi yang dikembangkan dapat langsung diintegrasikan ke lingkungan IT perusahaan, meningkatkan akurasi pencatatan dan pelaporan transaksi *petty cash*, serta memudahkan akses pengguna. Kesimpulannya, teknologi *low-code programming* efektif dalam pengembangan aplikasi bisnis kompleks dengan efisiensi waktu dan biaya.

**Kata kunci:** Aplikasi, Low-Code Programming, OutSystems, Petty Cash, PT Bank Central Asia Tbk.

### **1. PENDAHULUAN**

Persaingan teknologi pada era modern sekarang semakin ketat sehingga semua ingin berlomba-lomba untuk menemukan dan menggunakan teknologi termutakhir. Hal ini menuntut para individu maupun

perusahaan untuk terus berinovasi dan meningkatkan efisiensi agar dapat bersaing dengan individu maupun perusahaan lain. Dengan adanya inovasi terbaru, perusahaan dapat meningkatkan efektivitas, efisiensi, serta produktivitas dalam mengelola sumber daya, proses, dan informasi. Salah satu penerapan teknologi

informasi yang banyak digunakan oleh perusahaan adalah pengembangan aplikasi.

Aplikasi adalah suatu subkelas *software* dan alat terapan yang memanfaatkan kemampuan komputer secara langsung dan terpadu untuk melakukan tugas tertentu yang diinginkan serta difungsikan secara khusus bagi *user* [1] [2]. Aplikasi dapat membantu perusahaan dalam melakukan berbagai aktivitas, seperti pengolahan data, komunikasi, transaksi, analisis, dan lain-lain. Aplikasi juga dapat memberikan nilai tambah bagi perusahaan, seperti meningkatkan kualitas layanan, memperluas pasar, meningkatkan loyalitas pelanggan, dan lain-lain.

Proses pengembangan aplikasi membutuhkan banyak aspek pendukung, seperti sumber daya manusia yang memiliki keahlian dalam pemrograman, desain, dan analisis. Selain itu, pengembangan aplikasi memerlukan waktu yang panjang, biaya yang besar, dan risiko yang tinggi. Untuk mengatasi tantangan ini, pendekatan *low-code programming* menjadi solusi yang lebih cepat, murah, mudah, dan fleksibel. *Low-code programming* memungkinkan pengembangan aplikasi dengan kebutuhan pengkodean minimal, menggunakan alat, komponen kode khusus, dan skrip *boilerplate* yang memfasilitasi *proses drag-and-drop* untuk lingkungan pengembangan visual [3].

PT Bank Central Asia Tbk (BCA), telah menerapkan *low-code programming* dalam pengembangan aplikasi internalnya, termasuk aplikasi pencatatan transaksi *petty cash* menggunakan platform OutSystems. Aplikasi ini penting untuk mengelola kas kecil dan akuntansi, namun pencatatan manual memiliki beberapa masalah seperti memerlukan waktu dan tenaga yang banyak serta rentan terhadap kesalahan. Dengan *low-code programming*, BCA dapat meningkatkan efisiensi dan efektivitas dalam pencatatan transaksi *petty cash*.

PT Bank Central Asia Tbk (BCA) didirikan pada 10 Oktober 1955 dan merupakan bank swasta terbesar di Indonesia. Dengan aset mencapai Rp1.288 triliun dan ekuitas Rp209 triliun per Desember 2021, serta jaringan yang luas mencakup lebih dari 1.300 cabang, 17.000 ATM, dan 400.000 EDC di seluruh Indonesia, BCA melayani lebih dari 25 juta nasabah dari berbagai segmen dan industri [4].

*Low-code* adalah pendekatan pengembangan perangkat lunak yang memungkinkan pengembang untuk membuat aplikasi menggunakan antarmuka visual dan pengaturan konfigurasi, dengan penulisan kode yang minimal. Hal ini memungkinkan pengembangan aplikasi yang lebih cepat dan dapat diakses oleh orang-orang dengan berbagai tingkat keahlian teknis. Pendekatan ini memungkinkan pengembang untuk memanfaatkan komponen-komponen yang sudah ada dan mempercepat proses pengembangan aplikasi [5] [6]. *Low-code programming* mampu menyederhanakan dan merampingkan pekerjaan pengembang profesional,

memungkinkan mereka untuk melakukannya dan memberikan aplikasi perusahaan dalam waktu singkat dengan kualitas yang tinggi [7].

OutSystems adalah platform *low-code* yang memungkinkan para individu maupun organisasi untuk membuat aplikasi dengan cepat dan mudah hanya dengan menggunakan desain visual dan integrasi. Dengan OutSystems, para pengguna dapat mengembangkan aplikasi untuk berbagai industri dan kebutuhan, seperti portal pelanggan, aplikasi internal, sistem inti, dan lainnya. OutSystems juga memberikan kinerja tinggi, keamanan, dan skalabilitas untuk aplikasi yang sedang dikembangkan oleh pengguna [8].

*Petty cash* adalah sejumlah uang tunai yang disisihkan oleh perusahaan untuk membiayai pengeluaran kecil dan tak terduga yang tidak praktis jika harus dibayar menggunakan cek. Pengeluaran ini meliputi biaya seperti pembelian perangk, membayar tumpangan taksi, dan biaya telegraf atau layanan pos. Pengelolaan *petty cash* biasanya dilakukan oleh petugas yang bertanggung jawab untuk menyimpan, mengawasi, dan mencatat setiap penggunaan dana tersebut. Setiap transaksi dicatat dalam buku *petty cash* dengan disertai kuitansi sebagai bukti pengeluaran [9] [10].

Dalam penelitian berjudul "*Acceleration of Learning Management System Application Development in the Education Sector Using the Low Code Concept on Microgen*" (2023) oleh Ega Wachid Radiegyta, Daniel Hasiholan Tinambunan, Rido Dwi Kurniawan, dan Richardus Eko Indrajit, hasil penelitian menunjukkan bahwa penggunaan konsep *low-code* pada platform Microgen secara signifikan mempercepat pengembangan aplikasi sistem manajemen pembelajaran. Temuan ini mengindikasikan bahwa Microgen memungkinkan pengembangan yang efisien, hemat sumber daya, dan responsif terhadap kebutuhan pasar dan pengguna [11]. Hasil dari penelitian ini juga mendukung manfaat yang diamati dari platform OutSystems yang digunakan dalam penelitian ini. Pengembangan menggunakan pendekatan *low-code* memungkinkan pembuatan aplikasi dengan cepat dan fleksibel, secara efektif memenuhi kebutuhan bisnis spesifik dengan minimal pengkodean.

Berdasarkan uraian di atas terlihat bahwa *low-code programming* merupakan salah satu metode pengembangan aplikasi yang memiliki banyak kelebihan dan potensi. Pengembangan aplikasi yang menggunakan teknologi *low-code* dengan menggunakan platform OutSystems diharapkan dapat membantu bisnis mengatasi tantangan era digital. *Low-code programming* juga dapat memberikan solusi terbaik bagi perusahaan untuk meningkatkan efisiensi, efektivitas, dan produktivitas pengelolaan sumber daya, proses, dan informasi.

