

Pengembangan Aplikasi Layanan Multiguna Menggunakan *Low-Code* Platform (Studi Kasus: Astra Credit Companies)

Azarya Abednego¹, Suyoto², Eddy Julianto³

Program Studi Informatika, Fakultas Teknologi Industri, Universitas Atma Jaya Yogyakarta Jl. Babarsari No. 43, Janti, Caturtunggal, Kec Depok, Kabupaten Sleman, 55281, Daerah Istimewa Yogyakarta, Indonesia

Email: [1abgdazarya@gmail.com](mailto:abgdazarya@gmail.com), [2suyoto@uajy.ac.id](mailto:suyoto@uajy.ac.id), [3eddy.julianto@uajy.ac.id](mailto:eddy.julianto@uajy.ac.id)

Abstrak. Perusahaan Astra Credit Companies merupakan salah satu perusahaan yang bergerak di industri pembiayaan mobil dan alat berat di Indonesia. Di dalamnya, terdapat Layanan pinjaman dana atau disebut juga layanan multiguna yang merupakan layanan pemberian pinjaman dana untuk masyarakat dengan jaminan BPKB, namun layanan tersebut masih dalam proses pengembangan. Dari permasalahan tersebut, penulis akan mengembangkan layanan pinjaman dana atau layanan multiguna di dalam laman situs *acc.co.id* menggunakan *Low-Code Platform*. *Low-Code Platform* dipilih sebagai platform pengembangan karena kecepatan dalam pembuatan sebuah perangkat lunak, sistem pembuatan User Interface secara drag and drop, bekerja dengan sistem cloud computing, dan banyak keunggulan lainnya.

Kata Kunci: Layanan Multiguna, *Low-Code Platform*, *Outsystems*

1. Pendahuluan

1.1. Latar Belakang

Ada beberapa cara yang ditempuh seseorang demi mendapatkan penghasilan lebih, seperti mencari pekerjaan sampingan, meminjam dana dari bank atau perusahaan layanan pinjaman dana, menjual kembali harta benda yang tidak terpakai, dan lain sebagainya. Cara yang paling populer atau sering digunakan di masyarakat adalah pinjaman dana atau sering disebut dengan kredit. Salah satu perusahaan yang menyediakan layanan peminjaman dana adalah Astra Credit Companies. Astra Credit Companies atau sering disingkat menjadi ACC, merupakan salah satu anak perusahaan dari PT. Astra Sedaya Finance yang bergerak di pembiayaan kendaraan transportasi dan alat berat di Indonesia. ACC memiliki bermacam-macam layanan yang tersedia bagi masyarakat umum seperti pembiayaan mobil baru dan pembiayaan multiguna, pembiayaan mobil bekas, pembiayaan syariah dan pembiayaan alat berat [1].

Pembiayaan multiguna atau disebut dengan *multi purposes financing*, merupakan layanan pengajuan pinjaman dana bagi masyarakat dengan jaminan berupa Buku Pemilik Kendaraan Bermotor atau disingkat dengan BPKB [2]. Masyarakat dapat mengajukan pinjaman multiguna dengan minimal pengajuan sebesar sepuluh juta rupiah dengan jaminan pinjaman dana tersebut, pembiayaan multiguna hanya memerlukan BPKB dari seseorang yang ingin melakukan pinjaman. Namun, laman situs pembiayaan multiguna masih terpisah dari laman situs utama *acc.co.id*, sehingga sangat sulit bagi masyarakat untuk mengetahui dan mengakses layanan tersebut.

Layanan multiguna merupakan layanan pengajuan pinjaman dana dari Astra Credit Companies untuk masyarakat dengan jaminan berupa BPKB. Layanan tersebut dapat diakses oleh pengguna secara melalui bagian produk dan layanan, lalu pengguna dapat memilih bagian pembiayaan multiguna, selanjutnya pengguna akan diteruskan ke laman situs *acc.co.id/fasilitasdanaacc* dan barulah pengguna dapat mengisi secara lengkap form dan berkas-berkas yang ada. Hal ini merupakan sebuah ketidakefektifan karena pengguna mengeksplorasi laman situs lebih dalam lagi dan membutuhkan waktu yang lebih lama. Oleh karena itu, laman situs layanan multiguna akan dibuat ulang dan dipindahkan ke laman situs *acc.co.id* dan layanan tersebut juga akan terpampang di halaman depan laman situs *acc.co.id*, sehingga masyarakat yang akan mengakses layanan tersebut tidak kesulitan dalam mencarinya dan waktu yang dikeluarkan untuk mencari layanan tersebut akan lebih cepat.

