

# Penerapan Low-Code Platform dalam Pengembangan Aplikasi Presensi

(Studi Kasus: PT. Astra Sedaya Finance)

Ghifar Maulana Akbar  
Jurusan Informatika  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
[18523270@students.uii.ac.id](mailto:18523270@students.uii.ac.id)

Moh. Idris, S.Kom., M.Kom.  
Jurusan Informatika  
Universitas Islam Indonesia  
Yogyakarta, Indonesia  
[moh.idris@uii.ac.id](mailto:moh.idris@uii.ac.id)

Permasalahan wabah Covid-19 menjadi permasalahan yang sangat berdampak bagi beberapa perusahaan. Berkurangnya produktivitas suatu perusahaan banyak diakibatkan dari permasalahan internal yang terjadi, seperti contoh kurangnya pengawasan atau kontrol terhadap pegawai yang diakibatkan dari penerapan sistem *Work from Home* yang dianjurkan oleh pemerintah untuk setiap perusahaan dikarenakan oleh pandemi. Salah satunya yaitu pengawasan terhadap presensi atau kehadiran serta pencatatan laporan kinerja yang dilakukan oleh para pegawai perusahaan itu sendiri. Teknologi yang kian meningkat penggunaannya menjadi target solusi untuk mengatasi permasalahan yang terjadi. Oleh sebab itu dikembangkannya ACCAbsensi yaitu aplikasi yang memiliki fungsi utama untuk melakukan pencatatan presensi serta pelaporan kinerja secara online hanya dengan menggunakan *smartphone* saja. Aplikasi ini dikembangkan oleh Astra Credit Companies dalam mengatasi salah satu permasalahan yang terjadi pada situasi pandemi Covid-19. Dengan menggunakan *low-code platform Outsystems*, aplikasi dapat dikembangkan dengan cepat dan tepat tanpa harus mengetikkan banyak kode secara umumnya.

## I. PENDAHULUAN

Pada awal tahun 2020, dunia dihebohkan dengan wabah pandemi Covid-19 yang melanda dan datang tiba-tiba. Semakin besarnya penyebaran virus Corona yang sudah ada di Indonesia mendesak pemerintah pusat maupun pemerintah daerah mengambil langkah preventif dengan tujuan memutus rantai penularan virus tersebut. Salah satunya dengan menetapkan pembatasan penerapan kerja secara *Work from Office* (WFO) serta penerapan kerja secara *Work from Home* (WFH).

Dihadapkan dengan kondisi yang diakibatkan oleh wabah Covid-19 membuat beberapa pegawai mengalami kesulitan menerapkan sistem bekerja *WFH*. Hal yang membuat sulitnya bekerja di rumah bisa disebabkan oleh berbagai alasan seperti gangguan dalam mengerjakan pekerjaan rumah, mengurus anak, atau aktivitas anggota keluarga lainnya yang membuat tidak fokus menjalankan *WFH*. Selain itu, timbulnya rasa bosan dan jenuh membuat pegawai menjadi tidak fokus terhadap hal yang seharusnya dikerjakan. Pegawai menjadi mudah terdistraksi untuk melakukan hal lainnya. Karena hal tersebut, pegawai menjadi mudah lelah dan lebih stres yang mengakibatkan berkurangnya produktivitas dalam bekerja. Batasan antara kehidupan pribadi dengan pekerjaan menjadi tidak terkontrol, isolasi dengan kehidupan sosial dan profesional serta kesempatan untuk berbagi informasi menjadi kelemahan saat bekerja dilakukan di rumah [1].

Penerapan *WFH* menjadi tantangan tersendiri bagi perusahaan untuk mengontrol dan mengawasi kinerja para karyawannya. Salah satu hal yang memerlukan pengawasan adalah terkait kehadiran. Perlunya pengawasan dalam kehadiran sangat dibutuhkan dikarenakan saat penerapan *WFH* pengawasan yang dilakukan menjadi minim. Selain itu, karyawan juga akan menjadi rentan terhadap rasa malas selama bekerja dari rumah karena merasa tidak diawasi langsung selayaknya ketika bekerja di kantor. Jika hal tersebut berkelanjutan maka akan berakibat fatal bagi perusahaan. Kinerja para karyawan bisa menurun sehingga proses bisnis perusahaan akan terdampak. Maka dari itu, perusahaan harus segera mencari sebuah solusi untuk membantu meminimalkan dampak *WFH* terhadap kinerja para karyawan terutama yang berkaitan dengan pengawasan selama *WFH*.

Perusahaan Astra Credit Companies atau biasa di singkat dengan ACC adalah perusahaan pembiayaan mobil dan alat berat dalam kondisi baru ataupun bekas. Perusahaan tersebut menjadi salah satu dari sekian perusahaan yang terdampak wabah Covid-19. Terpaksa menerapkan sistem bekerja secara *Work from Home*, dilakukan demi memutus rantai penyebaran Covid-19. Diharuskannya merubah metode bekerja secara mendadak, dari bekerja yang dilakukan di kantor menjadi dilakukan di rumah, memunculkan permasalahan internal yang terjadi pada perusahaan Astra Credit Companies dalam mengontrol kinerja para pegawainya. Dilakukannya pengontrolan terhadap karyawan oleh atasan dengan cara manual, membuat pengawasan menjadi sedikit memakan waktu banyak yang lama-kelamaan membuat para atasan menjadi tidak sanggup untuk menangani kinerja satu-persatu para karyawannya. Maka dari itu, perusahaan Astra Credit Companies membuat suatu sistem yang diharapkan dapat menangani permasalahan internal yang terjadi, sistem tersebut diberi nama ACCAbsensi.