## 2. METODE PENELITIAN

Objek penelitian ini adalah penerapan teknologi *low-code programming* menggunakan platform OutSystems untuk mengembangkan aplikasi pencatatan transaksi *petty cash* di PT Bank Central Asia Tbk. Penelitian difokuskan pada proses implementasi *low-code programming* dalam divisi *Corporate Communication & Social Responsibility* pada Biro Direksi dan Komisararis. Melalui subjek ini, penelitian akan mengeksplorasi bagaimana teknologi *low-code programming* dapat dioptimalkan untuk pencatatan transaksi *petty cash* di lingkungan perbankan.

Metodologi *agile* adalah kumpulan metode pengembangan perangkat lunak yang berbasis pada model iteratif dan inkremental. *Agile* memungkinkan pengembangan perangkat lunak yang cepat beradaptasi terhadap perubahan kebutuhan. Metode ini lebih menekankan pada individu dan interaksi mereka daripada proses atau alat yang digunakan [12].

Prinsip utama *agile*, yang dikenal sebagai *Agile Manifesto*, mencakup empat nilai: individu dan interaksi lebih penting daripada proses dan alat, perangkat lunak yang berfungsi lebih penting daripada dokumentasi yang lengkap, kolaborasi dengan klien lebih penting daripada negosiasi kontrak, dan respons terhadap perubahan lebih penting daripada mengikuti rencana [13]. Gambaran dari metode *agile* dapat dilihat pada Gambar 1.



Gambar 1. Metode *agile* [14]

### 2.1. Plan

Pada tahapan ini membuat perencanaan sistem yang akan dikembangkan dengan cara pengumpulan data terhadap *user* berupa wawancara langsung atau kuesioner untuk mendapatkan kebutuhan yang *user* inginkan [15]. Karyawan di Biro Direksi dan Komisararis PT Bank Central Asia Tbk menjadi fokus sebagai *user* aplikasi I-PTC untuk pencatatan transaksi *petty cash*.

*Sprint Planning* kemudian dilakukan untuk menganalisis kebutuhan pengguna yang akan diselesaikan dalam periode tertentu, dengan menetapkan fitur-fitur yang akan diimplementasikan [16]. *Sprint Planning* ini juga melibatkan pembuatan tabel *timeline* yang mencakup detail pekerjaan, progres, PIC, dan durasi pengerjaan untuk setiap fitur. Selanjutnya, *Weekly Meeting Plan* digunakan untuk memastikan efisiensi dan kolaborasi tim dalam mengatasi kendala, meninjau progres proyek secara rutin, dan membahas rencana kerja selama pekan

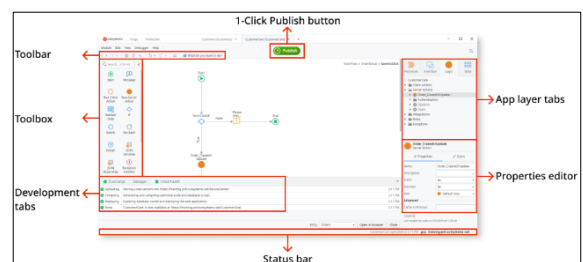
yang akan datang. Agenda pertemuan mingguan mencakup *review action items*, update progres proyek, diskusi pemecahan masalah, pendekatan untuk pekerjaan mendatang, meninjau *timeline*, serta pertanyaan dan diskusi umum untuk saling memberikan masukan.

### 2.2. Design

Dalam tahap desain, dilakukan perancangan aplikasi berdasarkan kebutuhan pengguna sebelumnya. Untuk aplikasi pencatatan transaksi *petty cash*, perlu dirancang *user interface* (UI) sesuai preferensi pengguna. Rancangan UI ini akan dilakukan melalui platform *low-code programming*, seperti OutSystems, dengan sistem *drag-and-drop* elemen visual. Selain itu, akan menggunakan *flowchart*, *Entity Relationship Diagram* (ERD), dan *Data Flow Diagram* (DFD) untuk merancang struktur dan alur kerja aplikasi secara detail.

### 2.3. Develop

Pada tahap pengembangan aplikasi, teknologi *low-code programming* menggunakan platform populer seperti OutSystems akan digunakan. Platform ini memungkinkan pengembangan aplikasi *mobile* atau *website* dengan cepat dan efisien. OutSystems menyediakan jaringan yang terintegrasi yang mencakup seluruh siklus hidup pengembangan: pengembangan, jaminan kualitas, implementasi, pemantauan, dan manajemen [17]. OutSystems memiliki komponen utama, termasuk *Service Studio*, di mana pengguna dapat membuat aplikasi baru dengan memilih menu 'New Application' dan membuat *module* baru untuk mengembangkan aplikasi.



Gambar 2. *Workspace* OutSystems [18]

Pada Gambar 2 terlihat *module workspace* pada OutSystems yang terdiri dari beberapa komponen penting, termasuk *processes* yang menjelaskan aliran pekerjaan dalam aplikasi, *interface* yang menampilkan tampilan aplikasi dan digunakan untuk berinteraksi dengan pengguna, *logic* yang berisi *script code* untuk memproses data, dan *database* yang digunakan untuk menyimpan dan mengelola data aplikasi. Selain itu, terdapat *forge* dalam OutSystems, sebuah repositori *online* yang berisi komponen perangkat lunak yang dapat dibagikan dan digunakan kembali oleh pengembang.

**2.4. Test**

Tahapan selanjutnya merupakan pengujian aplikasi. Dalam penelitian ini aplikasi akan diuji pada sisi pengguna. Pengujian ini berfungsi untuk memastikan aplikasi dapat berjalan sesuai kebutuhan atau tidak. Pada sisi pengguna akan diuji menggunakan metode *User Acceptance Testing* (UAT). UAT adalah proses di mana pengguna akhir sistem memvalidasi bahwa sistem tersebut memenuhi persyaratan mereka dan bekerja seperti yang diharapkan dalam skenario dunia nyata. Ini adalah tahap akhir pengujian sebelum sistem dirilis ke pengguna [19].

**2.5. Deploy**

Pada tahap deploy aplikasi dalam metode *agile*, aplikasi yang telah dikembangkan dirilis atau diimplementasikan ke lingkungan *production* setiap kali fitur baru atau yang direvisi selesai dikembangkan. Hal ini memungkinkan pengembang untuk mendapatkan *feedback* dari pengguna dan menyesuaikan aplikasi jika diperlukan. Dalam OutSystems, proses deployment dilakukan dengan mengklik tombol “1-Click Publish” pada *Service Studio*, sehingga aplikasi dapat diunggah ke OutSystems *Cloud* atau *environment* personal untuk diakses melalui URL *environment* yang telah dimiliki.

**2.6. Review**

Tahap *review* dalam metode *agile* adalah evaluasi berkala yang dilakukan selama siklus pengembangan yaitu setiap satu kali *sprint* diselesaikan untuk memastikan jalannya proyek sesuai dengan rencana awal. Proses ini penting karena memastikan produk berkembang sesuai dengan kebutuhan pengguna.