2. Tinjauan Pustaka

Penelitian pertama yang akan ditelusuri berjudul “*Low-code Platform for Automating Business Processes in Manufacturing*”, yang ditulis oleh Robert Waszkowski pada tahun 2019. *Low-Code Platform* yang digunakan pada penelitian bernama Aurea BPM, yang berfokus pada pengembangan *database*, pengembangan proses bisnis, dan pengembangan antaruka pengguna. Penggunaan *Low-Code Platform* dalam jurnal ini bertujuan untuk memberikan dukungan ekstensif untuk pemodelan, otomatisasi, mengelola dan mengoptimalkan proses bisnis. *Low-Code Platform Aurea BPM* dipilih oleh Robert Waszkowski sebagai penulis karena kurangnya *programmer* dan tingkat persyaratan yang meningkat untuk frekuensi perubahan yang ada di lingkungan sistem teknologi informasi [3].

Penelitian kedua berjudul “*OutSystems as a Rapid Application Development Platform for Mobile and Web Applications*”, yang ditulis oleh Dmitry Golovin pada tahun 2017. Tujuan penelitian ini adalah mengatasi ketidakpuasan mahasiswa *Lahti University of Applied Sciences* terhadap layanan mahasiswa yang ada pada aplikasi *mobile* dengan menggunakan *OutSystems*. *OutSystems* digunakan oleh Dmitry Golovin karena didalamnya memuat dua hal penting yaitu server dan aplikasi desktop untuk pengembang. Aplikasi desktop memungkinkan pengembang untuk membuat model *database* sederhana, alur bisnis proses, dan aturan didalam aplikasi yang dikembangkan, serta menyebarkannya ke server produksi atau ke digital platform yang ada. *OutSystems* dapat mengimplemetasikan layanan mahasiswa yang ada di *Lahti University of Applied Sciences* sedemikian rupa sehingga layanan tersebut dapat digunakan mahasiswa dengan puas dan *OutSystems* kompatibel dengan semua perangkat *mobile* baik yang berbasis iOS, maupun yang berbasis Android [4][5][6].

Penelitian ketiga dilakukan oleh Tatu Virta pada tahun 2018, dengan judul “*Relation of low-code development to standard software development: Case Biit Oy Master’s*”. Penelitian ini mempunyai beberapa tujuan antara lain mengetahui pengembangan *Low-Code*, memungkinkan *Low-Code* ditawarkan kepada para pengembang perangkat lunak, platform apa yang memiliki *Low-Code* didalamnya, dan bagaimana *Low-Code* berbeda dengan pengembangan perangkat lunak standar. Keuntungan yang didapatkan setelah menggunakan Perangkat lunak *Low-Code Salesforce* adalah dapat menyederhanakan proses otomatisasi, kecepatan pengembangan meningkat, dan tidak diperlukan unit tes didalamnya. Namun, perangkat lunak *Low-Code Salesforce* juga memiliki beberapa kerugian dalam penggunaannya seperti alur kerja didalamnya tidak dapat menangani volume data yang besar, ada beberapa karyawan di Biit, yang diwawancarai oleh Tatu Virta lebih memilih untuk melakukan pengkodean secara manual karena lebih efisien dan lebih cepat untuk dibuat, dan lain sebagainya [7][8].

Penelitian keempat berjudul “*Next Generation Methods for Development of Enterprise Information Systems*” yang ditulis oleh Barbora Jancovicova pada tahun 2019. Penelitian mengangkat pengembangan sebuah sistem informasi di sebuah perusahaan dengan menggunakan *Low-Code platform*. *Low-Code platform* memungkinkan pengguna untuk membuat aplikasi sederhana tanpa memahami secara mendalam tentang bahasa pemrograman, sehingga pengguna dapat menghasilkan aplikasi yang operasional dengan platform ini. Salah satu hal penting dalam pemilihan *Low-Code platform* adalah platform tersebut menyediakan versi uji coba gratis yang dapat digunakan untuk mencoba fungsionalitasnya. *Low-Code platform* yang bernama *Mendrix* dipilih oleh Barbora Jancovicova, karena platform tersebut memiliki posisi kuat di pasar, memiliki versi gratis yang bisa digunakan untuk pengembangan dan terdapat tutorial serta dokumentasi ekstensif guna mendukung pengembangan sebuah aplikasi [9][10].