ACCAbsensi adalah aplikasi berbasis *mobile* yang diharapkan menjadi salah satu solusi dalam memecahkan permasalahan yang terjadi di perusahaan ACC. Dari aplikasi ini nantinya pencatatan terhadap kehadiran kerja yang dilakukan secara *online* dari karyawan akan disimpan. Selain itu, juga terdapat fitur tambahan seperti pelaporan tugas-tugas yang dikerjakan tiap karyawan setiap harinya.

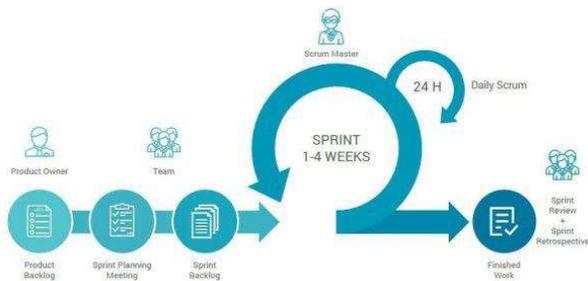
Makalah ini akan menjelaskan tentang pengembangan aplikasi ACCAbsensi dalam rangka untuk mengatasi salah satu permasalahan yang terjadi di perusahaan ACC. Pengembangan aplikasi ini nantinya akan menggunakan platform *mobile Outsystems* dikarenakan sistem yang dikembangkan harus digunakan sesegera mungkin oleh

perusahaan. Maka dari itu waktu pengembangan yang singkat dan penggunaan platform yang memudahkan pengembangnya menjadikan platform *Outsystems* dipilih untuk digunakan dalam pengembangan aplikasi ACCAbsensi.

## II. LANDASAN TEORI

### A. Scrum

*Scrum* merupakan kerangka kerja yang digunakan untuk mengelola pengembangan produk kompleks dari awal tahun 1990-an. *Scrum* menggunakan pendekatan bertahap (*incremental*) dan berkala (*iterative*) untuk meningkatkan *prediktabilitas* dan pengendalian resiko [2]. Bukan sebuah proses, teknik, ataupun metodologi, akan tetapi *scrum* adalah sebuah kerangka kerja di mana di dalamnya dapat menggunakan beberapa proses dan teknik. *Scrum* dapat mengekspos ketidak-efektifan manajemen produk dan teknik kerja yang dilakukan, sehingga dari hal tersebut secara terus menerus akan mampu meningkatkan kinerja produk. Proses pengembangan pada *Scrum* dikelompokkan ke dalam 5 fase seperti pada Gambar 1.



Gambar 1. Fase Scrum

### B. SIT

*System Integration Testing* atau biasa disingkat menjadi *SIT*. Merupakan suatu aktivitas pengujian atau *testing* yang dilakukan dengan tujuan untuk memastikan suatu aplikasi yang sedang diuji mampu sukses berinteroperasi dengan *software* lain yang telah ditentukan [3]. Dapat dipahami bahwa *SIT* adalah suatu aktivitas *testing* terhadap suatu sistem secara menyeluruh yang meliputi banyak komponen subsistem. Dari aktivitas ini, dapat dipastikan bahwa aplikasi yang diuji mampu beroperasi dengan sukses. *SIT* menerapkan teknik *black box testing* (metode pengujian perangkat lunak yang berfokus pada sisi fungsionalitas, khususnya pada input dan output aplikasi) dan menguji interoperabilitas sistem secara high-level. Dalam menemukan *error* dan *defect*, penggunaan teknik *negative testing* dan *error guessing* sangat umum digunakan pada fase *testing* ini.

### C. UAT

Menurut William E. Perry, *User Acceptance Testing* atau yang bisa disingkat menjadi *UAT* adalah aktivitas pengujian yang dilakukan oleh *end-user*, dimana *user* tersebut yang secara langsung berinteraksi dengan sistem. Verifikasi juga dilakukan untuk menguji fungsi dari sistem apakah telah berjalan sesuai dengan kebutuhan ataupun fungsinya [4].

*User Acceptance Testing* umumnya berusaha menunjukkan bahwa sistem telah memenuhi persyaratan tertentu. *User Acceptance Testing* merupakan pengujian yang dilakukan oleh *user* dengan menggunakan teknik pengujian

*black box* untuk menguji sistem terhadap spesifikasinya. Hal tersebut dilakukan sebagai upaya dalam memastikan semua fungsionalitas yang relevan telah diuji.

Dari beberapa definisi dapat disimpulkan bahwa *User Acceptance Testing* adalah aktivitas pengujian yang dilakukan oleh *user* dari sistem tertentu dengan tujuan untuk memastikan fungsi yang terdapat pada sistem tersebut beroperasi dengan baik dan sesuai dengan kebutuhan dan persyaratan dari *user*.