**2.7. Launch**

Pada tahap peluncuran perangkat lunak yang telah selesai dikembangkan akan diluncurkan ke publik atau diimplementasikan pada lingkungan yang sudah ditentukan. Setelah peluncuran, pengembang akan memantau dan memperbaiki masalah yang muncul pada perangkat lunak yang telah diluncurkan [20].

**3. HASIL DAN PEMBAHASAN**

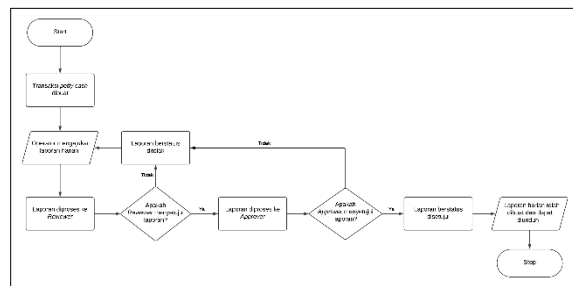
Hasil dari penelitian ini adalah sebuah aplikasi bernama I-PTC yang mana mempunyai fungsi untuk mencatat transaksi *petty cash* di PT Bank Central Asia Tbk pada Biro Direksi dan Komisaris. Aplikasi ini dikembangkan menggunakan teknologi *low-code programming* dengan menggunakan platform OutSystems. Berikut ini merupakan analisis dan pembahasan dari penelitian yang telah dilakukan.

**3.1. Design**

Pada tahap ini, dibuat rancangan desain aplikasi, termasuk *Flowchart*, *Data Flow Diagram* (DFD), dan *Entity Relationship Diagram* (ERD), untuk memberikan gambaran yang jelas mengenai aplikasi yang akan dikembangkan.

**3.1.1. Flowchart**

*Flowchart* adalah representasi visual dari langkah-langkah, urutan, dan keputusan dalam suatu proses dengan menggunakan simbol-simbol tertentu yang dihubungkan oleh garis atau tanda panah [21]. Berikut adalah garis besar *flowchart* pencatatan transaksi *petty cash* pada aplikasi I-PTC yang ditunjukkan pada Gambar 3.

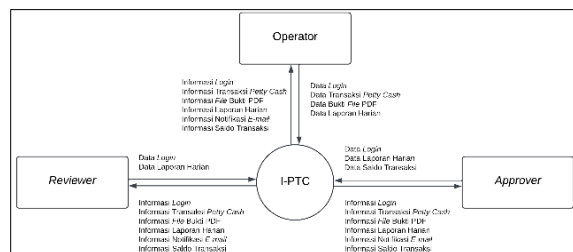


Gambar 3. *Flowchart* aplikasi I-PTC

Ditunjukkan pada Gambar 3 bahwa aplikasi akan mencatat transaksi *petty cash* yang dimuat oleh *user operator*. Setelah transaksi dicatat, maka operator dapat mengajukan laporan harian yang berisi transaksi kepada *user reviewer* dan *approver* untuk disetujui atau ditolak.

**3.1.2. Data Flow Diagram (DFD)**

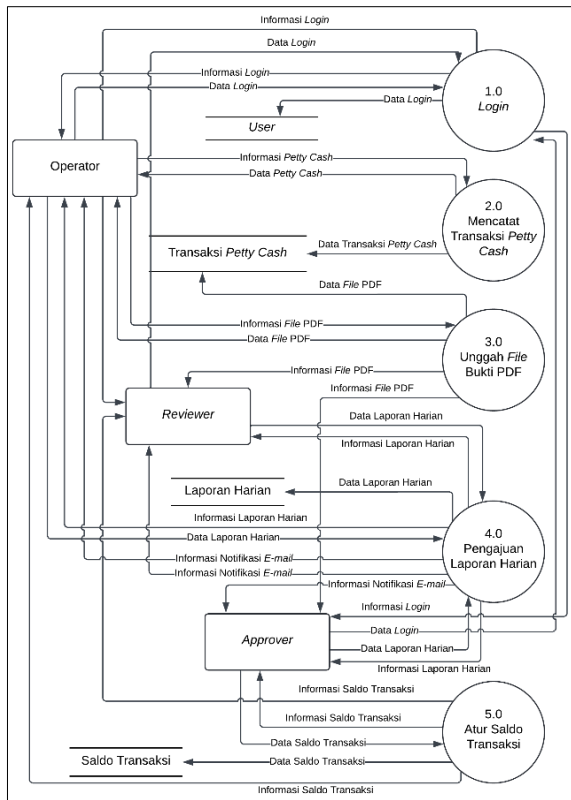
*Data Flow Diagram* (DFD) adalah model logis dari data atau proses yang dibuat untuk menggambarkan asal data dan tujuannya dalam sistem. DFD menunjukkan tempat penyimpanan data, proses yang menghasilkan data tersebut, dan interaksi antara data yang tersimpan dengan proses yang diterapkan pada data tersebut [22]. DFD berupa gambaran visual tentang apa saja yang terlibat dalam program dari awal hingga akhir. Berikut adalah DFD level 0 dari aplikasi I-PTC.



Gambar 4. *Data flow diagram* level 0 aplikasi I-PTC

Pada DFD level 0 dijelaskan hak akses serta informasi apa saja yang pengguna akan dapatkan pada aplikasi I-PTC seperti pada *user operator*,

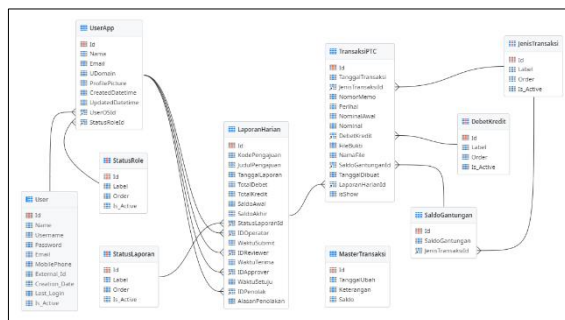
reviewer, dan approver. Lalu terdapat juga DFD level 1 yang menjelaskan proses-proses yang ada pada aplikasi seperti login, dan lain-lain. DFD level 1 ditunjukkan pada Gambar 5.



Gambar 5. Data flow diagram level 1 aplikasi I-PTC

3.1.3. Entity Relationship Diagram (ERD)

Entity Relationship Diagram (ERD) adalah diagram berbentuk notasi grafis yang berada dalam pembuatan database yang menghubungkan antara data satu dengan yang lain [23]. Dalam aplikasi I-PTC, terdapat tabel master utama seperti transaksi petty cash, laporan harian, data pengguna, dan saldo master transaksi. ERD memberikan gambaran tentang hubungan entitas dan atributnya, menjadi landasan dalam membangun struktur basis data aplikasi. Gambaran lengkap dari ERD aplikasi I-PTC dapat dilihat pada Gambar 6 di bawah ini.