3. Metodologi Penelitian

Google Venture mencoba menerapkan versi praktis dari *design thinking*, yang dikenal dengan sebutan *Design Sprint*. *Design Sprint* atau sering disebut dengan *Sprint* merupakan salah satu metode singkat dalam membangun sebuah produk dan prototipe perangkat lunak dalam tahapan lima hari. Dalam lima hari waktu pengerjaan tersebut banyak hal yang harus dikerjakan mulai konsep produk, prototipe, sampai menciptakan fitur-fitur didalamnya. Tahapan yang terdapat di dalam metode ini antara lain [11] : (1) *Understand* merupakan tahap yang digunakan

untuk menyamakan persepsi terhadap produk yang ada. Calon pengguna akan memasuki tahap wawancara dan hasilnya akan didiskusikan bersama semua anggota tim Hasil diskusi tersebut menciptakan sebuah struktur permasalahan yang nantinya dipecah di hari berikutnya. (2) *Diverge*, merupakan proses yang dilakukan masing-masing individu untuk melontarkan ide sebanyak-banyaknya dan akan dirancang secara kasar melalui kertas agar anggota tim memiliki bayangan terhadap ide tersebut. (3) *Decide*, dalam tahapan ini, seluruh anggota tim akan berkumpul untuk memilih hasil rancangan mana yang terbaik, dan rancangan yang memiliki suara terbanyak akan diperbaiki dan di desain lebih rapi supaya dapat dijadikan prototipe. (4) *Prototype*, tahapan ini akan lebih banyak dikerjakan oleh tim *developer*, karena mereka yang akan mengubah desain menjadi prototipe, serta prototipe dibuat secara cepat supaya dapat menampilkan fungsionalitas dari produk. (5) *Validate*, tahap terakhir, calon pengguna akan mencoba dan menguji prototipe yang ada. Hasil dari tahap validasi menentukan proses iterasi selanjutnya.

4. Hasil dan Diskusi

4.1. Analisis Perangkat Lunak

Aplikasi layanan multiguna akan dikembangkan bermanfaat untuk mempermudah masyarakat awam yang belum pernah sama sekali mengakses layanan multiguna yang ada di dalam Perusahaan Astra Credit Companies. Lalu aplikasi tersebut akan masuk ke dalam proses pengujian perangkat lunak. Pengujian tidak dilakukan langsung oleh masyarakat karena proyek pengembangan aplikasi layanan multiguna masih ada di dalam *environment development*, sehingga pengujian akan dilakukan oleh *Quality Control* dengan proses UAT atau *User Acceptance Test*. *User Acceptance Test* merupakan proses pengujian aplikasi kepada pengguna untuk memeriksa apakah solusi dari sebuah fungsi sesuai dengan kebutuhan pengguna. Proses UAT akan dilakukan pada keseluruhan laman web acc.co.id termasuk layanan simulasi multiguna didalamnya.

Tabel 1. Hasil Pengujian UAT

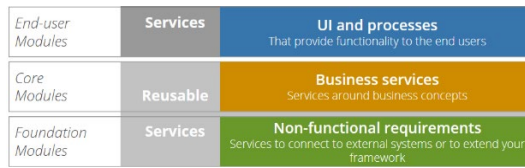
Identifikasi	Deskripsi	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria Evaluasi hasil	Hasil yang didapat	Kesimpulan
P-01-01	Pengujian terhadap halaman <i>Homepage</i>	Aplikasi akan dijalankan melalui Service Studio Outsystems	-	Aplikasi muncul pada browser dan muncul halaman <i>Homepage</i>	Aplikasi muncul pada browser dan muncul halaman <i>Homepage</i>	Aplikasi muncul pada browser dan muncul halaman <i>Homepage</i>	Handal
P-02-01	Pengujian terhadap halaman <i>Homepage</i> menuju halaman Simulasi Kredit	Memilih Simulasi Kredit	Menekan tab pada bagian "Simulasi Kredit"	Aplikasi masuk ke halaman Simulasi Kredit	Aplikasi masuk ke halaman Simulasi Kredit	Aplikasi masuk ke halaman Simulasi Kredit	Handal