### D. Low-code Platform

Platform dengan kode rendah adalah sebuah alat untuk programmer dan non-programmer. Ini memungkinkan pembuatan dan penyampaian dengan cepat dari aplikasi bisnis dengan upaya minimal untuk menulis dalam bahasa pengkodean serta upaya yang minimal dalam instalasi, konfigurasi, dan implementasi. Dengan jumlah perusahaan low-code platform yang berkembang pesat, penggunaan low-code platform dapat menjadi solusi signifikan untuk melangkah maju dalam menciptakan aplikasi bisnis yang penting. Sejauh ini, yang tercepat dan mungkin juga metode termurah dalam mengembangkan perangkat lunak [5].

Cara cepat mengembangkan aplikasi yang sekaligus bisa mendesain secara langsung dengan hanya menggunakan sedikit kode manual saja adalah keunggulan utama dari low-code platform. Low-code platform menyediakan antarmuka berbasis grafik dalam mengonfigurasi aplikasi yang akan dibuat, sehingga pengembang tidak perlu lagi melakukan implementasi dengan hanya menuliskan bahasa pengkodean seperti umumnya. Platform ini hadir dalam serangkaian tools yang dapat membantu para pengembang menciptakan aplikasi dengan hanya menggunakan antarmuka drag and drop saja. Selain itu memungkinkan pengembang secara cepat membangun sebuah aplikasi dengan bantuan User Interface yang modern, integrasi, data, dan logic, tanpa menulis ribuan kode dan syntax yang kompleks.

### E. Outsystems

*Outsystems* merupakan platform yang sering dikenal sebagai platform “*low-code*” yang memungkinkan dalam pengembangan seluruh aplikasi dilakukan secara visual, mudah diintegrasikan dengan sistem lain yang ada dan penambahan kode sesuai kebutuhan. Platform tersebut dibuat oleh para *engineer* dengan perhatian yang obsesif terhadap detail. Setiap aspek dari platform *Outsystems* dirancang untuk membantu pengembangan suatu aplikasi berbasis *mobile* maupun web dengan lebih baik dan cepat [6].

Banyak keunggulan yang diberikan jika menggunakan platform *Outsystems*, keunggulan utamanya jika dibandingkan dengan platform lain, *Outsystems* tidak memerlukan banyak kode dasar. Dikarenakan menggunakan *artificial intelligence* membuat para pengembang aplikasi dapat mempersingkat waktu pengembangan.

## III. METODOLOGI

Dalam pengembangan aplikasi ACCAbsensi, penggunaan kerangka kerja *Scrum* diterapkan. Proses pengembangan pada *scrum* sendiri terdiri dari 5 fase [7]. Fase-fase tersebut adalah sebagai berikut :

### A. Initiate

Fase berikut ini adalah tahapan awal dari proses pengembangan yang dilakukan dengan kerangka kerja *Scrum*. Dalam fase ini diantaranya pembentukan tim dilakukan.

Setelah tim telah terbentuk, pembuatan dan penentuan *product backlog* (daftar pekerjaan prioritas yang digunakan untuk pengembang agar sesuai *roadmap* dan *requirements*) dilakukan [7]. Upaya ini dilakukan adalah sebagai fase dasar dalam mengembangkan suatu aplikasi agar selama proses pengembangan menjadi lancar.

### B. Plan and Estimate

Fase berikutnya adalah fase di mana tujuan utama dari dilakukannya fase ini yaitu perencanaan. Setelah *product backlog* di tentukan, kemudian pada fase ini *user story* dituliskan beserta dengan penjabaran *task-task* pada tiap *user story*. Tidak hanya itu, penilaian terhadap setiap *user story* dan *task* diperlukan untuk menentukan tingkat prioritas dalam pengembangan aplikasi yang sedang dikerjakan [8]. Jika dirasa *user story* sudah direncanakan dengan baik, selanjutnya yaitu membuat estimasi pengerjaan dalam pengembangan yang nantinya akan masuk kedalam perencanaan *Sprint*.

Setelah beberapa hal di atas telah dilakukan, *Sprint backlog* ditentukan sebagai upaya dalam memperjelas kembali dalam pengembangan aplikasi yang akan dikembangkan, seperti setiap *task* dalam pengembangan fitur atau menu yang memiliki estimasi durasi pengerjaan berbeda masing-masingnya.

### C. Implement

Pada fase ini yaitu merupakan fase dalam mengeksekusi setiap *task* yang telah didefinisikan dan dibentuk sebelumnya pada fase *plan and estimate*. Pengembangan dan segala aktivitas-aktivitas yang dilakukan dalam upaya mengembangkan suatu aplikasi dilakukan pada fase ini. Dan pada fase ini juga, *daily meeting* dilaksanakan dengan tujuan untuk mengevaluasi semua pekerjaan yang telah dilakukan oleh setiap anggota tim selama jangka waktu yang telah ditentukan. Selain itu, seluruh masalah yang ditemui selama implementasi dapat dibicarakan dalam *daily meeting* tersebut dengan tujuan mencari solusi yang tepat agar *Sprint* berjalan dengan lancar.

### D. Review dan Restropect

Setelah fase implemmentasi dilakukan, pada fase berikut ini adalah fase dilakukannya *review* terhadap hasil pekerjaan yang telah dilakukan selama satu *Sprint*. Namun sebelum dilakukannya *review*, *Manual Guide* harus dibuat terlebih dahulu oleh tim pengembang guna sebagai petunjuk untuk membantu pada tahap pengujian yang dilakukan oleh tim penguji. Dalam fase ini juga, pelaksanaan *System Integration Testing* dan *User Acceptance Testing* dilakukan.