Gambar 6. Entity Relationship Diagram aplikasi I-PTC

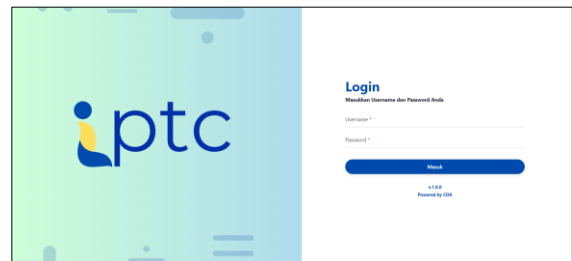
3.2. Develop

Dalam tahap ini, aplikasi pencatatan transaksi petty cash yang dinamai I-PTC dikembangkan menggunakan platform low-code programming OutSystems. Proses pengembangan mencakup langkah-langkah khusus metode agile untuk memenuhi kebutuhan awal pengguna. Dengan low-code programming, pengembangan menjadi lebih cepat dan efisien karena penggunaan komponen visual dan template siap pakai yang mengurangi coding manual.

OutSystems menyediakan antarmuka drag-and-drop, alat otomatisasi, dan logic flow sebagai back-end aplikasi, mempercepat proses pengembangan dan iterasi berdasarkan feedback pengguna. Logic flow OutSystems terdiri dari client action dan server action, masing-masing menjalankan logika dari sisi klien dan sisi server atau back-end aplikasi.

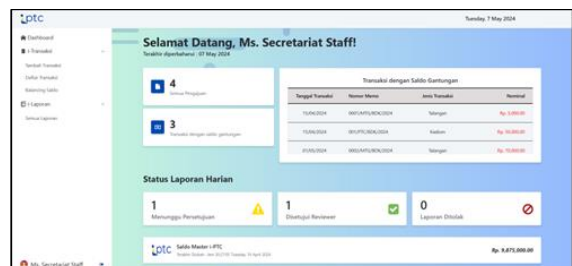
3.2.1. Halaman Login dan Dashboard

Halaman login aplikasi I-PTC merupakan halaman awal yang akan ditampilkan ketika pengguna memasuki aplikasi. Halaman ini penting untuk menyediakan akses aman bagi pengguna, yang harus memasukkan kredensial untuk mengakses fitur aplikasi. Fitur keamanan, seperti enkripsi data, melindungi informasi pengguna. Gambar 7 menampilkan halaman login I-PTC.



Gambar 7. Halaman login

Setelah pengguna berhasil login, maka setiap pengguna akan memasuki halaman dashboard yang mana halaman ini akan memberikan gambaran menyeluruh mengenai data dan aktivitas terkini yang relevan untuk semua pengguna. Setiap dashboard dirancang sesuai dengan peran pengguna dalam sistem, memberikan akses ke informasi dan fitur yang sesuai dengan kebutuhan spesifik mereka. Contoh halaman dashboard untuk user operator ditampilkan pada Gambar 8.

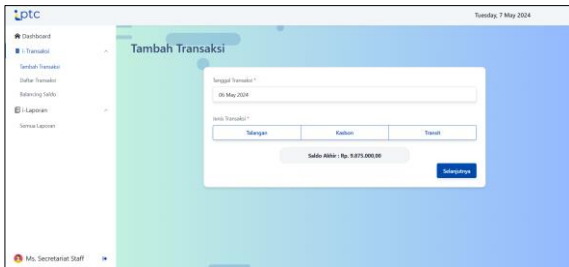


Gambar 8. Halaman dashboard user operator



### 3.2.2. Halaman Tambah Transaksi Petty Cash

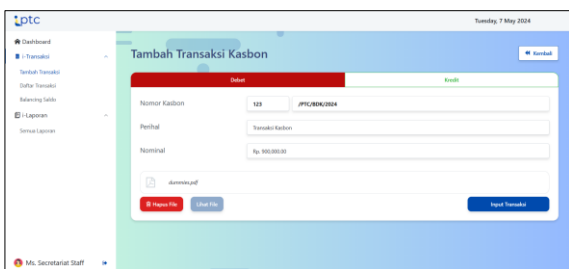
Halaman tambah transaksi di aplikasi I-PTC memungkinkan pengguna mencatat transaksi *petty cash* dengan cepat dan mudah. Pengguna dapat memasukkan detail transaksi, seperti tanggal, jumlah, jenis, dan perihal transaksi. Jenis transaksi *petty cash* terbagi menjadi tiga: talangan, kasbon, dan transit, seperti terlihat pada Gambar 9.



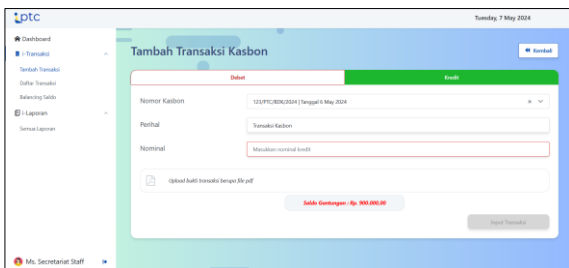
Gambar 9. Halaman tambah transaksi *petty cash*

Ketika tombol "Selanjutnya" ditekan, maka halaman akan beralih sesuai jenis transaksi yang dipilih. Pengguna yang memilih jenis transaksi talangan, kasbon, atau transit akan masuk ke halaman tambah transaksi yang terdiri dari dua jenis: debit dan kredit.

Pada transaksi debit, pengguna dapat memasukkan nomor memo, nominal, jumlah, dan bukti transaksi (opsional), seperti terlihat pada Gambar 10. Pada transaksi kredit, pengguna dapat memilih transaksi *petty cash* debit yang akan dikreditkan. Setelah memilih transaksi, pengguna dapat memasukkan nominal kredit yang diinginkan. Tampilan rinci dapat dilihat pada Gambar 11.



Gambar 10. Halaman tambah transaksi kasbon (debit)

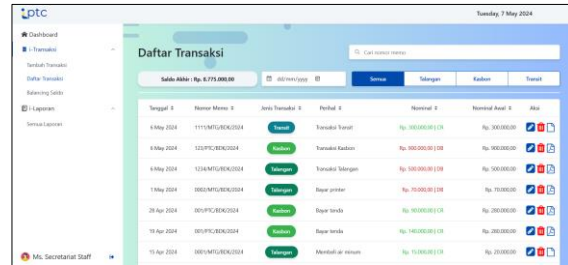


Gambar 11. Halaman tambah transaksi kasbon (kredit)

### 3.2.3. Halaman Daftar Transaksi

Halaman daftar transaksi di aplikasi I-PTC menampilkan semua transaksi *petty cash* yang dicatat oleh *user operator*. Pengguna dapat melihat detail

transaksi, seperti tanggal, nomor memo, jumlah, kategori, dan perihal, dalam tampilan yang terorganisir. Fitur pencarian dan filter memudahkan pengguna menemukan transaksi tertentu dengan cepat. Halaman ini ditampilkan pada Gambar 12.