Identifikasi	Deskripsi	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria Evaluasi hasil	Hasil yang didapat	Kesimpulan
P-03-01	Pengujian terhadap halaman Simulasi kredit menuju bagian Fasilitas Dana	Memilih layanan fasilitas dana pada halaman Simulasi Kredit	Menekan pilihan “Fasilitas Dana”	Muncul form pengisian simulasi kredit fasilitas dana	Muncul form pengisian simulasi kredit fasilitas dana	Muncul form pengisian simulasi kredit fasilitas dana	Handal
P-03-01	Pengujian terhadap form Fasilitas Dana	-Memilih tahun pada dropdown brand -Memilih brand pada dropdown brand -Memilih tipe pada dropdown tipe - Memasukkan nilai pinjaman -Memilih tenor pada dropdown tenor	-Pilih tahun -Pilih brand mobil -Pilih tipe mobil -Mengisi nilai pinjaman -Pilih tenor	Muncul hasil perhitungan angsuran per bulan	Muncul hasil perhitungan angsuran per bulan	Muncul hasil perhitungan angsuran per bulan	Handal
P-03-02	Pengujian terhadap form Fasilitas Dana	Dropdown tahun tidak dipilih	Dropdown tahun tidak dipilih	Muncul warning “Tahun harus dipilih”	Muncul warning “Tahun harus dipilih”	Muncul warning “Tahun harus dipilih”	Handal
P-03-03	Pengujian terhadap form Fasilitas Dana	Dropdown brand mobil tidak dipilih	Dropdown brand mobil tidak dipilih	Muncul warning “Brand harus dipilih”	Muncul warning “Brand harus dipilih”	Muncul warning “Brand harus dipilih”	Handal

Identifikasi	Deskripsi	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria Evaluasi hasil	Hasil yang didapat	Kesimpulan
P-03-04	Pengujian terhadap form Fasilitas Dana	Dropdown tipe tidak dipilih	Dropdown tipe tidak dipilih	Muncul warning “Tipe harus dipilih”	Muncul warning “Tipe harus dipilih”	Muncul warning “Tipe harus dipilih”	Handal
P-03-05	Pengujian terhadap form Fasilitas Dana	Nilai pinjaman kurang dari 10 juta rupiah	Nilai pinjaman kurang dari 10 juta rupiah	Muncul warning “Nilai pinjaman harus lebih dari 10 juta rupiah”	Muncul warning “Nilai pinjaman harus lebih dari 10 juta rupiah”	Muncul warning “Nilai pinjaman harus lebih dari 10 juta rupiah”	Handal
P-03-06	Pengujian terhadap form Fasilitas Dana	Dropdown tenor tidak dipilih	Dropdown tenor tidak dipilih	Muncul warning “Tenor harus dipilih”	Muncul warning “Tenor harus dipilih”	Muncul warning “Tenor harus dipilih”	Handal
P-04-01	Pengujian terhadap form Fasilitas Dana menuju form pengisian data diri	Setelah nilai angsuran per bulan muncul, maka tombol “Ajukan Sekarang” muncul	Menekan tombol “Ajukan Sekarang”	<i>Pop-up</i> form pengisian data diri muncul	<i>Pop-up</i> form pengisian data diri muncul	<i>Pop-up</i> form pengisian data diri muncul	Handal
P-05-01	Pengujian terhadap form pengisian data diri	-Mengisi nama lengkap sesuai KTP -Mengisi nomor handphone yang aktif -Mengisi alamat email	-Mengisi nama lengkap sesuai KTP -Mengisi nomor handphone yang aktif -Mengisi alamat email	Pop-up “Berhasil Ajukan” muncul	Pop-up “Berhasil Ajukan” muncul	Pop-up “Berhasil Ajukan” muncul	Handal