*System Integration Testing* prosesnya dilakukan setelah setiap *sub-sistem* satu-persatu telah lulus diuji *undergone unit testing*. SIT kemudian menguji sistem secara keseluruhan, setelah terintegrasi. Hasil SIT kemudian diteruskan ke *User acceptance testing* (UAT) [9].

*User Acceptance Testing* prosesnya diawali dengan menyediakan dokumentasi persyaratan bisnis, kemudian dilanjutkan dengan proses bisnis (alur kerja) atau skenario dan yang terakhir yaitu pengujian menggunakan data [10]. Tipe UAT yang digunakan adalah *Black Box Testing* yang dikategorikan sebagai uji coba fungsionalitas, *end-user* akan menguji fungsi *software* tanpa melihat struktur kode.

Dari fase ini, semua proses yang telah dihasilkan di *review* kembali agar dapat ditentukan apa saja yang perlu ditingkatkan untuk proses pengembangan pada *Sprint*

berikutnya [8]. *Sprint* dianggap berhasil apabila pada pengembangan yang dilakukan dapat menghasilkan *deliverable product* yang sesuai dengan *user acceptance* yang telah ditentukan dan disetujui oleh *product owner*.

### E. Release

Fase berikut adalah fase yang terakhir dilakukan. Jika produk telah memenuhi seluruh kriteria dari *user acceptance*, maka produk aplikasi tersebut siap di *deployment* dan diberikan seluruhnya ke klien. Setelah produk sudah berhasil pada fase ini berarti *Sprint* dianggap telah selesai dilaksanakan.

## IV. HASIL DAN PEMBAHASAN

### A. Analisis Aplikasi

Aplikasi berbasis *mobile* ACCAbsensi memiliki 4 fitur utama, di antaranya yang pertama yaitu fitur presensi, guna sebagai bukti kehadiran secara *online* beserta keterangannya yang akan disimpan dalam *database* yang nantinya akan dikumpulkan menjadi sebuah data. Kemudian yang kedua terdapat fitur *ToDo*, fitur ini dibuat guna sebagai pembuatan daftar hal yang akan dikerjakan dalam kurun waktu yang telah ditentukan, diharapkan dalam fitur ini dapat menjadi laporan pekerjaan yang telah dilakukan dalam kurun waktu tersebut. Dan yang ketiga terdapat fitur riwayat, dari fitur berikut dapat diketahui riwayat presensi dan keterangan yang lengkap berdasarkan hasil dari dilakukannya semua presensi yang telah dilakukan. Yang terakhir terdapat fitur profil, sudah umum ditemukan pada aplikasi-aplikasi *mobile* lainnya, yaitu fitur yang berguna untuk menampilkan seluruh data diri dari pengguna.

Proses pengembangan aplikasi tersebut dikembangkan dengan menggunakan platform *Ousystems* yang diharapkan dapat menanggulangi permasalahan yang terjadi dengan cepat dan tepat.

### B. Proses Initiate

Pada fase ini, tim telah dibentuk berisikan dari 5 anggota. Di dalam tim tersebut terdiri dari 4 bidang yang berbeda, yaitu *front-end developer*, *back-end developer*, *data analyst*, dan *quality control*. Dari keempat bidang berbeda tersebut, tentunya dipimpin oleh seorang *scrum master* yang mempunyai tugas utama mendukung agar pengembangan dengan metode *Scrum* berjalan dengan lancar. Setelah tim telah dibentuk, *product backlog* dibuat guna untuk mengetahui apa saja *requirements* yang harus diselesaikan selama satu periode *Sprint*. Detail dari *product backlog* yang dibuat bisa dilihat pada Tabel I.

TABLE I. PRODUCT BACKLOG

Product Backlog	Story Point
Sebagai user dapat login ke sistem dengan username dan password	3
Sebagai user dapat registrasi ke sistem dengan mengisi form pendaftaran - nama lengkap - username - password	3
Sebagai user dapat melakukan presensi masuk	5
Sebagai user dapat melakukan presensi keluar	5
Sebagai user dapat melihat durasi waktu bekerja	5
Sebagai user dapat melihat <i>to do list</i> hari ini	3

Product Backlog	Story Point
Sebagai user dapat melihat rangkuman presensi hari ini, durasi waktu bekerja, dan <i>to do list</i> hari ini	1
Sebagai user dapat melihat <i>to do list</i>	3
Sebagai user dapat menambah dan mengedit <i>to do list</i>	5
Sebagai user dapat melihat riwayat presensi dan <i>to do list</i>	1

### C. Proses plan and estimate

Setelah mendapatkan hasil dari *product backlog*, data tersebut akan dibahas dan dirancang kembali yang kemudian dibentuk menjadi *Sprint backlog*. Dari *Sprint backlog* tersebut, semua perencanaan *task-task* dan penjabarannya ditentukan dengan tujuan untuk memperjelas masing-masing *task* yang harus dikerjakan, dimulai dari skala prioritas, hingga durasi waktu pengerjaan. Pada Tabel II berikut, hasil dari *Sprint backlog* yang telah dibuat.