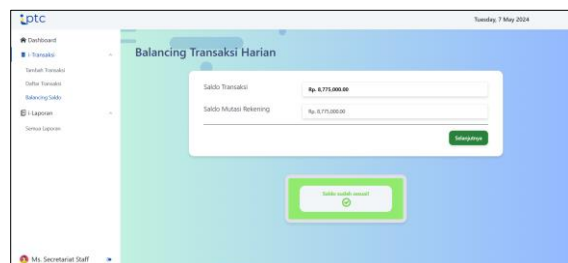


Gambar 12. Halaman daftar transaksi *petty cash*

Halaman ini juga memungkinkan pengguna untuk mengubah, menghapus, dan melihat bukti transaksi yang terunggah pada setiap transaksi *petty cash*. Selain itu, pengguna dapat melihat *file PDF* terunggah dengan menekan ikon PDF pada salah satu transaksi yang memiliki ikon PDF.

### 3.2.4. Halaman Pengajuan Laporan Harian

Pada halaman ini, operator mencocokkan saldo mutasi rekening dengan saldo transaksi aplikasi. Jika saldo tidak sesuai, *balancing* saldo gagal. Jika saldo mutasi rekening sesuai dengan saldo transaksi aplikasi, tahap *balancing* dapat dilanjutkan. Tampilan awal halaman *balancing* saldo terlihat pada Gambar 13.



Gambar 13. Halaman *balancing* saldo

Ketika tombol "Selanjutnya" ditekan, halaman peninjauan laporan harian akan ditampilkan. Laporan harian ini mencakup semua *transaksi petty cash* yang dicatat pada hari yang sama saat *balancing* saldo dilakukan. Pada halaman ini, pengguna dapat memasukkan judul laporan serta memilih *reviewer* dan *approver* laporan harian. Saat pengajuan dilakukan, aplikasi otomatis mengirimkan *e-mail* notifikasi kepada *reviewer*. Tampilan halaman ini dapat dilihat pada Gambar 14.

Ketika laporan harian *petty cash* berhasil diajukan, aplikasi akan mengirimkan notifikasi *e-mail* kepada *user* yang bersangkutan untuk meninjau laporan harian. Contoh dari *e-mail* yang akan terkirim kepada pengguna aplikasi I-PTC dilampirkan pada Gambar 15.

**LAPORAN PETTY CASH**  
Tanggal: 6 May 2024

Tanggal Transaksi	Nomor Memo	Jenis Transaksi	Perihal	Debit	Kredit
6 May 2024	1234/MTG/BDK/2024	Talangan	Transaksi Talangan	Rp. 500,000.00	
6 May 2024	123/PTC/BDK/2024	Kasbon	Transaksi Kasbon	Rp. 900,000.00	
6 May 2024	1111/MTG/BDK/2024	Transit	Transaksi Transit		Rp. 300,000.00
<b>Total</b>				<b>Rp. 1,400,000.00</b>	<b>Rp. 300,000.00</b>

Saldo Awal Rp. 9,875,000.00

---

Dana Gantungan

Tanggal Transaksi	Nomor Memo	Jenis Transaksi	Perihal	Nominal
15 April 2024	0001/PTC/BDK/2024	Talangan	Membeli air minum	Rp. 1,000.00
15 April 2024	001/PTC/BDK/2024	Kasbon	Bayar tenda	Rp. 50,000.00
1 May 2024	0002/MTG/BDK/2024	Talangan	Bayar printer	Rp. 70,000.00
6 May 2024	1234/MTG/BDK/2024	Talangan	Transaksi Talangan	Rp. 500,000.00

Judul Pengajuan Laporan : Pengajuan Transaksi Harian ke-20

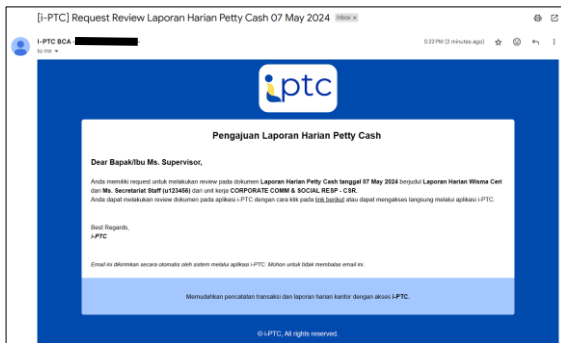
Operator : Ms. Secretariat Staff    Reviewer : Ms. Supervisor

Approver : Mr. Head Of Department

Saldo Akhir Rp. 8,775,000.00

Ajukan Laporan Petty Cash!

Gambar 14. Halaman tinjau laporan harian



Gambar 15. Contoh e-mail terkirim ke pengguna aplikasi I-PTC

### 3.2.5. Halaman Review Laporan Harian

**Pengajuan Laporan Petty Cash**  
Kode Pengajuan: KAT7MTVB54

Progress Pengajuan Laporan: 33%

Judul Pengajuan: Pengajuan Transaksi Harian ke-20

Saldo Awal: Rp. 9,875,000.00    Saldo Akhir: Rp. 8,775,000.00

File Bukti Terunggah: bukti1.pdf, bukti2.pdf

**Transaksi Petty Cash**

Tanggal Transaksi	Nomor Memo	Jenis Transaksi	Perihal	Debit	Kredit
6 May 2024	1234/MTG/BDK/2024	Talangan	Transaksi Talangan	Rp. 500,000.00	
6 May 2024	123/PTC/BDK/2024	Kasbon	Transaksi Kasbon	Rp. 900,000.00	
6 May 2024	1111/MTG/BDK/2024	Transit	Transaksi Transit		Rp. 300,000.00
<b>Total</b>				<b>Rp. 1,400,000.00</b>	<b>Rp. 300,000.00</b>

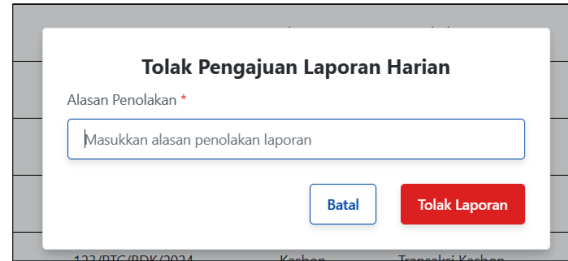
**Transaksi dengan Saldo Gantungan**

Tanggal Transaksi	Nomor Memo	Jenis Transaksi	Perihal	Nominal
15 April 2024	0001/MTG/BDK/2024	Talangan	Membeli air minum	Rp. 1,000.00
15 April 2024	001/PTC/BDK/2024	Kasbon	Bayar tenda	Rp. 50,000.00
1 May 2024	0002/MTG/BDK/2024	Talangan	Bayar printer	Rp. 70,000.00
6 May 2024	1234/MTG/BDK/2024	Talangan	Transaksi Talangan	Rp. 500,000.00

Tolak Laporan    Setujui Laporan

Gambar 16. Halaman review laporan harian

Setelah operator mengajukan laporan harian transaksi *petty cash* melalui halaman pengajuan laporan, laporan tersebut akan melewati dua tahap pengecekan oleh *reviewer* dan *approver*. Halaman ini memungkinkan pengguna untuk meninjau, menyetujui, atau menolak laporan sesuai kebutuhan. Halaman tersebut dapat dilihat pada Gambar 16.