Identifikasi	Deskripsi	Prosedur pengujian	Masukan	Keluaran yang diharapkan	Kriteria Evaluasi hasil	Hasil yang didapat	Kesimpulan
P-05-02	Pengujian terhadap form pengisian data diri	Nama sesuai KTP tidak di isi	Nama sesuai KTP tidak di isi	Muncul warning "Nama sesuai KTP harus di isi"	Muncul warning "Nama sesuai KTP harus di isi"	Muncul warning "Nama sesuai KTP harus di isi"	Handal
P-05-03	Pengujian terhadap form pengisian data diri	Nama sesuai KTP kurang dari 3 karakter	Nama sesuai KTP kurang dari 3 karakter	Muncul warning "Nama sesuai KTP harus lebih dari 3 karakter"	Muncul warning "Nama sesuai KTP harus lebih dari 3 karakter"	Muncul warning "Nama sesuai KTP harus lebih dari 3 karakter"	Handal
P-05-04	Pengujian terhadap form pengisian data diri	Nomor handphone tidak di isi	Nomor handphone tidak di isi	Muncul warning "Nomor handphone harus di isi"	Muncul warning "Nomor handphone harus di isi"	Muncul warning "Nomor handphone harus di isi"	Handal
P-05-05	Pengujian terhadap form pengisian data diri	Email tidak di isi	Email tidak di isi	Muncul warning "Email harus di isi"	Muncul warning "Email harus di isi"	Muncul warning "Email harus di isi"	Handal
P-06-01	Pengujian terhadap Pop-up berhasil ajukan menuju homepage	Pop-up "Berhasil Ajukan" muncul dan menekan tombol "Kembali ke Homepage"	Menekan tombol "Kembali ke Homepage"	Muncul halaman Homepage	Muncul halaman Homepage	Muncul halaman Homepage	Handal

4.2. Perancangan Perangkat Lunak



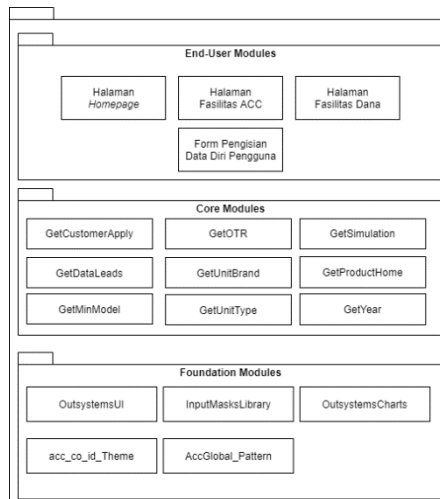
Gambar 1. The Architecture Canvas

Perancangan perangkat lunak akan menggunakan arsitektur yang dimiliki Outsystems yaitu *The Architecture Canvas*, seperti yang terlihat pada Gambar 1. *The Architecture Canvas* merupakan desain arsitektur berorientasi layanan sederhana atau disebut *Service-Oriented Architectures (SOA)*, yang memungkinkan untuk sebuah *service* dapat digunakan kembali di modul aplikasi yang berbeda. *The Architecture Canvas* yang dimiliki Outsystems memiliki tiga lapisan utama atau *Layers* dan lapisan utama tersebut terbagi lagi atas beberapa *Sub Layers*, seperti pada Gambar 2 [12].



Gambar 2. Sub Layers dari The Architecture Canvas

Bagian *End-user Modules* akan berisikan layar antarmuka yang sudah di desain dan layar antarmuka tersebut digunakan sebagai penghubung antara *Users* dengan sistem didalamnya. Bagian *End-user Modules* tidak dapat digunakan berulang-ulang karena bagian tersebut merupakan antarmuka utama dari sebuah aplikasi tertentu. Bagian *Core Modules* merupakan bagian inti dimana semua proses bisnis berada didalamnya. *Core Modules* dapat digunakan berulang-ulang di semua modul di Outsystems, baik pada modul di aplikasi yang sama maupun pada modul di aplikasi yang berbeda. Bagian *Foundation Modules* merupakan modul atau fungsi yang berasal dari Outsystems sebagai layanan yang diberikan Outsystems kepada penggunanya, seperti *plug-ins* atau fungsi tambahan yang diperlukan saat pembentukan aplikasi. Aplikasi yang dikembangkan dengan *The Architecture Canvas* sehingga aplikasi akan terbagi menjadi *End-user Modules*, *Core Modules*, dan *Foundation Modules*, seperti yang ditunjukkan pada Gambar 3.