TABLE II. SPRINT BACKLOG

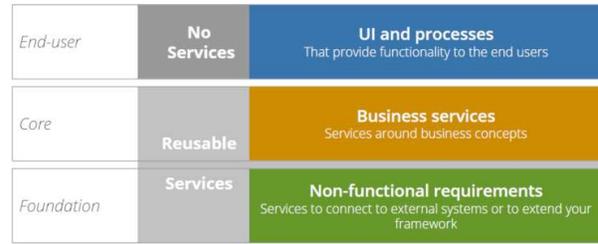
Product Backlog	Story Point	Status	Timeline
Sebagai user dapat login ke sistem dengan username dan password	3	Not in development	Oct 31
Sebagai user dapat registrasi ke sistem dengan mengisi form pendaftaran <ul style="list-style-type: none"> <li>- nama lengkap</li> <li>- username</li> <li>- password</li> </ul>	3	Not in development	Oct 31
Sebagai user dapat melakukan presensi masuk	5	Not in development	Oct 31
Sebagai user dapat melakukan presensi keluar	5	Not in development	Oct 31
Sebagai user dapat melihat durasi waktu bekerja	5	Not in development	Oct 31
Sebagai user dapat melihat <i>to do list</i> hari ini	3	Not in development	Oct 31
Sebagai user dapat melihat rangkuman presensi hari ini, durasi waktu bekerja, dan <i>to do list</i> hari ini	1	Not in development	Oct 31
Sebagai user dapat melihat <i>to do list</i>	3	Not in development	Oct 31
Sebagai user dapat menambah dan mengedit <i>to do list</i>	5	Not in development	Oct 31
Sebagai user dapat melihat riwayat presensi dan <i>to do list</i>	1	Not in development	Oct 31

### D. Proses perancangan sistem

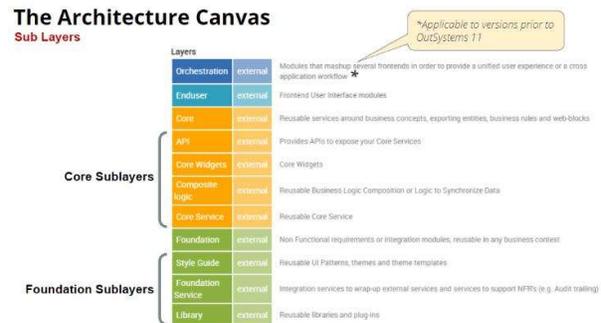
Fase *implement* merupakan fase terpenting dimana aplikasi dikembangkan. Semua kegiatan pengembangan aplikasi yang dilakukan harus sesuai dengan hasil dari perencanaan pada fase *plan and estimate*. *Sprint backlog* menjadi dasar dari tim pengembang aplikasi guna menghindari pengembangan diluar dari permintaan *product owner*. Pada fase ini, *Sprint* umumnya berjalan sekitar 2 hingga 4 minggu, dikarenakan pengembangan aplikasi adalah bagian yang perlu diperhatikan pada metode *scrum* tersebut.

Pengembangan aplikasi menggunakan arsitektur yang dimiliki platform *Outsystems* itu sendiri yaitu *The Architecture Canvas*. Desain arsitektur yang berorientasi layanan sederhana berikut memungkinkan untuk sebuah *services* dapat digunakan kembali pada modul aplikasi yang

berbeda. Desain arsitektur ini memiliki tiga *layer* utama dan didalam *layer* utama tersebut dilengkapi dengan beberapa *sub-layers*. Pada Gambar 1 berikut adalah *layer* utama dari desain arsitektur tersebut dan *sub-layers* nya terdapat pada Gambar 2.



Gambar 1. The Architecture Canvas layers



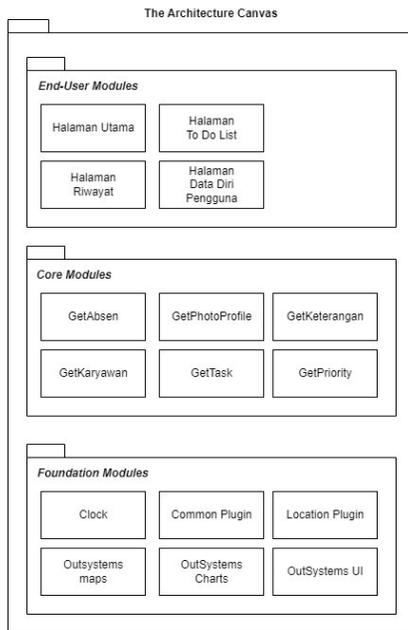
Gambar 2. The Architecture Canvas sub-layers

*End-user Modules* merupakan modul yang berisi desain layar antarmuka yang akan digunakan sebagai perantara atau penghubung antara pengguna aplikasi dengan sistem. Modul ini tidak dapat digunakan kembali seperti dua modul lainnya dikarenakan pada modul ini merupakan layar antarmuka utama yang digunakan pada aplikasi tertentu.

*Core Modules* merupakan bagian terpenting dari *layers* dikarenakan semua proses bisnis berada didalamnya. Dapat dikatakan pada modul ini adalah sebagai tempat semua data disimpan dan diolah. Dari hal tersebut, maka modul ini dapat digunakan berulang-ulang di semua modul yang terdapat pada *Outsystems*, tidak hanya pada aplikasi yang sama, melainkan juga bisa dilain aplikasi sekaligus.