Gambar 17. Pop-up input alasan penolakan laporan harian

User *reviewer* atau *approver* dapat menyetujui atau menolak laporan yang diterima. Jika laporan disetujui, akan tampil *pop-up* penyetujuan, dan laporan akan diteruskan kepada *user* selanjutnya. Namun, jika laporan ditolak, *user* dapat memasukkan alasan penolakan pada *pop-up* yang muncul seperti yang ditampilkan pada Gambar 17.

### 3.2.6. Halaman Semua Pengajuan Laporan

Laporan harian transaksi *petty cash* yang diajukan atau pernah diajukan pada aplikasi I-PTC akan tercatat dalam menu "Semua Laporan". Ada lima status pengajuan laporan harian: menunggu persetujuan, disetujui oleh *reviewer*, laporan diterima, laporan ditolak, dan terdapat pengajuan baru.

Pada halaman ini, ditampilkan progres dari pengajuan laporan harian dan detail dari laporan harian. Detail yang ditampilkan meliputi apakah pengguna telah meninjau laporan, kapan pengguna menyetujui atau menolak laporan, transaksi *petty cash* yang diajukan, saldo transaksi, serta opsi pembatalan pengajuan laporan harian. Ketika laporan harian sudah memiliki status "Laporan Diterima", akan muncul kotak yang memuat tombol untuk mengunduh laporan dalam format file PDF. Tampilan halaman detail laporan harian yang sudah diterima ditunjukkan pada Gambar 18.

**Pengajuan Laporan Petty Cash**  
Kode Pengajuan: QTYATGQ92H

Progress Pengajuan Laporan: Selesai dan Diterima

Judul Pengajuan: Pengajuan Laporan Harian ke-3

Laporan Harian: 19/04/2024

Saldo Awal: Rp. 9,775,000.00    Saldo Akhir: Rp. 9,800,000.00

Di-Ajukan Oleh: Ms. Secretariat Staff (20/04/2024, jam: 14:44)

Di-Review Oleh: Ms. Supervisor (20/04/2024, jam: 14:40)

Di-Approve Oleh: Mr. Head Of Department (20/04/2024, jam: 14:50)

Download Laporan

Gambar 18. Halaman detail laporan harian yang diterima

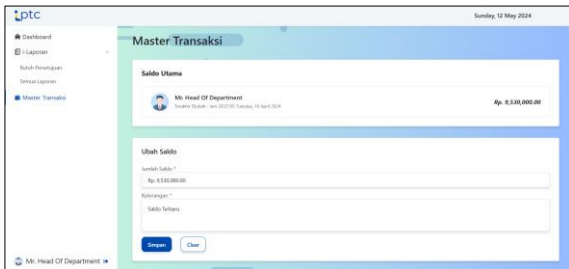
Dalam file PDF tersebut, terdapat laporan harian yang berisi transaksi *petty cash* dan transaksi dengan dana gantungan, disertai dengan detail dari pengajuan

laporan dan lampiran bukti transaksi *petty cash* yang telah diunggah sebelumnya. Tampilan dari file PDF laporan harian yang telah diunduh dapat dilihat pada Gambar 19.

Gambar 19. PDF laporan harian

### 3.2.7. Halaman Saldo Master Transaksi

Halaman terakhir dalam aplikasi I-PTC adalah halaman saldo master transaksi, yang berfungsi untuk mengatur saldo transaksi pada transaksi *petty cash* yang akan dicatat. Halaman ini hanya dapat diakses oleh user *approver* yang mana dapat mengubah saldo transaksi dan menambah keterangan perubahan saldo seperti pada Gambar 20.



Gambar 20. Halaman master transaksi.

### 3.3. Test

Tahapan pengujian sistem ini bertujuan untuk mengevaluasi kesesuaian rancangan dengan implementasinya, serta mengidentifikasi dan mengatasi potensi bug atau kesalahan pada aplikasi.

#### 3.3.1. User Acceptance Testing (UAT)

UAT (*User Acceptance Testing*) merupakan salah satu tahap penting dalam pengembangan aplikasi di mana pengguna akhir melakukan uji coba aplikasi secara menyeluruh. Tujuannya adalah untuk memastikan bahwa aplikasi memenuhi kebutuhan dan harapan pengguna serta dapat digunakan dengan nyaman dan efektif dalam lingkungan produksi yang sebenarnya.

Proses UAT melibatkan penggunaan formulir skenario *testing* untuk menguji fungsionalitas dan fitur aplikasi dengan kasus pengguna realistis. Formulir tersebut mencakup langkah-langkah konkret yang harus diikuti oleh pengguna, data masukan yang diperlukan, hasil yang diharapkan, dan kriteria keberhasilan. Berdasarkan uji coba yang telah

dilakukan, hasil dari UAT dapat dilihat pada Tabel 1 di bawah ini.

Tabel 1. Hasil UAT aplikasi I-PTC

<b>Jumlah Skenario Diuji</b>	127 Skenario
<b>Jenis Skenario</b>	Positif: 99 Skenario Negatif: 28 Skenario
<b>Skenario Positif</b>	Lulus semua dengan hasil yang diharapkan ( <i>feedback</i> positif)
<b>Skenario Negatif</b>	Hasil sesuai harapan ( <i>form</i> tidak dapat disimpan/muncul notifikasi kesalahan)
<b>Status Keberhasilan</b>	Semua skenario (127 skenario) mendapat status "Passed"
<b>Kesimpulan</b>	Secara fungsionalitaas, aplikasi I-PTC telah memenuhi seluruh kasus penggunaan yang diujicobakan

### 3.4. Deploy

Setelah proses pengujian selesai dan aplikasi I-PTC lolos dari tahap UAT, langkah selanjutnya adalah melakukan *deployment*. Proses *deployment* pada pengembangan aplikasi berbasis *low-code programming* dapat dilakukan dengan menekan tombol "One-Click Publish" di OutSystems *Service Studio*, seperti yang ditunjukkan pada Gambar 21.



Gambar 21. Proses *deployment* aplikasi di OutSystems

Aplikasi yang telah di-*publish* dapat diakses melalui link *environment*. Meskipun aplikasi sudah dapat diakses secara *online*, tetap dalam tahap pengembangan, sehingga jika masih terdapat malfungsi atau *bug*, aplikasi dapat diperbaiki dan di-*deploy* ulang.

### 3.5. Review

Tahap ini melibatkan evaluasi hasil pengembangan aplikasi I-PTC setelah tahap pengujian dan deployment. *Review* dilakukan untuk mengevaluasi performa aplikasi, respons pengguna, serta mengidentifikasi potensi *bug* atau penyempurnaan yang dapat dilakukan di masa mendatang.