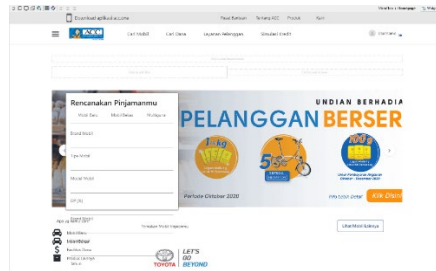


Gambar 3. Package Diagram

4.3. Implementasi Sistem

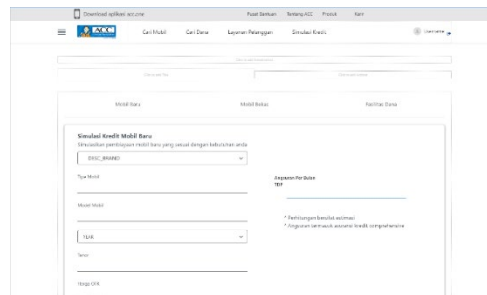
Bagian implementasi sistem ini akan menampilkan dan menjelaskan antarmuka dari fungsi yang telah dibangun. Terdapat juga gambar dari fungsi yang terdapat pada antarmuka dan kegunaan dari fungsi tersebut.

Halaman *homepage* atau tampilan utama akan menampilkan beberapa layanan yang ada di dalam Astra Credit Company. Halaman ini sudah dikerjakan oleh pengembang lain dan penulis hanya akan menggunakannya sebagai tampilan awal saja, seperti yang terlihat pada Gambar 4.



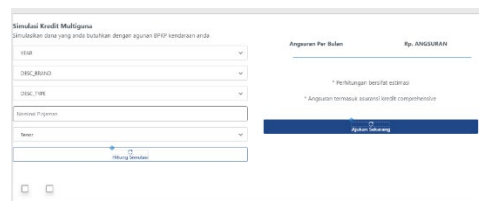
Gambar 4. Halaman Homepage

Halaman simulasi kredit merupakan halaman yang menampilkan layanan-layanan di dalam Astra Credit Companies. Terdapat tiga layanan utama yaitu simulasi mobil baru, simulasi mobil bekas, dan simulasi fasilitas dana. Halaman simulasi kredit yang akan ditampilkan akan terlihat seperti pada Gambar 5.



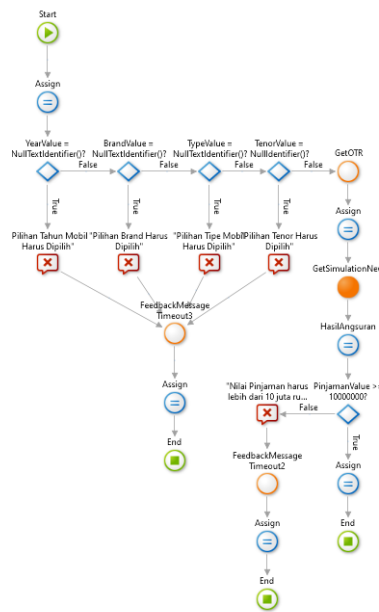
Gambar 5. Halaman simulasi kredit

Halaman antarmuka fasilitas dana akan berisikan form yang digunakan untuk melakukan simulasi fasilitas dana dan mengajukan simulasi fasilitas dana. Form tersebut berisikan beberapa *input* antara lain tahun, brand mobil, tipe mobil, tenor bulan dan nominal pinjaman. Setelah semua *input* telah terisi maka tombol hitung simulasi dapat di tekan dan perhitungan untuk angsuran bulanan akan muncul. Dari perhitungan tersebut maka pengguna dapat mengajukan simulasi tersebut dengan menekan tombol ajukan sekarang. Halaman antarmuka fasilitas dana akan terlihat seperti pada Gambar 6.



Gambar 6. Form Simulasi Kredit Fasilitas Dana

Fungsi yang dilakukan pada tombol hitung simulasi adalah perhitungan angsuran bulanan dari *input* yang telah dilakukan oleh pengguna akan terlihat seperti pada Gambar 7. Fungsi tersebut digunakan untuk melakukan perhitungan angsuran per bulan berdasarkan input yang dilakukan pengguna di form simulasi.