*Foundation Modules* yaitu modul yang berasal dari *Outsystems* itu sendiri. Modul ini adalah layanan yang diberikan oleh *Outsystems* kepada penggunaannya. Layanan tersebut berupa *plug-ins* atau fungsi tambahan yang diperlukan pada saat pengembangan aplikasi dilakukan.

Pada aplikasi ACCAbsensi sendiri dikembangkan menggunakan desain arsitektur *The Architecture Canvas* sehingga aplikasi terbagi menjadi *End-user Modules*, *Core Modules*, dan *Foundation Modules* yang ditunjukkan pada Gambar 3 berikut.

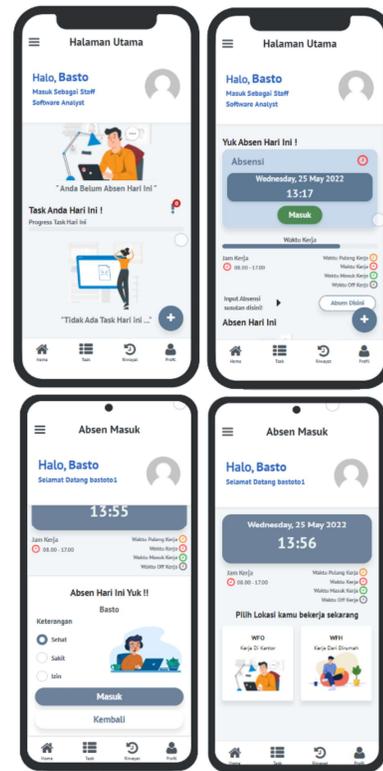


Gambar 3. Package Diagram ACCAbsensi

Selanjutnya adalah bagian implementasi sistem yang akan menampilkan dan menjelaskan antarmuka dari fungsi yang telah dibentuk. Selain itu juga terdapat penjelasan fungsi atau dapat disebut dengan *logic* pada setiap antarmuka.

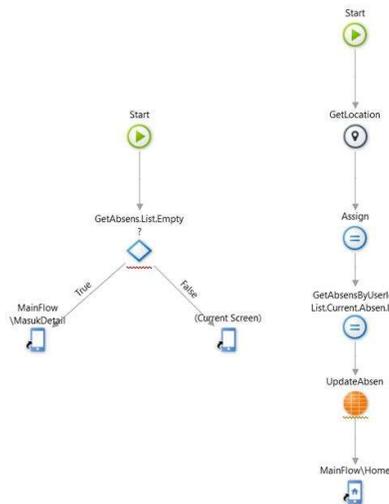
**E. Proses implementasi sistem**

Halaman utama atau *homepage* akan menampilkan beberapa fitur yang diantaranya terdapat fitur presensi masuk dan keluar. Namun pada fitur ini tidak hanya mencatatkan presensi kehadiran saja, melainkan terdapat pencatatan waktu datang dan waktu keluar yang akan disimpulkan apakah karyawan tersebut tepat waktu atau tidak dalam mulai bekerja. Selain itu keterangan masuk juga terdapat pada fitur ini sebagai syarat dicatatnya presensi. Dan tentunya terdapat fitur *tag* lokasi dimana lokasi karyawan saat bekerja dapat diketahui dalam fitur ini. Pada halaman utama berikut juga terdapat fitur riwayat absen yang dilakukan pada hari tersebut beserta task yang harus dikerjakan pada hari itu juga. Berikut adalah tampilan dari halaman utama pada Gambar 4.



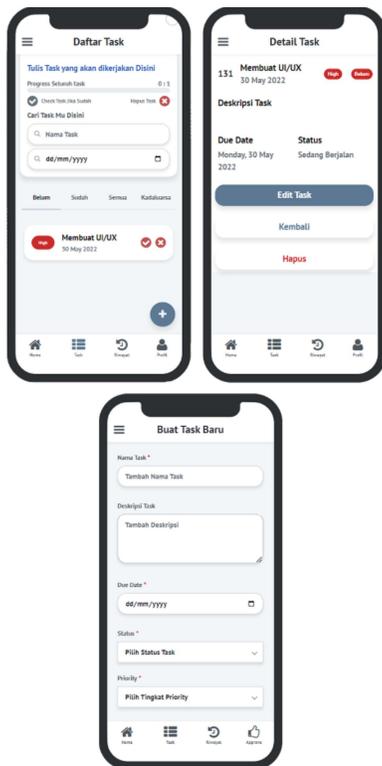
Gambar 4. Halaman Utama

Fungsi *logic* yang dilakukan pada tombol absen masuk adalah pencatatan keterangan presensi, waktu masuk dan keluar, serta lokasi pada saat bekerja. Fungsi berikut dibuat guna agar pencatatan presensi yang dilakukan dapat menjadi data yang dapat disimpan dan diolah dengan baik nantinya. Detail dari fungsi *logic* yang dibuat terdapat pada Gambar 5.



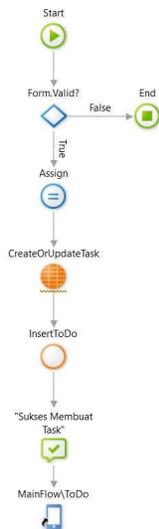
Gambar 5. Alur fungsi *logic* presensi

Halaman *To Do List* merupakan halaman di mana semua pekerjaan dicatat. Tidak hanya menampilkan detail dari tugas yang akan dikerjakan selama bekerja, melainkan dari halaman ini pengguna juga dapat membuat dan mengubah informasi tugas yang akan dikerjakan tersebut. Halaman *To Do List* yang akan ditampilkan akan terlihat seperti pada Gambar 6.