Evaluasi ini dapat melibatkan *feedback* dari pengguna akhir, pemangku kepentingan, dan pengembang untuk memastikan aplikasi memenuhi ekspektasi dan kebutuhan yang diinginkan. Evaluasi dari pengembangan aplikasi dilakukan secara mingguan sesuai dengan rencana *sprint* sebelumnya.

Jika terdapat ketidaksesuaian pada aplikasi yang sedang dikembangkan, tahapan pengembangan akan kembali ke tahap design dan diteruskan kembali hingga tahap *review*. Namun, jika aplikasi sudah sesuai dengan kebutuhan, tidak ada *bug* atau kerusakan fungsi, dan dokumentasi aplikasi sudah lengkap, maka tahapan pengembangan dapat dilanjutkan ke tahap selanjutnya.



### 3.6. Launch

Tahap terakhir dalam pengembangan aplikasi I-PTC menggunakan metode *agile* adalah peluncuran aplikasi. Pada tahap ini, aplikasi yang telah melalui proses pengembangan, pengujian, dan *deployment* siap untuk digunakan oleh pengguna secara resmi. Peluncuran melibatkan penyiapan lingkungan produksi, pengujian sistem secara menyeluruh untuk memastikan tidak ada masalah yang tersisa, pelatihan pengguna, dan dukungan teknis awal untuk memastikan aplikasi berjalan lancar. Aplikasi yang telah diluncurkan akan terintegrasi dengan infrastruktur aplikasi *low-code programming* yang sudah ada sebelumnya. *Launch* juga menandai dimulainya fase pemeliharaan dan dukungan berkelanjutan untuk menjaga performa aplikasi tetap optimal.

## 4. DISKUSI

Penerapan *teknologi low-code programming* menggunakan platform OutSystems dalam pengembangan aplikasi pencatatan transaksi *petty cash* di PT Bank Central Asia Tbk (BCA) telah menghasilkan berbagai keuntungan signifikan dalam hal efisiensi dan efektivitas. Proses pengembangan yang biasanya memerlukan waktu lama dan tenaga besar dapat dipercepat dengan penggunaan antarmuka visual dan komponen *drag-and-drop* yang disediakan oleh OutSystems. Hasil pengujian menunjukkan bahwa aplikasi I-PTC mampu memenuhi kebutuhan dan harapan pengguna akhir, dengan semua skenario *User Acceptance Testing* (UAT) berhasil dilewati tanpa kesalahan signifikan. Hal ini membuktikan bahwa aplikasi tersebut stabil, andal, dan mudah digunakan.

Namun, tantangan seperti keterbatasan kustomisasi masih dihadapi, yang kadang memerlukan tambahan pengkodean manual. Temuan ini sejalan dengan penelitian oleh Ega Wachid Radiegyta et al. (2023), yang menunjukkan bahwa penggunaan konsep *low-code* pada platform Microgen mempercepat pengembangan aplikasi sistem manajemen pembelajaran secara signifikan, namun tetap memerlukan penyesuaian manual untuk fitur-fitur khusus [11].

Pendekatan *low-code* juga menunjukkan bahwa pengembangan aplikasi yang lebih cepat dan responsif terhadap kebutuhan pengguna seperti yang ditemukan oleh Singhal et al. (2023) dalam penelitian mereka tentang aplikasi mobile e-commerce menggunakan pendekatan *low/no-code*. Mereka menemukan bahwa pendekatan ini mempercepat dan menyederhanakan proses pengembangan, yang sesuai dengan pengalaman penelitian ini menggunakan OutSystems untuk aplikasi I-PTC [24].

Rekomendasi untuk pengembangan lebih lanjut mencakup pelatihan berkelanjutan bagi tim pengembang untuk memaksimalkan penggunaan platform, kombinasi pendekatan *low-code* dengan

pengembangan tradisional untuk mengatasi keterbatasan, pengujian berkelanjutan untuk memastikan performa aplikasi tetap optimal, peningkatan kolaborasi tim melalui metode *agile*, serta dukungan pemeliharaan yang proaktif untuk menangani masalah pasca peluncuran.

Dengan langkah-langkah ini, diharapkan pengembangan aplikasi di masa mendatang dapat memberikan nilai tambah yang lebih besar bagi perusahaan. Dengan mengacu pada penelitian-penelitian tersebut, jelas bahwa penggunaan teknologi *low-code programming* tidak hanya meningkatkan efisiensi dan efektivitas pengembangan aplikasi tetapi juga menghadirkan tantangan yang perlu diatasi melalui strategi pengembangan yang komprehensif dan kolaboratif.

## 5. KESIMPULAN

Berdasarkan penelitian tentang penerapan teknologi *low-code programming* dalam pengembangan aplikasi di PT Bank Central Asia Tbk, dapat disimpulkan bahwa aplikasi I-PTC berhasil dikembangkan sebagai solusi manajemen *petty cash* yang efisien. Aplikasi ini menyediakan berbagai fitur penting, termasuk pencatatan transaksi *petty cash*, pengelolaan saldo, pembuatan laporan harian dalam format PDF, serta proses persetujuan dan peninjauan laporan.

Implementasi metode *agile* terbukti efektif dalam menyelesaikan pengembangan aplikasi I-PTC sesuai kebutuhan pengguna dan telah terintegrasi dengan baik dalam infrastruktur IT perusahaan. Pendekatan ini memungkinkan penyesuaian cepat terhadap perubahan kebutuhan dan memastikan proses pengembangan yang iteratif serta kolaboratif. Penggunaan teknologi *low-code programming* melalui platform OutSystems mempermudah dan mempercepat proses pengembangan dengan menyediakan antarmuka visual, komponen *drag-and-drop*, serta *template* siap pakai, yang secara signifikan meningkatkan efisiensi.

Meskipun menghadapi tantangan karena masih tergolong baru dan referensi pembelajaran yang terbatas, teknologi *low-code programming* terbukti mampu memberikan solusi yang efektif dan efisien dalam pengembangan aplikasi bisnis kompleks, terutama dalam konteks pencatatan dan pengelolaan transaksi *petty cash* di PT Bank Central Asia Tbk.