Gambar 7. Alur Fungsi Hitung Simulasi

5. Kesimpulan dan Saran

Beberapa keuntungan yang didapatkan saat mengembangkan aplikasi dengan *Low-code* platform antara lain : (1) *Low-code* platform Outsystems memiliki kemampuan untuk mengatasi *input* yang diluar batas kenormalan, seperti mengatasi input yang kosong dan input angka tidak dapat di isi huruf. (2) *Low-code* platform Outsystems dapat mengatasi perubahan fitur atau desain dengan cepat, seperti perubahan dari yang sebelumnya ada *dropdown* area menjadi tidak ada *dropdown* area.

Namun, pada tahap pengujian juga diperoleh beberapa kerugian yang didapatkan saat mengembangkan aplikasi dengan *Low-code* platform antara lain : (1) Jika memerlukan *plug-ins* atau fungsi pendukung lainnya pada fase pengembangan, maka pengembang harus mencarinya pada *forge* Outsystems dan mengimplementasikannya. Jika fungsi pendukung atau *plug-ins* tersebut tidak ada, maka pengembang harus membuatnya sendiri. (2) Tema dari aplikasi yang sedang dikembangkan sering bertabrakan dengan tema aplikasi yang terbawa saat pembuatan aplikasi pada tahap awal. (3) Saat proses pengembangan antarmuka pada *Low-Code* platform Outsystems, antarmuka yang di desain saling bertumpuk satu dengan yang lainnya, sehingga sulit untuk mendesainnya.

Berdasarkan hasil implementasi yang telah dilakukan sebelumnya, maka kesimpulan yang didapatkan antara lain : (1) *Low-code* platform Outsystems mampu mengembangkan layanan multiguna pada laman web acc.co.id. Terbukti dari hasil pengujian UAT yang telah dilakukan oleh *Quality Control*. (2) *Low-code* platform Outsystems juga mampu mengatasi perubahan-perubahan cepat yang terjadi selama masa pengembangan berlangsung.

Referensi

- [1] PT. Astra Sedaya Finance, *Laporan Tahunan Astra Sedaya Finance 2019*. Jakarta: PT. Astra Sedaya Finance, 2019.
- [2] Astra Credit Companies, “KREDIT MULTIGUNA - Astra Credit Companies.” <https://www.acc.co.id/fasilitasdanaacc/> (accessed Nov. 09, 2020).
- [3] R. Waszkowski, “Low-Code Platform For Automating Business Processes In Manufacturing,” *IFAC-PapersOnLine*, vol. 52, no. 10, pp. 376–381, 2019, doi: 10.1016/j.ifacol.2019.10.060.
- [4] D. Golovin, “OutSystems as a Rapid Application Development Platform for Mobile and Web Applications,” Lahti University of Applied Sciences, 2017.

- [5] OutSystems, “Architecture | Evaluation Guide | OutSystems.” <https://www.outsystems.com/evaluation-guide/architecture/> (accessed Jan. 06, 2021).
- [6] OutSystems, “What is Low-Code? The Full Guide to Low-Code | OutSystems.” <https://www.outsystems.com/low-code-platforms/> (accessed Jan. 05, 2021).
- [7] T. Virta, “Relation Of Low-Code Development To Standard Software Development: Case Biit Oy,” Lappeenranta University of Technology.
- [8] Salesforce, “Salesforce Launches New Low-Code Tools on the Lightning Platform Empowering Teams to Collaborate and Build Apps Fast,” *September 13, 2018*. <https://www.salesforce.com/news/press-releases/2018/09/13/salesforce-launches-new-low-code-tools-on-the-lightning-platform-empowering-teams-to-collaborate-and-build-apps-fast/> (accessed Jan. 06, 2021).
- [9] B. Jančovičová, “Next Generation Methods for Development of Enterprise Information Systems,” Czech Technical University, 2019.
- [10] Mendix, “Mendix Features & Tools for Professional Developers.” <https://www.mendix.com/professional-developers/> (accessed Jan. 16, 2021).
- [11] K. Olla, “Ciptakan Produkmu dalam 5 Hari dengan Design Sprint,” 2017. <https://www.jagoanhosting.com/blog/ciptakan-produkmu-dalam-5-hari-dengan-design-sprint/> (accessed Jan. 06, 2021).
- [12] Outsystems, “The Architecture Canvas - OutSystems.” https://success.outsystems.com/Support/Enterprise_Customers/Maintenance_and_Operations/Designing_the_Architecture_of_Your_OutSystems_Applications/The_Architecture_Canvas (accessed Mar. 22, 2021).