Gambar 6. Halaman To Do List

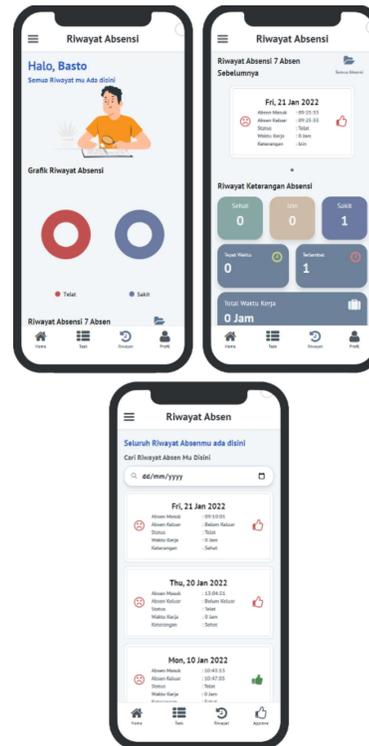
Logic pada tombol buat *task* pada halaman *To Do List* merupakan fungsi yang digunakan untuk melakukan pencatatan *task* pada saat jam bekerja berlangsung. Form yang berisi nama, hingga durasi dan skala prioritas *task* diisi guna sebagai pencatatan *task* yang akan dikerjakan nantinya. Jika *task* telah selesai dilaksanakan, maka terdapat juga *logic* pada tombol centang sebagai penanda bahwa tugas telah selesai dikerjakan. Berikut adalah detail dari *logic* pada tombol buat *task* dapat dilihat pada Gambar 7.



Gambar 7. Alur fungsi *logic* buat *task*

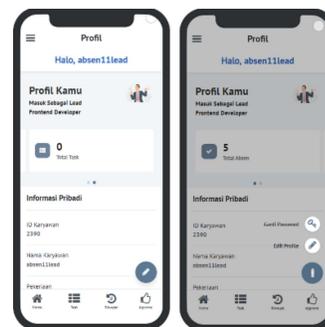
Halaman Riwayat akan berisikan daftar riwayat presensi yang telah dilakukan sebelumnya. Berdasarkan dari kumpulan data presensi yang telah dilakukan, *dashboard* dibuat untuk

mengetahui bagaimana kinerja yang dilakukan selama bekerja. Dari *dashboard* tersebut, maka pengguna akan mengetahui informasi mengenai seluruh presensi yang pernah dilakukan. Detail dari tampilan halaman riwayat bisa dilihat pada Gambar 8.



Gambar 8. Halaman Riwayat

Halaman Profil merupakan halaman yang umum ditemukan pada aplikasi *mobile* lainnya, yaitu sebagai informasi mengenai data diri beserta terdapat fitur perubahan jika ingin memperbarui informasi data diri ataupun kata sandi yang dimiliki. Halaman profil yang akan ditampilkan seperti pada Gambar 9.



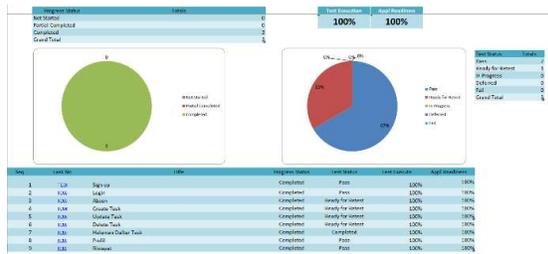
Gambar 9. Halaman Profil

#### F. Proses Review dan Restropect

Fase berikut adalah dimana fase melakukan *testing* atau pengujian terhadap aplikasi yang telah dikembangkan. Berdasarkan *test case* yang telah dibuat oleh tim penguji kemudian akan dibentuklah dokumen *System Integration Testing* dan *User Acceptance Testing*.

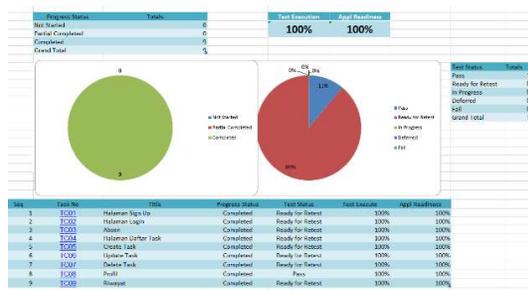
Hasil yang didapatkan dari proses pengujian *System Integration Testing* dari pengembangan yaitu bisa dikatakan

sukses dikarenakan mendapat 67% pengujian dianggap *pass*. Dan dilain sisi mendapat 33% pengujian dianggap meragukan dan siap untuk dilakukan uji ulang. Berikut adalah kesimpulan yang didapatkan dari hasil pengujian SIT pada Gambar 10.



Gambar 10. System Integration Testing

Hasil yang didapatkan dari proses pengujian *User Acceptance Testing* dari pengembangan yaitu bisa dikatakan sukses namun masih harus dilakukan pengecekan kembali dikarenakan mendapat 11% pengujian dianggap *pass*. Dan dilain sisi mendapat 89% pengujian dianggap meragukan dan siap untuk dilakukan uji ulang. Berikut adalah kesimpulan yang didapatkan dari hasil pengujian UAT pada Gambar 11.