## DAFTAR PUSTAKA

- [1] S. Wahyuni, V. Tasril, and J. P. J. Prayoga, "Desain Aplikasi Game Edukasi Pada Siswa Kelas 2 SD Negeri 024777 Binjai," *Warta Dharmawangsa*, vol. 16, no. 4, pp. 758–768, Oct. 2022, doi: 10.46576/wdw.v16i4.2431.
- [2] I. A. Tiawan and L. Afuan, "APLIKASI PENGELOLAAN KERJASAMA PEMBUATAN PROJEK PADA DINAS KOMUNIKASI DAN INFORMATIKA

- BANYUMAS,” *Jurnal Teknik Informatika (Jutif)*, vol. 1, no. 1, pp. 13–18, Jul. 2020, doi: 10.20884/1.jutif.2020.1.1.10.
- [3] A. Farisi, Yohannes, and Dafid, “PELATIHAN PEMBUATAN APLIKASI MOBILE TANPA CODING BAGI MAHASISWA POLTEKKES KEMENKES PALEMBANG,” *Jurnal Pengabdian Masyarakat Berkemajuan*, vol. 7, no. 1, pp. 372–379, Mar. 2023, [Online]. Available: <https://www.glideapps.com>.
- [4] PT Bank Central Asia Tbk, “Profil Perusahaan BCA,” Jakarta, 2022. [Online]. Available: [www.klikbca.com](http://www.klikbca.com)
- [5] T. C. Lethbridge, “Low-Code Is Often High-Code, So We Must Design Low-Code Platforms to Enable Proper Software Engineering,” in *Leveraging Applications of Formal Methods, Verification and Validation*, T. Margaria and B. Steffen, Eds., Cham: Springer International Publishing, Oct. 2021, pp. 202–212. doi: [https://doi.org/10.1007/978-3-030-89159-6\\_14](https://doi.org/10.1007/978-3-030-89159-6_14).
- [6] D. Lailatun Nisa and I. Fauzi, “Pengaruh Transformasi Digital Terhadap Kinerja Produk Aplikasi Low-Code No-Code B-Pro Bisnis Mahasiswa,” *Jurnal Riset Dan Publikasi Ilmu Ekonomi*, vol. 2, no. 2, pp. 293–305, 2024, doi: 10.61132/menawan.v2i2.402.
- [7] A. Budi, “IMPLEMENTASI E-COMMERCE MENGGUNAKAN LOW CODE PROGRAMMING PADA SAM’S FOODIE BERBASIS RESPONSIVE WEB,” *Jurnal Informatika dan Bisnis*, vol. 12, no. 1, Jun. 2023, doi: 10.46806/jib.v12i1.1001.
- [8] P. Rosado, “The #1 Low-Code Platform.” Accessed: Feb. 10, 2024. [Online]. Available: <https://www.outsystems.com/#:~:text=OutSystems%20is%20the%20most%20powerful,dev%20makes%20the%20impossible%20possible>.
- [9] N. Fitriyah, “ANALISIS PENGELOLAAN DANA KAS KECIL (PETTY CASH) PADA KLINIK MATA KMU LAMONGAN,” 2023.
- [10] S. Meida and N. Fathurrahmi Lawita, “Analisis Sistem dan Prosedur Pengelolaan Dana Kas Kecil pada PT. XYZ,” *Jurnal Ilmiah Mahasiswa Merdeka EMBA*, vol. 2, no. 2, 2023.
- [11] Ega Wachid Radiegyta, Daniel Hasiholan Tinambunan, Rido Dwi Kurniawan, and Richardus Eko Indrajit, “ACCELERATION OF LEARNING MANAGEMENT SYSTEM APPLICATION DEVELOPMENT IN THE EDUCATION SECTOR USING THE LOW CODE CONCEPT ON MICROGEN,” *Jurnal Teknik Informatika (Jutif)*, vol. 4, no. 4, pp. 913–922, Aug. 2023, doi: 10.52436/1.jutif.2023.4.4.1315.
- [12] A. Ariesta, Y. N. Dewi, F. A. Sariasih, and F. W. Fibriany, “PENERAPAN METODE AGILE DALAM PENGEMBANGAN APPLICATION PROGRAMMING INTERFACE SYSTEM PADA PT XYZ,” *Jurnal CoreIT: Jurnal Hasil Penelitian Ilmu Komputer dan Teknologi Informasi*, vol. 7, no. 1, p. 38, Jul. 2021, doi: 10.24014/coreit.v7i1.12635.
- [13] K. S. Haryana, “PENERAPAN AGILE DEVELOPMENT METHODS DENGAN FRAMEWORK SCRUM PADA PERANCANGAN PERANGKAT LUNAK KEHADIRAN RAPAT UMUM BERBASIS QR-CODE,” *Jurnal Computech & Bisnis*, vol. 13, no. 2, pp. 70–79, 2019.
- [14] C. Perkins, “Using Agile in Process Improvement,” LinkedIn. Accessed: Feb. 12, 2024. [Online]. Available: <https://www.linkedin.com/pulse/using-agile-process-improvement-charles-perkins/>
- [15] K. Anwar, L. D. Kurniawan, M. I. Rahman, and N. Ani, “Aplikasi Marketplace Penyewaan Lapangan Olahraga Dari Berbagai Cabang Dengan Metode Agile Development,” *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 2, pp. 264–274, Aug. 2020, doi: 10.32736/sisfokom.v9i2.905.
- [16] R. Gutama and T. Dirgahayu, “Implementasi Scrum Pada Manajemen Proyek Pengembangan Aplikasi Sistem Monitoring dan Evaluasi Pembangunan (SMEP),” Yogyakarta, 2020.
- [17] R. Martins, F. Caldeira, F. Sa, M. Abbasi, and P. Martins, “An overview on how to develop a low-code application using OutSystems,” in *2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE)*, IEEE, Oct. 2020, pp. 395–401. doi: 10.1109/ICSTCEE49637.2020.9277404.
- [18] Outsystems, “Outsystems 11 Service Studio Overview,” Outsystems Documentation. Accessed: Jun. 09, 2024. [Online]. Available: [https://success.outsystems.com/documentation/11/getting\\_started/service\\_studio\\_overview/](https://success.outsystems.com/documentation/11/getting_started/service_studio_overview/)
- [19] H. H. Hamria Hamka, “Game Edukasi Untuk Pembelajaran IPA SMP Kelas VIII Berbasis Android,” *JATISI (Jurnal Teknik Informatika dan Sistem Informasi)*, vol. 9, no. 1, pp. 274–288, Mar. 2022, doi:

- 10.35957/jatisi.v9i1.1491.
- [20] A. F. Rezy, M. Yuga Utama, and N. R. Ramadhan, "Pengembangan Aplikasi Klinik Berbasis Web Untuk Pengelolaan Rekam Medis Menggunakan Metode Agile," *Teknik dan Multimedia*, vol. 1, no. 2, 2023.
  - [21] A. Zalukhu, S. Purba, and D. Darma, "PERANGKAT LUNAK APLIKASI PEMBELAJARAN FLOWCHART," *Jurnal Teknologi Informasi dan Industri*, vol. 4, no. 1, pp. 61–70, Sep. 2023.
  - [22] K. A. Prasetyo and A. K. Nugroho, "WEB-BASED TECHNICAL SUPERVISION AND PLANNING REPORTS INFORMATION SYSTEM IN CV. TATA SAKA CONSULTANT," *Jurnal Teknik Informatika (JUTIF)*, vol. 3, no. 1, 2022, doi: 10.20884/1.jutif.2022.3.1.17.
  - [23] K. Afiihah, Z. F. Azzahra, and A. D. Anggoro, "Analisis Teknik Entity-Relationship Diagram dalam Perancangan Database: Sebuah Literature Review," *JURNAL INTECH*, vol. 3, no. 1, pp. 8–11, May 2022.
  - [24] R. Singhal, M. Thakur, L. Jaiswal, and S. Setia, "Exploring Smart Technologies for Native Mobile Application Development : Low/No- Code Approaches," Jun. 2023.