Gambar 11. User Acceptance Testing

### G. Proses Release

Pengembangan telah mencapai proses ini yaitu menandakan bahwa satu *Sprint* telah berhasil dilaksanakan. Dan aplikasi siap digunakan dikarenakan telah melewati fase pengujian yang kemudian akan siap untuk diberikan kepada pengguna.

### H. Pembahasan

Pengembangan dapat dilihat bahwa berjalan dengan lancar dikarenakan dapat melewati fase pengujian hingga *release*. Permasalahan yang terjadi pada perusahaan ACC dalam pengawasan dan pengontrolan akhirnya dapat diatasi menggunakan aplikasi ACCAbsensi ini dikarenakan sesuai dengan kebutuhan yang diperlukan oleh para atasan dari perusahaan ACC tanpa harus melakukan pengecekan secara manual dan dapat dibilang tidak memakan waktu banyak karena tidak perlu menghubungi setiap karyawannya, namun sebaliknya yaitu karyawan yang harus melapor ke atasannya.

## V. KESIMPULAN

Berdasarkan aplikasi yang telah dikembangkan, ACCAbsensi adalah aplikasi yang sesuai untuk mengatasi permasalahan yang terjadi pada perusahaan ACC untuk para

atasan dalam mengontrol dan mengawasi para karyawannya. Dari aplikasi ini mempermudah atasan untuk mendapatkan data dari hasil pencatatan yang telah di input tanpa melakukan pengecekan secara manual satu-persatu dari karyawannya. Selain itu, aplikasi dapat digunakan sesegera mungkin dikarenakan hanya memerlukan waktu singkat dalam pengembangan dikarenakan menggunakan platform *Outsystems*.

Beberapa keuntungan yang didapatkan dalam mengembangkan aplikasi dengan *low-code* platform *Outsystems* antara lain yaitu pengembangan menggunakan *low-code* platform dapat memangkas durasi pengembangan aplikasi tersebut dikarenakan platform yang digunakan menggunakan sistem *drag and drop*, membuat pembuatan design pada antarmuka menjadi lebih cepat tanpa menggunakan kode kasar. Selain itu dengan menggunakan *low-code* platform, mampu mengatasi input yang kosong dan juga input angka yang diisi oleh huruf berkat fitur yang dimiliki oleh platform *Outsystems* itu sendiri. Proses pengembangan aplikasi menggunakan *low-code* platform *Outsystems* dapat dikatakan lebih efektif dan mudah dibandingkan dengan pengembangan dengan menggunakan *hard-code* pada umumnya, karena banyak fitur-fitur tambahan yang disediakan oleh platform tersebut untuk mendukung pengembangan.

Melihat dari hasil pengembangan yang telah selesai dilakukan menggunakan platform *low-code Outsystems*, bahwa pengembangan bisa dilakukan dengan cepat dan sangat mempermudah kinerja dari tim pengembang. Saran untuk penelitian selanjutnya di mana *database* tempat data disimpan yang masih hanya berkuat pada *core module* saja mungkin bisa dikembangkan lagi dengan mengintegrasikan data aplikasi ke *database* luar dengan menggunakan *API (software* yang mengintegrasikan aplikasi yang dibuat dengan aplikasi yang lainnya).

## REFERENCES

- [1] Allen, T. D., Golden, T. D., & Shockley, K. M. 2015. *How effective is telecommuting? Assessing the status of our scientific findings*. Psychological Science in the Public Internet, 16(2), 40-68.
- [2] Rubin, Kenneth S. 2013. *Essential Scrum: A Practical Guide to the Most Popular Agile Process*. Addison-Wesley Professional.
- [3] Watkins, Jons dan Simons, Mills. 2010. *Testing IT An Off-the-Shelf Software Testing Process*. Cambridge University Press.
- [4] Perry, William E. 2007. *Effective Methods for Software Testing Third Edition*. Willey. [www.google.co.id/books/edition/Effective\\_Methods\\_for\\_Software\\_Testing\\_C/q1WwdUZd7IAC?hl=en&gbpv=0&kptab=overview](http://www.google.co.id/books/edition/Effective_Methods_for_Software_Testing_C/q1WwdUZd7IAC?hl=en&gbpv=0&kptab=overview)
- [5] Waszkowski, Robert. 2019. *Low-code platform for automating business processes in manufacturing*. IFAC-PapersOnLine, Vol 52, Issue 10, 376-381.
- [6] Menezes, Francisco. 2020. *Application Architecture: Best Practices for Future-Proofing Your Apps*. Diakses pada 24 Mei 2022. [www.outsystems.com/blog/posts/application-architecture/](http://www.outsystems.com/blog/posts/application-architecture/).
- [7] T. Satpathy. 2016. *Scrum Body Of Knowledge*. Sbok Guid.
- [8] Schwaber, K., & Sutherland, J. 2013. *The definitive guide to Scrum*. The rules of the game.
- [9] Pedamkar, Priya. 2022. *System Integration Testing*. Diakses 1 Juni 2022. [www.educba.com/system-integration-testing/](http://www.educba.com/system-integration-testing/).
- [10] T. M. Branch. 2008. *User Acceptance Testing (UAT) Process*. Inf. Technol. Manag. Branch, vol. Ed. Britis, pp. 111-112.